UNIVERZITET U BEOGRADU

FAKULTET ORGANIZACIONIH NAUKA

Bojan D. Jovanović

UPRAVLJANJE PERFORMANSAMA DISTRIBUIRANOG MULTIBIOMETRIJSKOG EKOSISTEMA

doktorska disertacija

Beograd, 2018.

UNIVERSITY OF BELGRADE

FACULTY OF ORGANIZATIONAL SCIENCE

Bojan D. Jovanović

MANAGE THE PERFORMANCE OF A DISTRIBUTED MULTIBIOMETRIC ECOSYSTEM

Doctoral Dissertation

Belgrade, 2018.

Mentor:

Prof. dr Dušan Starčević, redovni profesor
Univerzitet u Beogradu, Fakultet organizacionih nauka

Članovi Komisije:

Prof. dr Dejan Simić, redovni profesor
Univerzitet u Beogradu, Fakultet organizacionih nauka

dr Miroslav Minović, vanredni profesor
Univerzitet u Beogradu, Fakultet organizacionih nauka

dr Dragana Makajić-Nikolić, vanredni profesor
Univerzitet u Beogradu, Fakultet organizacionih nauka

Prof. dr Zoran Jovanović, redovni profesor
Univerzitet u Beogradu, Elektrotehnički fakultet

Datum odbrane: _____ 2018. godine

Zahvalnica

Imao sam privilegiju i sreću da upoznam prof. dr Dušana Starčevića, da radim i učim pod njegovim mentorstvom. Bez njegove podrške, naučne izvrsnosti, požrtvovanja, nesebičnosti kao i veličine u svakom pogledu, profesionalne i lične, ova disertacija, u najboljem slučaju, ne bi ni približno mogla biti ovakva. Zahvalan sam svom mentoru zbog toga.

Veliku zahvalnost dugujem i kolegama sa Katedre za informacione tehnologije, bez njihove saradnje, pomoći, prijateljske i profesionalne podrške bi bilo teže privesti kraju ovakav posao u ovome obimu.

Ovim radom se zahvaljujem mojoj porodici koja je svih ovih godina bila primorana da podnosi žrtvu mog bavljenja naučnim radom u uslovima koji su nam bili nametnuti; njihova podrška i razumevanje mog ličnog shvatanja značaja bavljena naukom nijednog trenutka nisu izostali.

Upravljanje performansama distribuiranog multibiometrijskog ekosistema

Sažetak

Ova doktorska disertacija se detaljno bavi problemom optimizacije performansi distribuiranog biometrijskog sistema kao vitalnih delova sistema za menadžment identiteta. Savremeni sistemi za upravljanje digitalnim identitetima se danas najčešće koriste za autentifikaciju korisnika prilikom pristupa entitetima informacione infrastrukture, bilo da je reč o korporativnom intranet okruženju ili Internetu. Tipičan korisnički zahtev je da se omogući kontrolisani pristup lokalnim entitetima sa bilo koje tačke u svetu. Digitalni identitet korisnika poseduje skup atributa koji identifikuju pojedinca, na primer, imenom i prezimenom, ličnim identifikacionim kodom (jedinstveni matični broj i/ili PIN) ili brojem pasoša, a sve češće i biometrijskim podacima. Takav digitalni identitet se nalazi upakovan u biometrijskim ličnim dokumentima. Izdavalac ovakvih dokumenata čuva u svom sistemu za upravljanje identitetima i biometrijske uzorke lica kome je izdao biometrijski dokument. Korišćenje biometrijskih uzoraka prilikom autentifikacije pruža veći nivo pouzdanosti u postupku utvrđivanja identiteta. Pristup optimizaciji u ovoj doktorskoj disertaciji je baziran na uvođenju konceptualnog modela multibiometrijskog ekosistema, njegovoj generičkoj arhitekturi kao i uvođenju modela kojim je moguće upravljati performansama mutlibiometrijskog ekosistema. Pri uvođenju konceptualnog modela pošlo se od meta modela softverskog ekosistema i njegovih osnovnih gradivnih elemenata. Model upravljanja performansama multibiometrijskog sistema je zasnovan na modelu višekriterijumskog odlučivanja koji se koristi pri dizajnu distribuiranog računarskog sistema. Uvođenjem heuristika u višekriterijumski model omogućeno je upravljanje nad multibiomtrijskim ekosistemom u realnom vremenu. U samom radu je prikazana primena predloženog modela na izabranom Mutimodalnom biometrijskom sistemu MMBio. Pokazana je upravljivost nad multibiometrijskim ekosistemom kako tokom izgradnje ekosistema tako i tokom izvršavanja u zavisnosti od opterećenja računarskog sistema.

Ključne reči: Softverski ekosistem, Multibiometrijski sistem, Performanse sistema, Distribuirani računarski sistemi

Naučna oblast: Organizacione nauke

Uža naučna oblast: Informacione tehnologije

UDK 57.087.1:004

Manage the performance of a distributed multibiometric ecosystem

Abstract

This doctoral dissertation deals in detail with the problem of optimizing the performance of the distributed biometric system as vital parts of the identity management system. Modern digital identity management systems are today most often used to authenticate users when accessing information infrastructure entities, whether it's a corporate intranet environment or the Internet. A typical user request is to allow controlled access to local entities from anywhere in the world. The digital identity of a user has a set of attributes that identify an individual, for example, name and surname, a personal identification code (unique identification number and /or PIN) or a passport number, and more often with an biometric data. Such a digital identity is packaged in biometric personal documents. In its identity management system the publisher of such documents keeps biometric samples of the person to whom biometric document was issued. The use of biometric samples during authentication provides a greater degree of reliability in the identification process. The approach to optimization in this doctoral dissertation is based on the introduction of the conceptual model of the multibiometric ecosystem, its generic architecture, and the introduction of a model that can manage the performance of multibiometric ecosystems. The meta model of the software ecosystem and its basic structural elements are starting point in introducing the conceptual model. The multibiometric system performance management model is based on a multicriterial decision-making model used in the design of a distributed computing system. By introducing heuristics into a multi-criteria model, it is possible to manage the multibiometric ecosystem in real time. The paper presents the application of the proposed model on the chosen multimodal biometric system MMBio. The management of the multibiometric ecosystem is shown both during the construction of the ecosystem and during the execution depending on the load of the computer system.

Keywords: Software ecosystem, Multibiometric system, System performance, Distributed computer systems

Scientific area: Organizational sciences

Specific scientific area: Information technology

UDC 57.087.1:004

Sadržaj

Tabele

Slike

1 Uvod

U savremenom svetu mobilnost ljudi je povećana. Ona sa sobom donosi mnogobrojne izazove. Tokovi roba, usluga i novca su time sve zamršeniji. U svim sferama života Internet je postao sveprisutan. Mnoge transakcije se odigravaju ”online”. U takvim komplikovanim aspektima života i rada neophodno je obezbediti siguran menadžment identiteta.

Ekspanzija elektronskog poslovanja i druge ”online”usluge su dovele do toga da dosadašnji, trandicionalni, načini utvrđivanja identiteta postaju neprimenljivi. Pored prodaje čitavi poslovni procesi su automatizovani i prebaćeni u eletronski svet. Komunikacija sa dobavljačima i poslovnim partnerima odvija se preko Interneta [5] . Sve ove promene su povećale rizike poput ugrožavanja privatnosti, zloupotrebe ili krađe identiteta. Šta više, može se reći da poprimaju alarmantne razmere.

Digitalni identitet je našao primene kako u poslovnoj sferi, tako i u brojnim drugim oblastima, kao što su e-uprava, zdravstvo, obrazovanje, itd [6] . Digitalni identitet korisnika poseduje skup atributa koji identifikuju pojedinca, na primer, imenom i prezimenom, ličnim identifikacionim kodom (jedinstveni matični broj i/ili PIN) ili brojem pasoša. Ovako koncipiran identitet pojedinca se može koristiti u raznim servisima koji imaju potrebu jednoznačne autentifijacije odnosno verifikacije pojedinca. Ovi servisi mogu biti informacionog ili transakcinog karaktera. Najčešću primenu ovakvih digitalnih identiteta možemo videti u servisima kojima svojim građanima omugućuju vlada, ministartstva, carina, agencije ili druge uprave kroz lična dokumenta: lična karta ili pasoš. Takođe primena primena digitalnog identiteta se može naći i u bankama koje omogućuju svojim klijentima usluge obavljanja novčanih transakcija izdajući platne kartice na osnovu ličnih dokumenata. Država se pojavljuje u ulozi garanta za postojanje jedinstvenog digitalnog identiteta za svakog građanina jedne zemlje. Izgradnjom PKI (Public Key Infrastructure) kao IKT infrastrukture i dodeljivanjem sertifikata identitetu obezbeđuje se sigurna i verifikovana digitalna komunikacija [7].

Sistemi za upravljanje identitetom obezbeđuju upravljanje nad digitalnim identitetima tako što obezbeđuju ključne aktivnosti a one su: identifikacija, autentifikacija, autorizacija i upravljanje korisničkim nalozima. Identifikacija predstavlja proces nalaženja određenog identiteta, dok je autentifikacija proces utvrđivanja validnosti identiteta [8]. U literaturi neki autori poistovećuju procese indetifikacije i autentifijacije [9]. Upravljanje korisničkim nalozima je povezano sa

autorizacijom jer se tokom aktivnosti upravljanja definišu prava pristupa, upotreba resursa, uloga u organizaciji, kao i druga korisnička prava kako u domenu informaciono komunikacionih tehnologija tako i u realnom svetu. Aktivnost autorizacije se uvek obavlja nad autentifikovanim identitetom.

U tradicionalnim sistema autentifikacije proces se vrši na osnovu klasa faktora autentifikacije. Najstarija klasa faktora se zasniva na nečemu što samo taj korisnik zna: lozinka, PIN. Druga klasa faktora se zasniva na nečemu što korisnik ima odnosno poseduje: identifikaciona kartica, bezbedonosni token koji može biti softverski ili hardverski [10].

U poslednjih nekoliko godina se razvila i hibridna klasa autentifikacije dvo-faktorska autentifikacija. Ona obuhvata uređeni par onoga što korisnik ima (bankovnu karticu, mobilni telefon) i onoga što korisnik zna PIN (unapred dodeljen bankovnoj kartici ili poslat porukom na mobilni telefon) [11]. Ovakav pristup rešava probleme zaštite identiteta koje smo imali u prošlosti. On ne rešava probleme koje imamo danas i koje ćemo imati pre svega u skorijoj budućnosti.

Današnji život se živi na Internetu. Lozinka se može izgubiti, zaboraviti, zapisati tako da je neko drugi može pročitati, presresti kroz otvorene komunikacione kanale, odnosno veoma lako se može izgubiti kontrola na njome. Posedovanje uređaja za dobijanje softverskog ili hardverskog tokena ne garantuje da vlasnik identiteta u trenutku vršenja autentifikacije stvarno poseduje uređaj.

Moguće rešenje problema tradicionalnih načina autentifikacije se može rešiti uvođenjem nove klase faktora autentifikacije gde su faktori vezani za ono što korisnik ima ili ono što korisnik jeste, odnosno uvođenjem biometrijske identifikacije [12].

Za biometriju se može reći da ona predstavlja skup tehnika za jedinstveneno prepoznavanje i identifikovanje ljudi na osnovu njihovih fizičkih, hemijskih i obrazaca ponašanja. Biometrija se već nekoliko godina unazad primenjuje u različite svrhe i načine. Od pristupa radnim stanicama, zaštite podataka, udaljenog pristupa resursima, sigurnosti u transakcijama, do biometrijskih dokumenata koja se mogu koristiti i van informaciono-komunikacionih tehnologija. Sve primene biometrije trebaju da zadovolje potrebe:

- fizičke sigurnosti
- sajber sigurnosti
- transakcionu sigurnost

Kod fizičke sigurnosti se obezbeđuje siguran pristup fizičkoj lokaciji: soba ili zgrada ili bilo kakav drugi skup objekata. Za sajber sigurnost može se reći da obezbeđuje pristup IKT resursu: radnoj stanici, lokalnoj računarskoj mreži ili nekom drugom resursu jedne organizacije bilo lokalno bilo sa udaljene lokacije. Transakciona sigurnost obezbeđuje validaciju transakcije kojom se pristupa resursu korisnika. Na primer, kod novčanih transakcija biometrijskom autentifikacijom se obezbeđuje validacija zaduženja ili kreditiranja računa korisnika. Ili u transakciji kojom se menjaju vrednosti atributa identiteta pojedinca bimetrijskom autentifikacijom i autorizacijom se pravi žurnal promene.

Biometrija se zasniva na biološkim merenjima koja se mogu proglasiti biometrijskom karakteristikom ukoliko zadovolje sledeće uslove:

- Univerzalnost - Svaka osoba treba da poseduje datu karakteristiku
- Osobenost - Bilo koje dve osobe trebaju da budu različite u pogledu date karakteristike
- Stalnost - Data karakteristika treba da bude stalna tokom nekog perioda vremena
- Merljivost - Karakteristika treba biti kvantitativno merljiva
- Performanse - Tačnost prepoznavanja korišćenjem biometrijske karakteristike mora da zadovoljava postavljene zahteve sistema gde sredstva uložena u postizanje performanse trebaju da zadovolje zadate granice
- Prihvatljivost - Korisnici biometrijskog sistema trebaju da iskažu spremnost u korišćenju svojih biometrijskih karakteristika
- Mogućnost prevare - Stepen mogućnosti falsifikovanja biometrijske karakteristike

Ni jedna karakteristika u potpunosti ne zadovoljava ove zahteve, odnosno nije moguće koristiti jednu biometrijsku karakteristiku u svim slučajevima korišćenja. Veći broj biometrijskih karakteristika je prihvatljiv za korišćenje pod određenim uslovima [1].

Jedan biometrijski sistem može se definisati kao sistem koji pri prepoznavanju osobe koristi obrasce prepoznavanja zasnovane na vektorima karakteristika nastalim na osnovu specifične fiziološke ili obrasca ponašanja koje osoba poseduje [13] .

Biometrijski sistemi koji koriste jednu biometrijsku karakteristiku nazivaju se unimodalni biometrijski sistemi. Problemi sa kojima se susreću unimodalni

Biometrijski identifiaktor	Univerzalnost	Osobenost	Stalnost	Merljivost	Performansa	Prihvatljivost	Mogućnost prevare
DNK	V	V	V	N	V	N	N
Uvo	S	S	V	S	S	V	S
Lice	V	N	S	V	N	V	V
Termogram lica	V	V	N	V	S	V	N
Otisak prsta	S	V	V	S	V	S	S
Hod	S	N	N	V	N	V	S
Geometrija šake	S	S	S	V	S	S	S
Iris	V	V	V	S	V	N	N
Kucanje na tastaturi	N	N	N	S	N	S	S
Miris	V	V	V	N	N	S	N
Otisak dlana	S	V	V	S	V	S	S
Retina	V	V	S	N	V	N	N
Potpis	N	N	N	V	N	V	V
Glas	S	N	N	S	N	V	V

Tabela 1. Upoređenje pogodnosti korišćenja različitih biometrijskih tehnologija na osnovu percepcije autora [1]. Visoka - V, Srednja - S, Niska - N.

biometrijski sistemi su vezani za uslove stalnosti i merljivosti biometrijske karakteristike. Dva uzorka iste biometrijske karakteristike jedne osobe se mogu razlikovati zbog nesavršenosti uslova pod kojim je vršeno merenje, zbog promena biometrijske karakteristike jedne osobe, zbog ambijentalnih uslova i načina na koji je osoba koristila biometrijski instrument u trenutku merenja. Zbog gore navedenih razloga sračunati vektor karakteristika biometrijskog uzorka se često neće poklopiti sa prethodno sačuvanim vektorom biometrijskog uzorka. Takođe se ne mogu koristiti ni obrasci prepoznavanja između uzetog i prethodno sačuvanog biometrijskog uzorka. Neće dati odgovor na pitanje jednakosti već odgovor na pitanje sličnosti. Odnosno koriste se tehnike dodeljivanja odgovarajućeg skora sličnosti između dva uzorka kvantifikovanog putem jednog broja.

1.1 Opis problema i motivacija

Kako je navedeno u uvodu postoje različiti problemi u unimodalnim biometrijskim sistemima. Ne postoji ni jedan unimodalni biometrijski sistem koji može sa stoprocentnom tačnošću uradi proces identifikacije osobe na osnovu uzetog biometrijskog uzorka. Kao rešenje nedostataka unimodalnih biometrijskih sistema pojavili su se multibiometrijski sistemi koji mogu da koriste fuziju više osobina jedne osobe ili višestruku ekstrakciju osobina i primenu različitih algoritama nad istom osobom [14] .

Biometrijski sistemi su dostigli granicu skalabilnosti zbog konstantnog unosa novih biometrijskih uzoraka. Baze biometrijskih podataka prelaze sa unimodalnih, u kojima se čuva samo jedan tip biometrijskih uzoraka, na multi-modalne, u kojima se čuvaju različiti tipovi biometrijskih uzoraka. Količina informacija koje treba

sačuvati raste čak i brže kada se zna da nekoliko biometrijskih modaliteta može biti povezano sa svakim identitetom u ovakvim bazama podataka. Obično su to otisci prstiju, slike lica, dužica oka, zajedno sa novim modalitetima kao što su otisci dlana, glas i DNK. Na primer, sve je više zemalja u svetu koje prelaze na biometrijska lična dokumenta, za koje se uzimaju otisci prstiju i slike lica [15, 16] . Ovi različiti modaliteti mogu zauzeti desetine petabajtova biometrijskog materijala u skladištima relacionih baza podataka. U bliskoj budućnosti, ovi sistemi neće biti samo pitani da identifikuju pojedinca u realnom vremenu, već će biti potrebno da dele informacije među različitim organizacijama da zadovolje širok spektar zadataka upravljanja nad identitetima. Razni biometrijski sistemi se testiraju, postavljaju i puštaju u upotrebu tako da će doći do izražaja potreba da se ima pristup ne samo podacima već i različitim biometrijskim alatima, na primer u mobilnim biometrijskim sistemima.

Moderni sistemi za upravljanje nad identitetima se danas koriste za autentifikaciju prilikom pristupa entitema informacione infrastrukture koja se iz Intranet okruženja seli na Internet okruženje. Lokalnim entitetima se pristupa sa bilo koje tačke u svetu. Digitalni identitet poseduje skup atributa koji identifikuju pojedinca koji je određen imenom i prezimenom, ličnim identifikacionim kodom (jedinstveni matični lični broj i/ili PIN) ili broj pasoša. Ovakav digitalni identitet se nalazi upakovan u biometrijska lična dokumenta. Izdavalac ovakvih dokumenata čuva u svom sistemu za upravljanje nad identitetima i biometrijske uzorke lica kome je izdao biometrijski dokument. Korišćenje biometrijskih uzoraka prilikom autentifikacije omogućuje veći nivo pouzdanosti prilikom određivanja identiteta.

Istraživanja u oblasti biometrije zavisi od efektivnog upravljanja nad ogromnom količinom podataka i obradom tih podataka na procesorima računara. Tekući istraživački projekti u oblasti biometrije zahtevaju terabajte slika i video zapisa subjekata zajedno sa detaljnim metapodacima biometrijskih uzoraka. Eksperimenti velikih razmera trenutno zahtevaju ogromni nivo eskpertize u poznavanju rada računarskih sistema. Korisnici moraju da budu delotvorni u konfigurisanju i korišćenju grid računarskih sistema, relacionim bazama podataka, distriuiranim fajl sistemima i biti svesni mnogih osnovnih funkcionalnih ograničenja i uticaja na performanse. Takođe problem skalabilnosti u radu ovih sistema ostaje otvoren. Problemi rešeni u eksperimetima malih razmera, ne garantuju da će eksperiment moći da se proširi na veće razmere bez uvođenja novih tehnika i tehnologija. Podatke, alate, tehnologije i tehnike je često jako teško deliti čak i među istraživačima jedne institucije jer se oslanjaju na kompleksnu hrpu ručno podešenog softvera.

Danas se mnogi naučni problemi rešavaju u velikim timovima u kojima su resursi distribuirani globalno te članovi tima imaju potrebu da pristupe podacima, računarskoj, mrežnoj i telekomunikacionoj infrastrukturi [17] . Upotreba distribuiranih sistema u rešavanju računskih problema se danas često naziva distribuirano računarstvo. U distribuiranom računarstvu je problem podeljen na mnošto poslova koji se izvršavaju na jednom ili više računara a komuniciraju međusobom razmenom poruka [18, 19] . Različite softverske i hardverske arhitekture se koriste u distribuiranom računarstvu. Na nižem nivou, neophodno je povezati više procesora sa nekim tipom mreže bez obrzira na to da li je mreža na štampanoj ploči ili je skup labavo spojenih uređaja i kablova. Na višem nivou potrebno je povezati procese koji rade na tim procesorima sa nekim tipom komunikacionog sistema. Sa tačke gledišta softvera moguće je koristiti nekoliko arhitektura ili kategorija: client-server, 3-tier, n-tier, distributirane objekte, klastere, grid, peer-to-peer, virtualizaciju (prostorno orijentisana arhitektura).

1.2 Cilj rada i polazna hipoteza

Osnovni cilj ove disertacije je da se definiše programski okvir za razvoj biometrijskih distribuiranih sistema korišćenjem poznatih klaud tehnologija. Na osnovu predmeta istraživanja, daju se sledeći podciljevi:

Glavna hipoteza: Pri zadatim projektnim zatevima u pogledu menadžmenta identiteta, mogućnostima i ograničenjima raspoloživog distribuiranog multibiometrijskog ekosistema, moguće je definisati metodologiju koja omogućuje efektivno i efikasno upravljanje performansama implementiranog sistema.

Pomoćne hipoteze:

H_1 – Moguće je identifikovati relevantne elemente i njihove veze u multibiometrijskom ekosistemu za potrebe sistema menadžmenta identiteta, modelirati odgovarajuću generičku arhitekturu i implementirati generičko radno okruženje distribuiranog multibiometrijskog sistema uz korišćenje poznatih i dostupnih cloud tehnologija.

H_2 - Na osnovu zadatih projektnih zahteva u pogledu željenih svojstava multibiometrijskog sistema moguće je postaviti model odgovarajućeg konkretnog radnog okruženja.

H_3 - Na osnovu modela traženog radnog okruženja i generičke arhitekture distribuiranog multibiometrijskog sistema može se razviti metodologija koja omogućava upravljanje performansama konkretne implementacije.

H_4 - Predložena metodologija upravljanja performansama multibiometrijskog sistema omogućava projektantima implementaciju sistema i u složenim okruženjima bez poznavanja detalja implementacije okruženja.

H_5 - Moguće je razviti i održavati katalog gradivnih elemenata multibiometrijskog sistema koji se koriste u realizaciji konkretnog obradnog modela.

1.3 Struktura rada

Rad je organizovan u jedanast poglavlja.

U prvom, uvodnom poglavlju, je predstavljen problem, predmet i ciljevi istraživanja. Polazi se od pregleda biometrije i biometrijskih karakteristika, opisa problema postavljenih ciljeva i kratkog pregleda organizacije disertacije.

U drugom poglavlju uvedeni su pojamovi identiteta i sistema za upravljanje identitetima, biometrijski sistemi kao i taksonomija biometrijskih sistema. U ovom poglavlju su detaljno analizirani biometrijski modaliteti, načini rada biometrijskog sistema kao i tipovi fuzije mutibiometrijskih sistema.

U trećem poglavlju su analizirani postojeći multimodalni sistemi i utrđeni parametri na osnovu kojih su analiziorane performanse.

Predstavljen je konceptualni model multibiometrijskog ekosistema za utvrđivanje identiteta u četvrtom poglavlju. Identifikovani si elementi ekosistema kao i tehnologije virtualizacije i računarstva u oblaku koje su gradivni elementi multibiometrijskog ekosistema.

U petom poglavlju je predložena generička arhitektura multibiometrijskog ekosistema. Takođe je dat pregled delova generičke arhitekture i objašnjene veze između njih.

U šestom poglalju je dat model radnog okruženja distribuiranog multibiometrijskog sistema. Opisana je hardverska, komunikaciona i softverska infrastruktura.

U sedmom poglalju su navedeni korisnički zahtevi i model logičke arhitekture konkretnog obradnog sistema.

Osmo poglavlje opisuje model upravljanja performasama distribuiranog multibiometrijskog sistema. Definisani su parametri sistema koje je potrebno pratiti prilikom optimizacije performansi i dat je višekriterijumski model na osnovu koga je moguće izgraditi distribuirani multibiometrijski sistem. Uvedene su heuristike u model sa ciljem efikasnijeg upravljanja performansama na osnovu definisanih parametara sistema.

U devetom poglavlju je određena konkretna fizička arhitektura distribuiranog multibiometrijskog sistema koja je iskorištena za studiju slučaja.

Studija slučaja je predstavljena u desetom poglavlju. Eksperimentalni deo istraživanja je realizovan u dve faze. U prvoj fazi analizirana je izgradnja šablona multibiometrijskog sistema i verifkovana je kroz instalaciju sistema. U drugoj fazi je ispitana moguća transformacija Multimodalnog biometrijskog sistema MMBio urađenog na projektu TR-32013 ”Multimodalna biometrija u upravljanju identitetima”.

Jedanaesto poglavlje sadrži diskusiju dobijenih rezultata. Verifikovani su modeli upravljanja performansama multibiometrijskog distribuiranog sistema.

U dvanaestom poglavlju su dati zaključci, doprinosi i predlozi pravaca budućih istraživanja. Na kraju rada dat je pregled korišćene literature

2 Menadžment identiteta i multibiometrijski sistemi za utvrđivanje identiteta

2.1 Sistemi za upravljanje identitetima

2.1.1 Upravljanje identitetom

Još uvek ne postoji precizna definicija identiteta. Postoji više različitih definicija u zavisnosti od konteksta upravljanja identitetima. Ako se identitet povezuje samo sa osobom moće se koristiti definicija koju predlažu Pfitzman i Hansen [20] koji definišu identitet kao : ”Identitet jedne osobe može da sadrži više parcijalnih identita od gde svaki predstavlja tu osobu u specifični kontekst ili ulogu. Parcijalni identitet je podskup vrednosti atributa jednog kompletnog identiteta, gde je kompletni identitet unija svih vrednosti atributa svih identiteta te osobe.”. Međutim, postoje i definicije identiteta koje pokrivaju ne samo ljude nego širi skup subjekata. Među njima može se izdvojiti Bishop [21], koji navodi da subjekat identiteta mogu biti softverski agenti (na primer: Veb servisi i klijent sa strana softvera) ili hardverski uređaji (na primer: računar, mobilni telefon ili mrežna oprema). Štaviše, kako računarska okruženja postaju sve prisutna, identitet se može dodeliti kako veštačkim objektima (na primer: dobra, delovi mašina, zgrade) tako i prirodnim objektima (na primer: stoka, usevi) koji se prate i kojima se upravlja uz pomoć senzora. Poslednjih nekoliko godina se dosta radi i napreduje u donošenju standarda u oblasti upravljanja identitetima. Neki od tih standarda uključuju i definiciju identiteta. Kao primer može se navesti standard ITU-T Y.2720 [22] u kome je definisan identitet kao ”informacija o entitetu koja je dovoljna da identifikuje taj entitet u specifičnom kontekstu”. Prema Y.2720 standardu, jedan identitet se sastoji iz tri različita tipa podatka: identifikatora, kredencijala i atributa.

- Identifikator: predstavlja bilo koju formu podatka koji se može iskoristiti da identifikuje subjekat. Kombinovanjem cifara, slova i simbola se može dobiti korisničko ime, broj pasoša, broj mobilnog telefona, pseudonim i URI.
- Kredencijal/Akreditiv: skup podataka koji obezbeđuje dokaz za tvrdnju o delovima ili celom identitetu. Jedan kredencijal se može generisati na osnovu jednog ili više kredencijala. Mogući primeri su lozinke, digitalni sertifikati, Kerberos tiketi.

- Atributi: skup podataka koji opisuju karakteristike subjekta. Podaci uključuju osnovne informacije koje identifikuju subjekat (na primer: ime i prezime, mesto stanovanja, datum rođenja), osobenosti subjekta i informacije nastale na osnovu aktivnosti subjekta. (na primer: starost, pol, uloge, zvanja, reputacija, zapisi o aktivnosti)

Kategorizacija identiteta se može raditi sa različitih perspektiva posmatranja. Diskusije koje se vode o identitetu pokrivaju širok spektar naučnih disciplina uključujući sociologiju, psihologiju, filozofiju kao i računarske nauke. Identiteti se uglavnom koncipiraju kroz perspektivu procesa i strukturnu perspektivu barem što se tiče kompjuterske nauke. Što se tiče strukturne perspektive, identitet se posmatra kao skup atributa koji karakterišu osobu, dok se kroz perspektivu procesa jedan identitet konceptualizuje za potrebe identifikacije kao "skup procesa koji se odnose na otkrivanje informacija o osobi i upotrebi tih informacija"[23].

Moguća kategorizacija se može izvršiti na osnovu ko poseduje i kontroliše identitet. Durand izdvaja tri kategorije [24]:

1. Lični identitet: osoba poseduje i u potpunosti kontrološe identitet.
2. Korporativni identitet (dodeljeni identitet): Ovakav identitet se odnosi na specifičan kontekst (na primer poslovni odnos) i predstavlja privremeno dodeljene ili izdate karakteristike osobe - zvanje, broj telefona ili slično.
3. Apstraktni ili agregirani identitet: osoba se ne posmatra kao pojedinac, nema svoje ime, ali nastaje kao rezultat filtriranja nad datim skupom karakteristika. Primeri mogu biti sledeći: ima vozačku dozvolu duže od tri godine, živi na selu, rekreativno se bavi sportom, kontaktiran je od strane prodavca.

Na osnovu Durandove kategorizacija proističe nekoliko aspekata u upravljanju identitetima koji se tiču životnog ciklusa identiteta, zaštite i privatnosti. U ISO/IEC 27000 standardu [25] definisana su tri aspekta zaštite identiteta: poverljivost, integritet i dostupnost. Navedenim aspektima su označena mesta mogućih slabosti u sistemima za upravljanje identitetima u kojima je moguće izvršiti napade, na primer krađa identiteta i pecanje (eng. fishing). Konvencionalni napadi protiv kojih se treba zaštiti u sistemima za upravljanje identitetima su opisani u RFC 3552 [26]. Privatnost obuhvata veći broj koncepata, od prava "da me ostave na miru" pa do zaštite ličnosti, intimnosti. Za potrebe upravljanja digitalnim identitetima privatnost se može definisati kao pravo subjekta da kontroliše svoje identitete u transakcijama identiteta kroz sistem za upravljanje identitetima [2]. Zaštita privatnosti u upravljanju identitetima ima veoma veliki značaj posebno ako se

posmatraju prava korisnika u korišćenju usluga u potrošačkom društvu. Danas, upravljanje identitetima najčešće se upotrebljava u slučajevima u kojima zaštita privatnosti nije primarna, na primer u preduzećima i organizacijama. U ovakvim slučajevima zaposleni i poslovni partneri imaju manju kontrolu nad svojim identitetima, već je identitet povezan sa pravima i obavezama posla koji obavljaju.

U privatnom i poslovnom životu danas Internet je postao deo komunikaciono-informacione infrastrukture. Svaki pojedinac koristi različite usluge koje pružaju davaoci servisa, od eUprave preko korporativnih portala do email naloga. U svakom od tih servisa osoba ima svoj identitet. Svaki od tih identiteta postoji samo u okruženju koje obezbeđuje identifikaciju te osobe za jedan ili više servisa tog okruženja. U svakom od tih okruženja osoba se pojavljuje kao subjekt koji ima jedinstveni identitet. Sa tog stanovišta može se reći da subjekt može da kontroliše nivo nepovezanosti među svojim identitetima koje zahtevaju servisi prilikom procesa identifikacije. Često se javlja potreba da se ti identiteti povežu. eUprava omogućuje komunikaciju između državnog organa ili agencije i pojedinca pri čemu se eUprava ne pojavljuje kao davalac usluge emaila. Ovakve situacije se rešavaju uvođenjem federacije identiteta. Upravljanje federativnim identitetima je način upravljanja identitetima koji omogućuje subjektu da uspostavi veze među svojim identitetima, gde se svaki od njih može koristiti za različite servise prelazeći geografske i organizacione granice. Proces uspostavljanja veza između identiteta subjekta se naziva federacija identiteta [27]. Upravljanje federativnim identitetima postaje veoma važna kako ljudima, organizacijama tako i društvu jer je sve veća potreba interakcije i kolaboracije među njima na globalnom nivou. Takođe važan aspekt u upravljanju federativnim identitetima predstavlja zaštita privatnosti zbog širenja upotrebe socijalnih mreža.

2.1.2 Sistemi za upravljanje identitetima

Prve implementacije sistema za upravljanje identitetima su nastale sa pojavom prvih višekorisničkih sistema. Osnovna uloga sistema za upravljanje identitetima je da obezbedi siguran pristup operativnom sistemu. Potreba za identifikacijom korisnika na sistemu je nastala iz potrebe zaštite korisničkih i sistemskih podataka tokom izvršavanja programa u računaru na nivou operativnog sistema [28]. Kontrola pristupa je obezbeđivala sprečavanje neautorizovanog pristupa sistemu i resursima. Mehanizmi kontrole pristupa definisali su kako subjekat može da pristupi nekom objektu, odnosno kako proces može da pristupi nekom resursu. Sistem za upravljanje identitetima je sve atribute jednog identiteta čuvao u

repozitorijumu koji se nalazio u lokalnom skladištu podataka. Za svaku osobu podaci o identitetu su se čuvali na lokalnom skladištu podataka. Svaka osoba je imala onoliko identiteta na koliko je računara imala pristup. Takođe svi ostali servisi koji su se izvršavali na jednom računaru su imali svoje lokalne identitete, čak iako su resursi koje koriste bili deljivi među računarima.

Pojavom lokalnih računarskih mreža i protokola koji su nastali iz IEEE 802.x standarda pojavila se potreba za promenama u sistemu za upravljanje identitetima. Potrebe korporativnih mrežnih okruženja za centralizovanim udaljenim upravljanjem operativnim sistemom su promenile lokaciju čuvanja atributa identiteta na lokalnim skladištima i otvorile su put ka mrežnim repozitorijumima identiteta. Uvođenjem polisa i uloga omogućeno je jednoobrazno upravljanje identitetima na nivou jedne lokalne računarske mreže. Sa ovim izmenama sistemu za upravljanje identitetima mogu se definisati sledeće uloge aktera:

- **Subjekti** - su svi entiteti čiji se atributi digitalno zapisuju i koriste u transakcijama. Postoji veliki broj mogućih atributa jednog identiteta i mogu se klasifikovati na sledeći način:
  - *Atributi zasnovani na dokumentima*: Vrednosti ovih atributa se dobijaju iz dokumenata koje izdaju insititucije jedne države: pasoš, izvod iz matične knjige rođenih, lična karta. Vrednosti atributa mogu biti broj pasoša, broj čine karte, jedinstveni matični broj, PIB.
  - *Demografski atributi*: Ovi atributi sadrže informacije o polu, starosti, državi porekla, državi stanovanja, adresi rođenja, adresi stanovanja.
  - *Finansijski atributi*: Ove atribute izdaju finansijske institucije, kao što su banke, i uključuju broj kreditne kartice, broj bankovnog računa, kreditnu zaduženost.
  - *Biometrijski atributi*: Ovi atributi sadrže karakteristike identiteta: otisak prsta, sliku lica, zapis govora, zapis načina hoda.
  - *Transakcioni atributi*: vrednosti ovih atributa su veoma dinamični i karakteriše ih interakcija između subjekata posredstvom komunikacione infrastrukture. Računi o plaćenim proizvodima ili uslugama, informacije o novčanim transakcijama između privrednih subjekata su neki od primera ovih atributa.

Za subjekte je zaštita privatnosti vrednosti ovih atributa od velikog značaja.

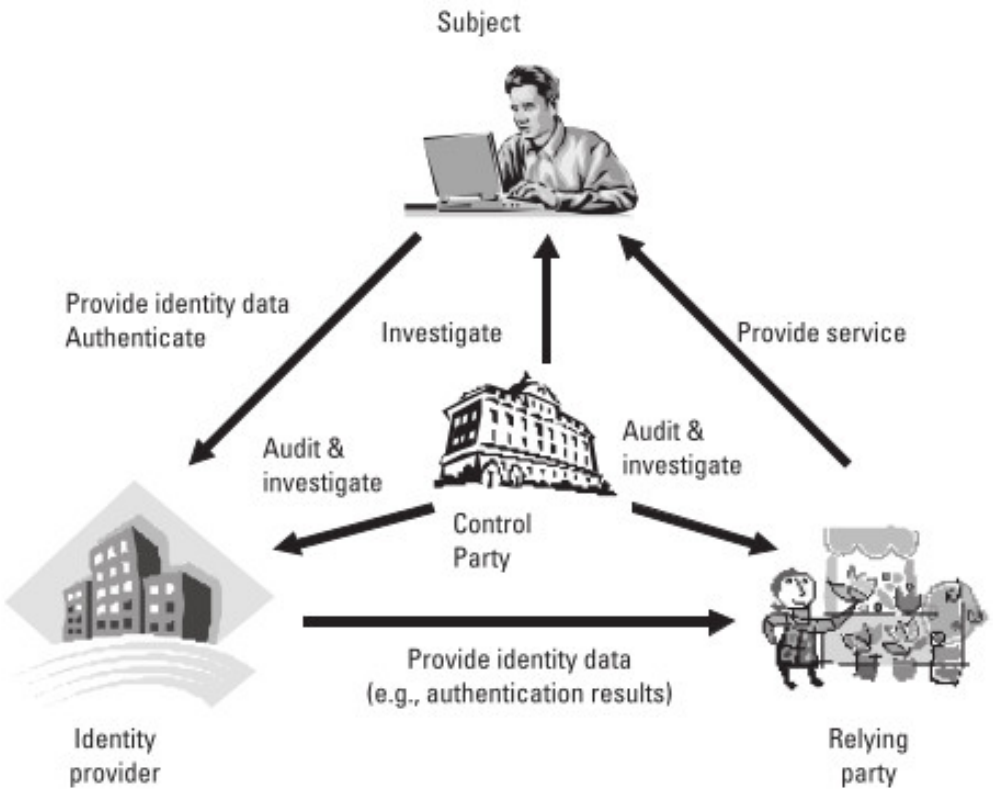
- **Provajderi identiteta** - su davaoci usluga identiteta subjektima. Oni izvršavaju 4 osnovna zadatka:

  1. Generisanje i dodelu specifičnih atributa subjekta identitetu
  2. Povezuju atribut identiteta jednog subjekta sa ostalim atributima identiteta istog subjekta
  3. Potvrđuju atribute identiteta
  4. Obezbeđuju kredencijale zapisane u atributima identiteta i neporecivost vrednosti atributa

  Potrebno je napomenuti da provajder identiteta može povezati vrednosti atributa identiteta sa atributima identiteta koje obezbeđuje neki drugi projader indentiteta.

- **Zastupnici** - obezbeđuju servise korisnicima ili agentima u ime korisnika ili pristupaju resursima na osnovu zahteva autentifikovanog korisnika. Trebaju da odrede nivo poverenja kredencijalima predatim od strane korisnika zahtevaoca kao skup atributa kojima se može pristupiti
- **Kontrolori** - su regulatorna tela koja mogu da imaju pristup informacijama identiteta. Njihov osnovni zadatak je prikupljanje informacija o aktivnostima i transakcijma identiteta.

Akteri mogu da imaju više uloga istovremeno. Na primer, akter može istovremeno da ima uloge i provajdera identiteta i zastupnika. Subjekat može da bude provajder identiteta i svoje identitete. Subjekti mogu direktno da komuniciraju sa provajderima identiteta i zastupnicima tokom transakcija identiteta [2].



Slika 1. Tipovi aktera u sistemu za upravljanje identitetima [2]

3 Multibiometrijski sistemi

3.1 Biometrijski sistemi

3.1.1 Biometrijski modaliteti

Postoje raličiti biometrijski identifikatori koji se mogu koristiti u biometrijskom sistemu pomoću kojih je moguće utvrditi identitet jedne osobe, kao što je navedeno u tabeli 1.

- **Otisak prsta** : Predstavlja šablon u kome su utisnuti grebeni i doline u epidermis prsta. Topologija epidermisa prsta se formira na osnovu genetskog materijala i faktora sredine. Genetski kod zapisan u lancima DNK daje opšta uputstva o načinu na koji se formira koža tokom razvoja fetusa, a specifičan način na koji se formira je skup slučajnih događaja (položaj fetusa u materici u određenom trenutku, sastav i gustina amniotske tečnosti). Zbog toga čak i otisci prstiju identičnih blizanaca bivaju različiti nakon rođenja. Otisci prstiju se formiraju nakon sedam meseci razvoja fetusa i topologija grebena i dolina se ne menja u toku života osobe. Na topologiju epidermisa prsta utiču faktori sredine koji mogu dovesti do ozlede prsta i promeniti izgled brebena i dolina. Otisak prsta ima status biometrijskog modaliteta visikog značaja usled predočenih činjenica [29].

  Od 19. veka pa do danas se koristilo više različitih tehnika uzimanja otiska prsta. “*Mastilo-tehnika*” se još uvek koristi. Međutim danas se prevashodno koriste različiti elektronski skeneri. Najvažni deo svakog skenera predstavlja senzor a pripadaju jednoj od sledeće tri grupe senzora: optički, silikonski ili ultrazvučni. Preciznost biometrijskih sistema zasnovanih na otisku prsta je na visokom nivou tako da otisak prsta spada u pozdane biometrijske modalitete. Nijedan algoritam za prepoznavanje nema apsolutnu preciznost te je za utvrđivanje performansi algoritama i sistema potrebno vršiti odgovarajuća testiranja. Evaluacija, u generalnom slučaju, može biti tehnološka, scenarijska ili operativna.

  Tehnološkom evaluacijom se testiraju algoritmi koji zadovoljavaju predefinisane zahteve nad određenim, predefinisanim podacima. Prednost tehnološke evaluacije jeste mogućnost da se dobijeni rezultat reprodukuje. Aktuelna evaluacija ovog tipa je NIST-ov “Fingerprint Vendor Technology

Evaluation" [30]. Prilikom scenarijskog i operativnog testiranja pored testiranja nad realnim podacima uzimaju se u obzir i senzori kao i celokupna aplikacija. U ovim načinima testiranja nije moguće ponoviti dobijene rezultate.

Otisak prsta ima široke mogućnosti primene. Koristi se za kontrolu pristupa objektima, uređajima, autentikaciju različitih vrsta transakcija, zaštitu ličnih dokumenata, kao i brojne druge svrhe. Upotreba u realnim biometrijskim sistemima potvrđuje činjenicu da je otisak prsta trenutno najkorišćeniji biometrijski modalitet i verovatno će tu poziciju zadržati i u budućnosti.

- **Geometrija šake** : Biometrijski sistemi zasnovani na prepoznavanju geometrije šake se zasnivaju na većem broju merenja uzetih sa ljudske šake. Ta merenja uključuju oblik šake, veličinu dlana i dužinu i širinu prstiju [31]. Tehnika uzimanja geometrije šake je veoma jednostavna, relativno lako se koristi i nije skupa. Faktori okruženja, kao što su suvi vazduh, suva koža ili druge individialne anomalije ne utiču na preciznost prepoznavanja sistema zassnovanih na geometriji šake. Na mnogo mesta širom sveta su instalirani komercijalni sistemi za prepoznavanje geometrije šake. Ipak, kao biometrijski modalitet geometrija šake nema dovoljno različitih karakteristika tako da takvi sistemi ne mogu biti skalirani za prepoznavanje identiteta pojednica veće populacije. Geometrija šake ne mora da se menja tokom perioda odrastanja osobe. Neke od bolesti mogu uticati na geometriju šake, artritis na primer, i otežati izdvajanje korektnih informacija. Takođe, fizička veličina mernih uređaja sistema za prepoznavanje geometrije šake onemogućuje njihovu primenu u nekim uređajima, laptopovi i drugi prenosni uređaji. Postoje sistemi za autentifikaciju, koji su dostupni danas, koji se zasnivaju na merenjima samo nekoliko prstiju umesto cele šake. Njihovi merni instrumenti su mnogo manji od onih koji se upotrebljavaju za merenje geometrije cele šake.
- **Otiska dlana** : Dlan kod ljudi sadrži grebene i doline slično onima koji se javljaju kod otiska prsta. Površina dlana je mnogo veća od površine prsta pa se očekivalo da će biti krakteristčniji nego otisci prstiju [32]. Senzori koji se koriste za uzimanje otiska dlana su glomazniji i skuplji od onih koji služe za utimanje otiska prsta jer moraju da snime mnogo veću površinu. Pored grebena i dolina ljudski dlan poseduje dlanene linije kao i bore koje mogu biti snimljene i sa senzorima niže rezolucije. Kada se koriste senzori visoke rezolucije za dlan, sve karakteristike dlana kao što su geometrija, grebeni i

doline, dlanene linije i bore se mogu kombinovati radi veće preciznosti biometrijskog sistema.

- **Glas** : Osobine nečijeg glasa zavise od oblika i veličine vokalnog trakta, usta, nazalnih šupljina i usana. koji se koriste u sintezi govor. Time se glas kvalifikuje kao i fiziološka i bihejvioristička biometrijska karakteristika [33]. Navedene fiziološke karakteristike su nepromenjlive za neku osobu. Bihejviorisički deo govora menja se tokom vremena usled starenja, medicinskih uslova, emocionalnog stanja, itd. Glas nije preterano karakterističan te nije pogodan za identifikaciju u širem obimu. Biometrijski sistemi za prepoznavanje govora se mogu podeliti na : sisteme zavisne od teksta koji se izgovara i sisteme nezavisne od teksta. Sistemi zasnovani prepoznavanju govora sa tekstom koji se izgovara koriste unapred određene fraze. Kod sistema za prepoznavanje govora nezavisnih od teksta koji se izgovara nude veću zaštitu od prevare. Teži su za projektovanje dok se mane ogledaju u osetljivosti na pozadinski šum i ostale faktore koji dovode do degadacije uzorka.

- **Lice** : Prilikom prepoznavanja identiteta drugih ljudi na osnovu crta lica donosimo odluku o identitetu. Proces prepoznavanja, čak i u nepovoljnim vizualnim uslovima, obavljamo brzo i lako. Prepoznavanje lica uz pomoć računara je oblast koja je još uvek u fazi razvoja i suočava se sa brojnim izazovima koje tek treba rešiti. U laboratorijskim uslovima, bez spoljnih smetnji moderni biometrijski sistemi za identifikaciju osoba na osnovu lica gotovo da i ne prave greške. Problemi kao što su promene u osvetljenju, uglu posmatranja, okluzija imaju značajan uticaj na performanse ovih biometrijskih sistema.

  U literaturi se mogu naći različiti pristupi biometrijskom prepoznavanju lica [34]. Neki od algoritama posmatraju lice kao celinu i porede ga sa odgovarajućim šablonima u biometrijskoj bazi. Najpopularniji algoritmi ovog tipa su implementirani u okviru softverskih biblioteka otvorenog koda [35]. Alternativni pristup posmatranju lica kao celine jeste korišćenje informacija sa delova lica kao što su brada, nos, usne, jagodice [36]. Merenjem njhovih dimenzija, međusobne udaljenosti i uglova izračunava se skup vrednosti koje karakteršu određano lice. U navedenim kategorijama algoritama se koriste dvodimenzionalne slike. Danas postoje i pristupi koji koriste trodimenionalni prikaz ljudskog lica [37].

Kao biometrijski modalitet lice se često koristi u biometrijskim sistemima. Oblasti primene su raznovrsne. Društvene mreže nude atuomatsko tagovanje osoba na slikama. Noviji modeli pametnih telefona imaju mogućnost korišćenja lica radi otključavanja i zaključavanja telefona. Veliki broj komercijalnih sistema za kontrolu pristupa određenim objektima koristi lice kao autentifikacionu metodu.

- **Termogram Lica** : Slika generisana usled toplote koju zrači ljudsko telo je takođe karakteristika neke osobe i može da se dobije pomoću infracrvene kamere na neivazivni način, slično regularnoj (u vidljivom spektru) fotografiji. Tehnologija može da se koristi za prikriveno prepoznavanje. Sistemi bazirani na termogramu ne zahtevaju kontakt, ali je dobijanje slike otežano u nekontrolisanim uslovima kada se u okolini tela nalaze izvori toplote.

- **Iris** : Iris je kružno područje ograničeno zenicom i beonjačom sa obe strane. Tekstura irisa se formira tokom razvoja fetusa i stabilizuje u prve dve godine života. Međutim, pigmentacija oka nastavlja da se menja tokom dužeg vremenskog perioda. Kompleksnost teksture irisa ima veoma karakteritične informacije pogodne za identifikaciju osobe [38]. Tekući biometrijski sistemi bazirani na prepoznavanju irisa imaju obećavajuću preciznost i brzinu. Takođe omogućavaju podršku za velike indentifikacione sisteme bazirane na prepoznavanju irisa. Iris svake osobe je različit čak i kod identičnih blizanaca. Detekcija veštačkog irisa je jednostavna [39]. Rani biometrijski sistemi zasnovani na prepoznavanju irisa su bili značajno skupi i zahtevali su saradju korisnika, noviji sistemi jefitiniji i lakši za korištenje [40, 41].

- **Hod** : Hod je bihejvioralna biometrijska karakteristika i on odražava ponašanje osobe tokom hodanja. Jedna je od retkih biometrijskih karakteristika koje se može iskoristiti za prepoznavanje osobe sa distance. Osim toga, ova osobina je veoma pogodna u situacijama kada je potrebno tajno pratiti osobu. Većina algoritama za prepoznavanje pokušava da dobije ljudsku siluetu da bi mogao da izvuče prostono-vremenske atribute osobe koja se kreće. Izbor dobrog modela koji će reprezentovati ljudsko telo je ključan za efikasno funkcionisanje prepoznavanja hoda u biometrijskom sistemu. Drugi tip algoritama primenjuje skup dinamički izvedenih pokretnih tačaka na telu osbe koja se kreće i prati njihove pozicije tokom vremena da bi opisao hodanje jedne osobe [42]. Na hod jedne osobe utiče niz faktora kao što su izbor obuće, površina po kojoj hoda, odeća, bol u nogama itd.

- **Kucanje na tastaturi** : Svaka osoba koristi tastaturu računara na karakterističan i svojstvrn način. Od ove biometrijska karakteristike se ne očekuje da bude jedinstvena za svaku osobu, ali može dati dovoljne informacije za verifikaciju identiteta [43]. Dinamika kucanja je više bihejvioristička biometrijska karakteristika jer se način kucanja neke osobe menja usled promene emocionalnih stanja, pozicije u odnosu na tastaturu, tipa tastature, itd. Kucanje korisnika se može lako pratiti i omogućava konstantnu verifikaciju identiteta tokom sesije posle indentifikacije na osnovu "jačih" biometrijskih karakteristika poput otiska prsta ili irisa.
- **Potpis** : Način na koji se neka osoba potpisuje je karakteristika te osobe. Takođe, potpisivanje zahteva kontakt sa sredstvom za pisanje i napor od strane osobe. Potpis je bihejvioralna biometrijska karakteristika koja se menja tokom vremena. Takođe utiču fizička i emocijalna stanja osobe. Potpis je od strane državne uprave, pravnog sistema i komercijalnih subjekata prihvaćen kao metod autentifikacije [44, 45]. Sa porastom broja mobilnih uređaja kao što su PDA i tableti, potpis se može kao biometrijska karakteristika na uređajima tog tipa. Kod nekih osoba uzastopni potpisi se mogu značajno razlikovati. Osim toga profesionalni falsifikatori mogu reprodukovati potpise tako da prevare sistem za verifikaciju biometrijskog sistema [46].

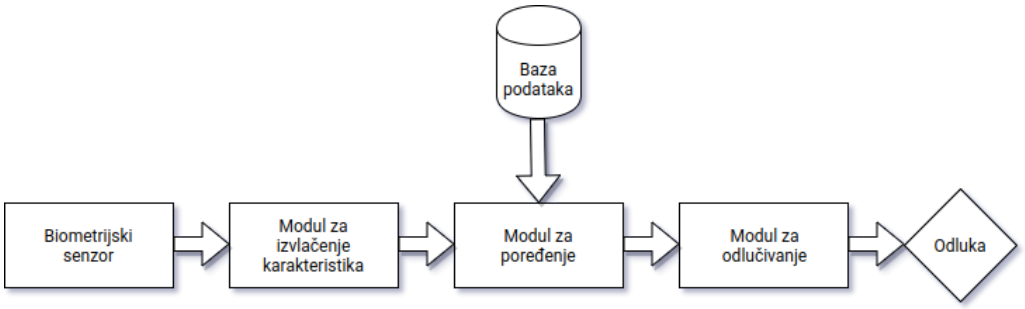
3.1.2 Načini rada biometrijskog sistema

Biometrijska autentifikacija ili biometrija, nudi jednostavan i pouzdan način rešavanja problema utvrđivanja identiteta osobe na osnovu "nečega što je ona" [31]. Biometrijske osobine jednoznačno povezuju osobu i njen identitet i te osobine se ne mogu lako izgubiti ili zaboraviti ili podeliti ili ukrasti ili krivotvoriti. Prilikom biometrijske autentifikacije subjekt mora biti prisutan za vreme autentifikacije. Nije moguće davanje lažnih osobina identiteta. Takođe, biometrija može koristiti funkcionalnost negativne identifikacije. Njen cilj je utvrđivanje da li je osoba upisana u sistem iako ta osoba tu tvrdnju osporava. Iz tih razloga sve je veća primena biometrijskih sistema u bezbedonosnim sistemima, državnim ili civilnim, kao zamena ili dopuna postojećim sistemima za pristup. Neki od sistema velikih razmera u kojima se koriste biometrijski sistemi su "Integrated Automated Fingerprint Identification System (IAFIS)" u Federalnom Istražnom Birou (FBI) [47], US-VISIT IDENT program [48], Schiphol Privium scheme" pri aerodromu Schiphol u Amsterdamu [49].

Tipični biometrijski sistem se sastoji od više različitih komponenti [1]:

1. **Biometrijski senzor** - Senzori su uređaji putem kojih se prikupljaju biometrijski uzorci jedne osobe. Ponekad se koriste algoritmi procene kvaliteta biometrijskog uzorka i utvrđuje da li je uzorak dovoljno dobrog kvaliteta i prepustio na dalju obradu ostalim komponentama. Na primer, prilikom uzimanja otiska prsta, biometrijski senzor registruje grebene i udubljenja otiska prsta i pamti ih kao sliku.
2. **Modul za izvlačenje karakteristika** - prikuplja karakteristike uzetog biometrijskog uzorka i formira *skup karakteristika* koji predstavlja novu reprezentaciju biometrijske osobine. U gore pomenutom primeru uzimanja otiska prsta, ovaj modul izvlači karakteristike minucija, odnosno njihove orijentacije i pozicije sa slike otiska prsta. Ovaj skup karakteristika dobijen sa slike otiska prsta će se sačuvati u bazi podataka biometrijskog sistema kao *šablon*. Šablon bi trebao da bude jedinstven za svaku osobu (ekstremno mala sličnost među različitim osobama). Takođe, šablon bi trebao biti nepromenljiv u odnosu na promene različitih uzoraka iste biometrijske osobine sakupljene od iste osobe (ekstremno mala promenljivost među uzorcima jedne osobe).
3. **Modul za poređenje** - poredi skup karakteristika dobijen od modula za izvlačenje karakteristika sa odgovarajućim šablonima koji se nalaze u bazi podataka. Kao rezultat rada ovog module se dobijaju skorovi poređenja ili drugačije rečeno stepen sličnosti odnosno različitosti dobijenog skupa karakteristika i šablona. Ako posmatramo primer otiska prsta, modul za poređenje će vratiti broj minucija koje se poklapaju.
4. **Modul za odlučivanje** - donosi odluku na osnovu rezultata (skora poređenja/stepana sličnosti ili različitosti) koji dobije od modula za poređenje. Ovaj modul može da radi u dva režima rada:
    - verifikacioni režim rada - u ovom režimu rada, na osnovu skora poređenja odlučuje da li da prihvati ili odbaci predloženi identitet
    - identifikacioni režim rada - u ovom režimu rada, odlučuje koji od identiteta je najverovatniji ili daje skup identiteta koji imaju iste ili približno iste verovatnoće.
5. **Baza podataka** - sadrži sve prikupljene šablone svih registrovanih korisnika. Šabloni su generisani tokom upisa korisnika na osnovu uzetih biometrijskih

uzoraka. Sirovi podaci uzeti sa biometrijskog senzora se mogu čuvati u bazi podataka ili informacije o njihovoj lokaciji kada se čuvaju van baze podataka.

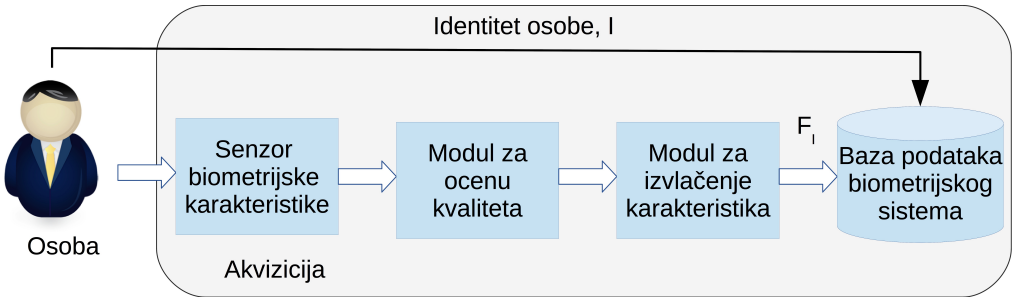


Slika 2. Arhitektura biometrijskog sistema

Nad datom arhitekturom prikazanoj na slici 2 se mogu definisati načini funkcionisanja biometrijskog sistema:

- Akvizicija
- Verifikacija
- Identifikacija

Akvizicija biometrijskih uzoraka se korišćenjem odgovarajućeg senzora za željenu biometrijsku karakteristiku vrši transformacijom biometrijskog uzorka u šablon koji će se sačuvati u bazi podataka biometrijskog sistema. Pre same transformacije se proverava kvalitet uzetog biometrijskog uzorka i ukoliko nije zadovoljavajući, ponovo se upotrebom senzora uzima biometrijska karakteristika. Dobijeni šablon se vezuje za odgovarajuću osobu i tip uzete biometrijske karakteristike.



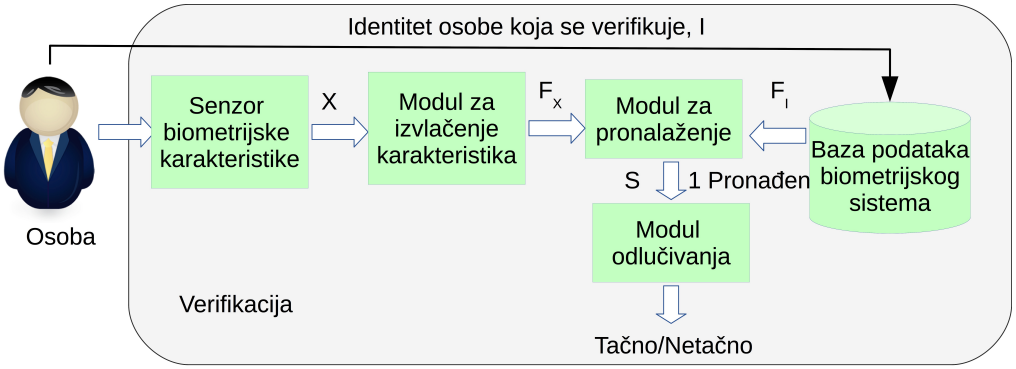
Slika 3. Akvizicija biometrijskog uzorka

Prilikom verifikacije uzeti biometrijski uzorak se upoređuje sa šablonom koji odgovara osobi za koju se proverava identitet. Kada uzeti biometrijski uzorak i šablon imaju veliki stepen sličnosti tada biometrijski sistem prihavata uzeti uzorak kao autentičan. U protivnom uzeti biometrijski uzorak se proglašava kao ne validan. Verifikacija se može formalno prikazati kao problem klasifikacije gde je potrebno

utvrditi da li je uređeni par identiteta i skupa karakteristika I, F_x autentičan. Ako sa F_I obeležimo sačuvani šablon koji odgovara identitetu I i sa S stepen sličnosti između F_x i F_I onda se pravilo po kome se odlučuje može dati na sledeći način:

$$(I, F_x) \in \begin{cases} \text{autentičan}, & \text{ako je } S \geq \delta \\ \text{lažan}, & \text{ako je } S < \delta \end{cases} \quad (1)$$

gde je δ prag tolerancije. Takođe, moguće je koristi stepen različitosti. U tom slučaju nejednačine u pravilu odlučivanja predstavljene u formuli 1 treba invertovati.



Slika 4. Verifikacija identiteta na osnovu uzetog uzorka

Identifikacija može biti pozitivna i negativna. Pozitivna identifikacija označava slučaj kada je u biometrijskom sistemu korisniku utvrđen identitet bez njegove ranije tvrdnje da poseduje identitet u sistemu. Pozitivna identifikacija daje odgovor na pitanje "Da li si ti neko ko je poznat sistemu?". Negativna identifikacija je poznata kao proveravanje ili trijaža i koristi se kada korisnik prikriva svoj pravi identitet. U ovom slučaju biometrijski sistem daje odgovor na pitanje "Ti si neko poznat sistemu a rekao si da nisi?". Često se koristi na aerodromima za identifikaciju putnika čiji se identiteti nalaze na poternicama. Negativna identifikacija se može koristiti u prevenciji izdavanja različitih ličnih dokumenata istoj osobi, takođe je i pogodna prilikom onemogućavanja da ista osoba pod različitim imenima protiv pravno prisvaja dobit. Bez obzira da li se radilo o pozitivnok ili negativnoj indetifikaciji biometrijski sistem uzeti biometrijski uzorak upoređuje sa šablonima svih osoba upisanih u bazu podataka tog biometrijskog sistema, čak iako za dati biometrijski uzorak postoji identitet osobe koji čiji šablon ima najveći stepen sličnosti ili za identitet osobe ne postoji zapis o uzetom bilo kakvom biometrijskom uzorku.

Ako sa F_x obeležimo skup karakteristika za koji utvrđujemo identitet, sa I identitet za koji se vrši identifikacija, a koji pripada skupu N identiteta sačuvanih u bazi podataka

datog biometrijskog sistema

$$I \in \{I_1, I_2, \ldots I_n, I_{n+1}\}$$

, I_{n+1} označava identitet za koji se ne može utvrditi da postoji u biometrijskom sistemu za uzeti biometrijski uzorak. Tada, za sačuvan šablon F_{I_n} koji odgovara identitetu I_n se može naći stepen sličnosti S_n između F_x i F_{I_n} pri čemu

$$n = 1, 2, \ldots, N$$

, se može dati pravilo po kome se odlučuje kao:

$$F_x \in \begin{cases} I_{n_0}, & \text{ako je } n_0 = arg\max_n S_n \text{ i } S_{n_0} \geq \delta \\ I_{N+1}, & \text{inače,} \end{cases} \quad (2)$$

gde je δ prag tolerancije.

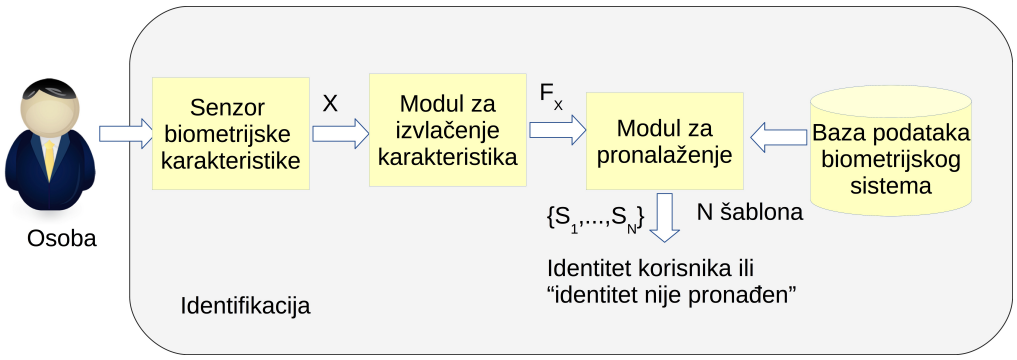
U praktičnoj primeni identifikacija je polu automatska, odnosno biometrijski sistem kao rezultat vraća

$$m$$

identiteta sa najvećim stepenom sličnosti

$$1 < m < N$$

a potom ekspert ručno odabira identitet koji najbolje odgovara zadatom uzorku. Za biometrijske sisteme sa velikim brojem identiteta identifikacija predstavlja značajno veći izazov od verifikacije.



Slika 5. Identifikacija na osnovu uzetog uzorka

Prilikom utvrđivanja identiteta većina biometrijskih sistema koji su u upotrebi danas obično koriste jednu biometrisku osobinu, te ih možemo nazvati unibiometrijskim

ili unimodalnim sistemima. Iako je se upotreba unimodalnih u raznim sitauacijama pokazala uspešnom to ne znači da nema mesta za daljim razvojem i istraživanjima biometrijskih tehnologija. Do sada su registrovani neki od problema koji se javljaju prilikom korišćenja unibiometrijskih sistema [14] :

1. **Šum u ulaznim podacima** - Biometrijski podaci koji dolaze u biometrijski sistem sa senzora mogu sadržati šum usled nesavršenih uslova pod kojima se uzimaju. Na primer, ožiljak na prstu, mutiranje glasa, nepovoljni uslovi osvetljenja mogu značajno da utiču na sliku lica ili dužice oka. Odbijanje legitimnih korisnika se javlja kao posledica izvitoperenih ulaznih podataka dobijenih od senzora.

2. **Neuniverzalnost** - Biometrijski sistem ne može da uzorkuje smislene biometrijske podatke i time onemogućiti akviziciju i upisivanje korisnika u biometrijski sistem. Sistem koji radi sa otiscima prstiju neće moći da izdvoji minucije usled lošeg kvaliteta otisaka prstiju, sistem za prepoznavanje dužice oka usled dugih trepavica, suzavosti oka ili nekih patoloških stanja oka neće moći da prikupe inicijalne podatke o osobi, na primer. Biće potrebno dodatno obraditi takve biometrijske uzorke da bi te osobe mogle da se uključe u biometrijski sistem.

3. **Gornja granica tačnosti identifikacije** - Poboljšavanje performansi unibiometrijskog sistema se ne može stalno unapređivati usavršavanjem modula za izvlačenje karakteristika i modula za poređenje. Postoji konačan skup različitih šablona u kojima se smešta biometrijski skup karakteristika. Kapacitet šablona je ograničen uočenim brojem varijacija su skupu karakteristika svake osobe i varijacijama između skupova karakteristika različitih osoba.

4. **Lažno predstavljanje** - Zlonamerna osoba može pokušati da lažira biometrijski modalitet kako bi se lažno predstavio biometrijskom sistemu. Lažno predstavljanje mimikrijom glasa [50] i potpisa [51] predstavljaju ranjivost unibiometrijskih sistema. Takođe upotreba sintetičkih materijala se koristi u lažnom predstavljanju unibiometrijskih sistema zaasnovanih na otisku prsta [52, 53]. Ciljani napadi mogu da ugroze bezbednost koju obezbeđuje biometrijski sistem a samim tim umanji efekte koje daje upotreba biometrijskih sistema u domenu zaštite pristupa [54].

3.1.3 Multibiometrijski sistemi

Prethodno navedena ograničenja se mogu prevazoći integracijom više biometrijskih izvora. Integracija više biometrijskih izvora se može postići fuzijom, na primer, višestruke osobine jedne osobe ili višetrukih modula za izvlačenje karakteristika i algoritama poređenja koji rade nad istom biometrijom. Takvi sistemi su poznati kao multibiometrijski sistemi [55, 56, 57, 58] i mogu da poboljšaju tačnost modula za poređenje posebno sa povećanjem broja identiteta i biometrijskih uzoraka u bazi podataka biometrijskog sistema. Prednosti multibiometrijskih sistema nad unibiometrijskim sistemima su [58] :

- Multibiometrijski sistemi povećavaju fleksibilnost sistema prilikom upisa korisnika u biometrijski sistem korišćenjem nekoliko različitih biometrijskih osobina (na primer, lice, glas, otisak prsta, dužice, ruke) iako će se zahtevati samo podskup biometrijskih osobina (na primer, lice i glas) prilikom provere identiteta. Ovakvim pristupom se povećava stepen univerzalnosti sistema prilikom akvizicije te je moguće povećati broj osoba koje se mogu upisati u bazu podataka biometrijskog sistema.
- Omogućuju filtriranje ili indeksiranje velikih baza podataka. Na primer, u bimodalnom sistemu koji radi nad otiskom prsta i slikom lica, skup karakteristika lica se može iskoristiti za sračunavanje vrednosti indeksa za dobijanje liste identiteta potencijalnih kandidata. Biometrijski modalitet otiska prsta se tada koristi za utvrđivanje konačnog identiteta iz liste potencijalnih kandidata.
- Postaje sve teže, ako ne i nemoguće, lažno predstavljanje za zlonamerne osobe. Svaki podsistem daje stepen sličnosti za pojedinačni biometrijski modalitet, a zatim po odgovarajućoj šemi fuzije modul za odlučivanje određuje da li je korisnik uljez ili ne. Pored toga, od korisnika se traži da prikaže sistemu slučajan podskup biometrijskih osobina u trenutku predstavljanja sistemu. U interakciji sa sistemom na zasnovanom na mehanizmu pitanja i odgovora multibiometrijski sistem može zaključiti da li je ispred njega živa osoba. Potrebno je napomenuti da je moguće primeniti mehanizam pitanja i odgovora i u unibiometrijskom sistemu, na primer: “Molim te reci, moj glas je moj pasoš”, “Trepni te dva puta i pomerite oči na desno”, itd.
- Multibiometrijski sistemi se uspešno nose sa problemom šuma u ulaznim podacima. Kada biometrijski podaci uzeti sa senzora sadrže šum, dostupnost

druge biometrijske osobine može pomoći u pouzdanom određivanju identiteta. Neki sistemi uzimaju u obzir i kvalitet pojedinačnog biometrijskog uzorka u procesu fuzije. Ovo je veoma važno kada se uzimanje biometrijskog uzorka odvija u veoma lošim uslovima koji ne dozvoljavaju pouzdano uzimanje uzorka. Na primer, ako postoji ambijentalni akustički šum koji mikrofonu onemogućuje da kvalitetno zabeleži glas osobe, slika lica se može iskoristiti za autentifikaciju. Procenom kvaliteta biometrijskih podataka se dodatno poboljšati kvalitet rada multibiometrijskog sistema.

- U praćenjenju osobe ili konstatnom monitoringu neke osobe multibiometrijski sistemi mogu biti od velike koristi. Ako biometrijski sistem koristi 2D kamere za snimanje lica i hoda osoba koje prolaze u zavisnosti od daljine osobe u odnosu na kameru, obe karakteristike ne moraju biti dostupne simultano. Jedna ili druga ili obe osobine se mogu koristiti u zavisnosti od lokacije osobe u odnosu na sistem za prikupljanje biometrijskih podataka i tako omogućiti praćenje osobe. Na primer: na železničkim i autobuskim stanicama, aerodromima, stadionima ili bilo kojim drugim događajima gde je prisutna velika grupa ljudi a potrebno je pronaći i izdvojiti nepoželjne osobe.
- Za jedan biometrijski sistem se može reći da je sistem otporan na greške jer može nastaviti sa radom čak i slučaju otkaza nekog od biometrijskih izvora usled greške na senzoru ili softveru. Sistem otporan na greške je veoma koristan u velikim sistemima za autentifikaciju koji rade nad velikim skupom osoba, na primer u primeni na pasoškoj kontroli granice.

Na dizajn multibiometrijskog sistema utiče mnogo faktora, od izbora broja biometrijskih osobina, nivoa integracije informacija dobijenih višestrukih osobina koje će naći u biometrijskom sistemu, primenjenih metodologija za integraciju informacija do odnosa cene i performanse. Izbor broja biometrijskih osobina je pre svega vođen prirodom primene biometrijskog sistema. Korišćenje višestrukih osobina povećava količinu informacija koje je potrebno obraditi a time se povećavaju zahtevi za računarskom snagom i naravno cena koštanja tih računarskih resursa. Takođe je potrebno ustanoviti i korelaciju između osobina koje će se koristiti. Današnji mobilni uređaji su snabdeveni kamerom te je kombinacija biometrijskih osobina glasa i lica pogodnija za korišćnje kog ovog tipa uređaja. Kod ATM uređaja je jednostavnija za korištenje kombinacije otiska prsta i slike lica [59].

3.1.3.1 Taksonomija Na osnovu izvora biometrijskih informacija [14]. Multibiometrijski sistem se pre svega oslanja na podatke koji dostavljaju sistemu od

strane različitih izvora biometrijskih informacija. Na osnovu prirode tih izvora multibiometrijski sistem se može klasifikovati u jednu od sledećih kategorija [58]:

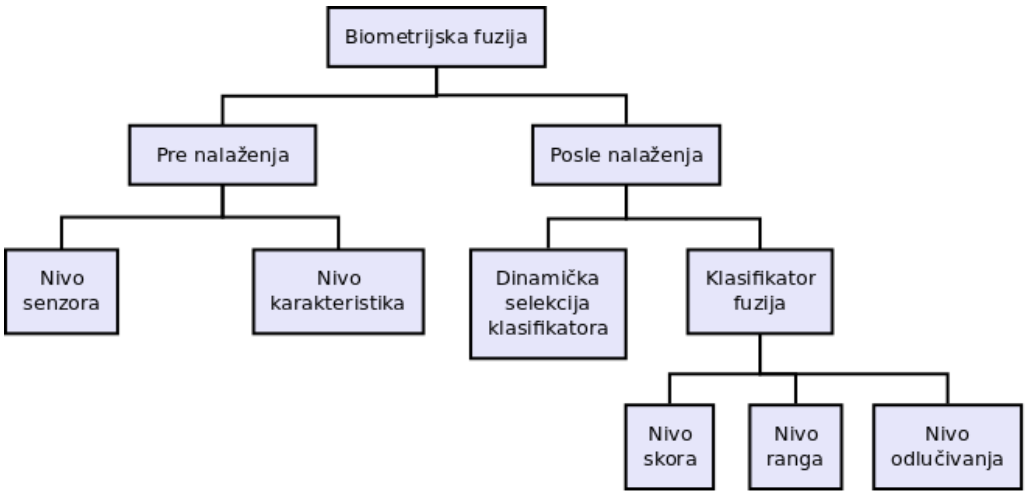
- **Multi-senzor sistemi** - Multi-senzor sistemi koriste više senzora za uzimanje jedne biometrijske osobine jedne osobe. Na primer, sistem za prepoznavanje lica može koristiti više 2D kamera za uzimanje slike lica [60], infracrveni senzor se može koristiti u sprezi sa senzorom vidljivog opsega svetlosti za uzimanje površinskih informacija lica jednog objekta [61, 62, 63], višespektralna kamera se može koristiti za uzimanje slike dužice oka, lica ili prsta [64, 65] ili optički i kapacitivni senzori se mogu koristiti za uzimanje slike otiska prsta [66]. Korišćenjem višestrukih senzora se mogu dobiti komplementarne informacije koje mogu poboljšati sposobnost prepoznavanja u sistemu. Zbog prirode osvetljenosti usled ambijentalnog svetla infracrvena i slika vidljivog dela sveslosti lica može dati različite nivoe informacija koje će poboljšati tačnost pronalaženja. Performanse sistema zasnovanog na 2D kamerama se može poboljšati korišćenjem informacija o telu dobijenih na osnovu slika sa 3D kamere.
- **Multi-algoritam sistemi** - Korišćenjem različitih algoritama za izvlačenje karakteristika i/ili algoritama za pronalaženje nad istim biometrijskim podacima može dovesti do boljih performansi pronalaženja. Zbog prirode multi-algoritam sistema konsolidacija rezultata algoritama za izvljačenje karakteristika je od velike važnosti. Takođe, za višetruke algoritme pronalaženja je važno da oni rade nad istim skupom karakteristika. Ovakvi sistemi ne zahtevaju dodavanje ili zamenu postojećih senzora koji se koriste u sistemu. Sa druge strane uvođenje novih modula za pronalaženje i/ili modula za izvlačenje karakteristika mogu da povećaju računarsku kompleksnost ovih sistema. Korišćenje minucija i informacija izvučenih iz teksture prsta sa slika otiska prsta je opisano u radu [67].
- **Multi-instance sistemi** - Kod ovih sistema se koriste višetruke instance iste biometrijske osobine osobe. Na primer korišćnje levog i desnog kažiprsta ili dužice levog i desnog oka se mogu koristiti za verifikaciju identiteta osobe [68, 69]. Ukoliko ovakav sistem koristi jedan senzor za uzimanje biometrijskih podataka različitih instanci jedne biometrijske osobine povećaće se vreme uzimanja biometrijskih podataka i dovesti do nelagodnosti za korisnika sistema. Sa druge strane ukoliko postoji potreba za simultanim uzimanjem više instanci jedne biometrijske osobine potrebno je koristiti

uređaje koji u sebi sadrže višestruke senzore za isti biometrijski modalitet što će dovesti do efikasnijeg uzorkovanja ali i više cene celokupnog sistema.

- **Multi-sample sistemi** - Jedan senzor se koristi za uzimanje višestrukih uzoraka iste biometrijske osobine sa ciljem da se uoče varijacije koje se pojavljuju u toj biometrijskoj osobini. U biometrijskom sistemu za prepoznavanje lica mogu se uzeti i sačuvati slike frontalnog, levog i desnog profila osobe da bi se uočile varijacije u izrazu lica. Kod sistema za preoznavanje otisaka prsta koji je opremljen malim senzorom može se uzeti niz parcijalnih slika jednog otiska prsta da bi se dobili različiti regioni otiska. Za dobijanje kompozitne slike otiska se može koristiti šema sklapanja mozaika od parcijalnih slika. Jedan od glavnih problema u multi-sample sistemima je određivanje broja uzoraka koji će biti uzeti od osobe. Ovakav pristup zahteva da se unapred uspostavi redosled radi optimizacije strategija integracije. Kod sistema za prepoznavanje lica se koriste slike fronatlnog, levog i desnog profila gde je unapred određeno da levi i desni profil trebaju da obuhvate tri četvrtine lica [70, 71].
- **Multimodalni sistemi** - Kod multimodalnih sistema identitet je vezan za postojanje više različitih biometrijskih osobina. U nekim od prvih multimodalnih biometrijskih sistema identiteti su zasnovani na karakteristikama lica i glasa [72, 56, 55]. U multimodalnim sistema se polazi od toga da će fizički nepovezane osobine (na primer, otisak prsta i dužica oka) dati bolje performanse sistema nego fizički povezane osobine (na primer, glas i pomeranje usni). Primena ovakvih sistema je bitno skuplja zbog većeg broja senzora pre svega, i razvoja odgovarajućeg korisničkog interfejsa. Preciznost identifikacije se može značajno poboljšati povećanjem broja osobina pomoću kojih se vrši identifikacija. Povećanje broja osobina, sa druge strane, povećava količinu podataka koju je potrebno obraditi prilikom identifikacije. Na broj osobina koje će se koristiti u specifčnoj primeni utiču razni faktori poput vremena potrebnog za upis novih korisnika u sistem, očekivani stepen greške, vreme obrade zahteva za identifikacijom, problemi sa navikama korisnika, itd.
- **Hibridni sistemi** - U literaturi se može naći pojam hibridnog sistema koji opisuje sistem koji integriše podskup prethodno nabrojanih sistema [73, 55].

Sanderson i Paliwal [74] uvode kategorizaciju nivoa fuzije kroz dve šire kategorije: fuzija pre poređenja i fuzija posle poređenja (Slika 6). Ovakva klasifikacija se pokazala neophodnom iz razloga velike količine podataka koje modul za poređenje mora da obradi. Radi poboljšanja performansi sistema pogodno je uraditi fuziju

informacija i redukovati količinu podataka koji se moraju obraditi. Korišćenje fuzije pre poređenja zahteva razvoj novih tehnika i algoritama jer moduli za poređenje koji se koriste u unimodalnim sistemima nisu više relavantni. Ovo može predstavljati poseban izazov u korišćenju multibiometrijskih sistema. Fuzija pre pronalaženja obuhvata fuziju na nivou senzora ili sirovih podataka i na skupu karakteristika. Fuzija posle pronalaženja uključuje fuziju na nivou skora ili stepena sličnosti, ranga i prilikom odlučivanja.



Slika 6. U biometrijskom sistemu fuzija se može postići na različitim nivoima

1. **Fuzija na nivou senzora** - predstavlja konsolidaciju sirovih podataka koristeći više senzora ili više snimaka istog biometrijskog uzorka sa jednog senzora. Sirovi biometrijski podaci presdstavljaju bogat izvor informacija o biometrijskom uzorku. Međutim često se dešava da zbog loših ambijentalnih uslova uzeti biometrijski uzorci sadrže šum koji može onemogućiti ispravno izvlačenje karakteristika [75, 76].
2. **Fuzija na nivou izvlačenja karakteristika** - predstavlja fuziju koja koristi više algoritama za izvlačenje karakteristika iz istog biometrijskog uzorka. Posle normalizacije, transformacije i redukcije se dobija skup karakteristika od pojedinačnih rezultata dobijenih od svakog algoritma ponaosob. Detekcija međusobnih povezanosti vrednosti karakteristika u različitim algoritmima može poboljšati tačnost prepoznavanja. Novo dobijeni skup karakteristika mora da se nalazi u vektorskom prostoru koji očekuje primenjena tehnika pronalaženja što se postiže upravo konsolidacijom skpa karakteristika [77, 78].
3. **Fuzija na nivou skora** - za dobijanje rezultata pronalaženja kombinuju rezultati više modula za pronalaženje. Karakteristika ove fuzije je da na

rezultujući vektor rezultata modula za pronalaženje mogu primeniti različiti aloritmi za dobijanje konačnog rezltata koji se predstavlja skalarno vrednošću. Metode koje se koriste na ovom nivou fuzije se mogu klasifikovati u više kategorija [58]: density-based schemes, transformation-based schemes i classifier-based schemes

4. **Fuzija na nivou ranga** - rezultat rada biometrijskog sistema u modu identifikacije se može predstaviti kao rang lista upisanih identiteta koji imaju neki stepen sličnosti sa uzorkom identiteta koji se pokušava pronaći. Cilj fuzije na nivou ranga je da konsoliduje rezultate pojedinačnih biometrijskih podsistema i omogući procesu donošenja odluke redukovanu listu pronađenih ideniteta. Ova rang lista treba da sadrži samo identitete sa najboljim skorom, odnosno sa najvećim stepenom sličnosti. Pogodnost ovog nivoa fuzije je postojanje interoperabilnosti između različitih biometrijskih sistema. Interoperabilnost u ovom slučaju znači da je moguće upoređivati rang liste bez upotrebe algoritama za normalizaciju [79].
5. **Fuzija na nivou odlučivanja** - većina komercijalnih biometrijskih sistema nudi pristup samo konačnom rezultatu. Ukoliko se multibiometrijski sistem gradi od skupa koemrcijalnih unibiometrijskih sistema jedina moguća fuzija je ona na nivou odlučivanja. Metode za fuziju na nivou odlučivanja preporučene u literaturi uključuju: „AND“ i „OR“ pravila [80], većinsko glasanje [60], ponderisano većinsko glasanje [81], Bajesovo zaključivanje [82],

4 Pregled i kritički osvrt na postojeće multibiometrijske sisteme

Rezultat izvršavanja multibiometrijskog sistema je predstavljen preko skalarne vrednosti koja kvantifikuje nivo sličnosti između uzetog biometrijskog uzorka i šablona koji je upamćen u bazi podataka multibiometrijskog sistema. Greške koje se javljaju u tom slučaju su:

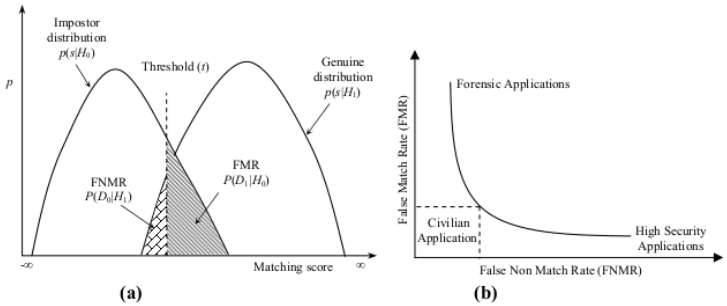
- ***False match*** : pogrešan zaključak da biometrijske karakteristike različitih osoba pripadaju istoj osobi
- ***False non match*** : pogrešan zaključak da biometrijske karakteristike iste osobe pripadaju različitim osobama

Ove greške se mogu nazvati kao lažno prihvatanje (eng. *false accept*) i lažno odbijanje (eng. *false reject*). Postoji međusobna zavisnost zavisnost ove dve vrste greški u svakom biometrijskom sistemu. Stepen lažnog prihvatanja (eng. *False Match Rate*-FMR) kao i stepen lažnog odbijanja(eng. *False Nonmatch Rate/–FNMR) funkcije su od praga tolerancije biometrijskog sistema. U slučaju da je prag niži, sistem je tolerantniji na različite varijacije i šum na ulazu, pa je stoga FNMR niži,a FMR viši. Ako povisimo prag tolerancije, FNMR biće viši usled manje tolerancije na šum, ali će zato FMR biti niži. Performanse biometrijskog sistema u zavisnosti od praga tolerancije predstavljaju se uz pomoć ROC (eng. /Receiver Operating Characteristics*) krivi. ROC kriva predstavlja grafik na cijim se osama nalaze vrednosti FMR i (1-FNMR).

Odlučivanje se vrši na sledeći način – ako je skor poređenja $S(X_Q, X_I)$ veći od praga tolerancije t tada odluči D_1 inače odluči D_0. U skladu sa prethodnim definicijama, možemo zaključiti da je FMR jednak verovatnoći greške prve vrste(hipotezu H0 odbacujemo, a ona je tačna), dok je FNMR jednak verovatnoći greške druge vrste(hipotezu H_0 prihvatamo, a ona nije tačna). Vrednost 1-FNMR predstavlja moć testa. Kako bi mogli da odredimo preciznost biometrijskog sistema potrebno je da odredimo raspodelu koje imaju skorovi poređenja biometrijskih karakteristika istih osoba (zakon verovatnoće $p(S(X_Q, X_I)|H_1))$), kao i raspodelu skorova poređenja biometrijskih karakteristika različitih osoba (zakon verovatnoće $p(S(X_Q, X_I)|H_0)$). Na Slici 7 grafički je prikazano izračunavanje kao i međusobna zavisnost FMR i FNMR koji se dobijaju prema sledećim formulama [1] :

$$FMR = \int_{t}^{\infty} p(S(X_Q, X_I)|H_0))dS$$

$$FMNR = \int_{-\infty}^{t} p(S(X_Q, X_I)|H_1))dS$$



Slika 7. a)Zavisnost FNMR i FMR od praga tolerancije b)Odnos FNMR i FMR u različitim biometrijskim aplikacijama [1]

Performanse biometrijskog sistema u identifikacionom režimu rada teže je odrediti, ali mogu biti ocenjene korišćenjem podataka o performansama sistema u verifikacionom režimu rada. Lažno prihvatanje i lažno odbijanje biometrijskog sistema u identifikacionom režimu rada označićemo sa FMRN i FNMRN, gde N predstavlja broj različitih šablona u bazi podataka. Radi veće jednostavnosti, pretpostavićemo da svaki šablon sadrži samo jednu biometrijsku karakteristiku, kao i da vršimo samo jedno poređenje po osobi. Tada je $FNMR_N \cong FNMR$, a $FMR_N = 1 - (1 - FMR)^N \cong N * FMR$(sa tim što je ova ocena precizna samo ako je $N * FMR < 0.1$)

Zahtevi za tačnošću zavise od tipa biometrijske aplikacije, Na primer za forenzičke aplikacije kao što su identifikacija kriminalaca, od suštinske važnosti je imati mali FNMR, dok FMR nije od velikog značaja. Takav zahtev je lako objasniti potrebom da po svaku cenu sistemu ne promakne kriminalac čak i po cenu manuelnog ispitivanja velikog broja potencijalno pogrešnih potvrda generisanih od strane sistema. Sa druge strane mali FMR je daleko važniji nego FNMR u slučaju bezbednosnih aplikacija, gde je glavni cilj sprečavanje potencijalnih uljeza da pristupe sistemu. Postoje i aplikacije gde su i FMR i FNMR podjednako važni, naprimer primena biometrije u bankomatima. Tada lažno prepoznavanje dovodi do direktnog gubitka novca, a lažno odbijanje do frustracija i gubitka proverenja mušterija. Na slici 2b prikan je odnos između FMR i FNMR.

Pored FMR i FNMR, za ocenu performansi biometrijskog sistema koriste se sledece mere:

- equal error rate ili crossover error rate (EER ili CER) – stopa jednakih greški, podrazumeva slučaj kada su i stopa prihvatanja i odbijanja grešaka iste. Što je ova stopa manja, to je sistem pouzdaniji
- failure to enroll rate (FTE ili FER) – stopa neuspešnog snimanja šablona
- failure to capture rate (FTC) – verovatnoća da sistem ne uspe da detektuje prisustvo korektno dostavljenih biometrijskih podataka
- template capacity – maksimalni broj zapisa u bazi podataka

Danas postoji više multibiometrijskih sistema koji su u upotrebi. Međutim ti sistemi su komercijalnog tipa te iz tog raloga nije bilo moguće utvrditi performanse tih rešenja. Multibiometrijski sistemi otvorenog koda nisu se ne mogu naći. Jedini multibiometrijski sistem kome se moglo pristupiti je Multimodalni biometrijski sistem koji je deo projekta "Multimodlna biometrija u sistemima za upravljanje identitetima" podržanog od strane Ministarstva prosvete i nauke, TR-32013, koji se rađen na Fakultetu organizacionih nauka pri Laboratoriji za Multimedijalne komunikacije. Da bi se dobila približna slika o performansama multibiometrijskih sistema rezultati dobijeni u Multimodalnom biometrijskom sistemu MMBio su upoređivani sa unimodalnim biometrijskim sistemima kojima je pristup bio dozvoljen.

4.1 NBIS rešenje za rad sa otiskom prsta

NBIS(eng. Nist Biometric Image Software) [(25)] u sebi sadrži sledeće aplikacije:

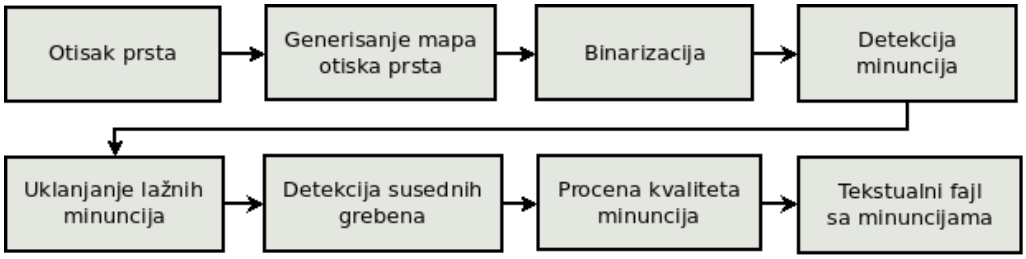
- PCASYS – omogućava klasifikaciju otisaka prstiju u neku od kategorija: levu ili desnu petlju, deltu, spiralu ili ožiljak. Za klasifikaciju koristi neuronske mreže (konkretno višeslojni perceptron)
- MINDTCT – omogućava ekstrakciju minucija kao i ocenu njihovog kvaliteta
- NFIQ – vrši procenu kvaliteta slike otiska prsta na skali od 1 do 5
- IMGTOOLS – skup alata za transformaciju slika kao i konverziju u različite formate
- BOZORTH3 – algoritam za poređenje minucija, funkcioniše na osnovu karakteristike koje generise MINDTCT

NBIS je dostupan na Linux i Windows platformama, a rađen je u programskom jeziku C. Pozivi aplikacija vrše se isključivo preko konzole. Najznačajnije komponente ovog

softverskog paketa su MINDTCT i BOZORTH3 aplikacije. Kod ovog rešenja nalazi se u javnom domenu, projekat je otvorenog koda.

4.1.1 MINDTCT - alat za ekstrakciju minucija

MINDTCT za ulaz dobija sliku otiska prsta na osnovu koje određuje sve minucije otiska, dodeljujući svakoj minucija lokaciju, orijentaciju, tip i kvalitet. Algoritmi i parametri programa podešeni su da rade sa otiscima skeniranim na rezoluciji od 19.69 ppmm i šemom boja od 256 nijansi sive. Mogući ulazi predstavljaju ANSI/NIST fajl, WSQ, JPEG ili IHead fajl. Takodje prilikom pokretanja programa, moguće je izabrati opciono unapređivanje slika na kojima je kontrast loš. S obzirom da kvalitet otisaka varira, od ključne je važnosti analiza otiska i određivanje područja sa lošijim kvalitetom, koja bi mogla da dovedu do slabijih performansi biometrijskog sistema. Nekoliko karakteristika može biti izmereno u cilju prikupljanja informacija o kvalitetu lokalnih regiona otiska prsta. To mogu biti pravac toka grebena u otisku prsta, detekcija oblasti sa slabim kontrastom, neodređenim tokom grebena i visokom zakrivljenošću. Prvi korak u radu MINDTCT alata jeste formiranje mapa koje opisuju ove karakteristike. Algoritam za detekciju minucija na ovom sistemu dizajniran je da operiše na binarizovanim otiscima, gde crni pikseli predstavljaju grebena a beli doline u epidermisu kože prsta. Kako bi se kreirao ovako binarizovan prikaz otiska, svaki piksel iz crno – bele skale (grayscale) mora biti analiziran kako bi se odredilo da li mu treba dodeliti crni ili beli piksel. Ovaj proces naziva se binarizacija. Pikselu se dodeljuje binarna vrednost na osnovu pravca toka grebena bloka u okviru koga se piksel nalazi. Ako za blok ne postoji pravac toka grebena, pikselu se dodeljuje bela boja. U slučaju da pravac toka grebena postoji, tada se intenziteti piksela koji se nalaze oko piksela u fokusu analiziraju pomoću rotirajuće rešetke. Na osnovu binarizovanih otisaka prstiju, vrši se prepoznavanje minucija primenom šablona.



Slika 8. Koraci za ekstrakciju minucija

Korišćenjem odgovarajućih šablona, potencijalni kandidati za minucije pronadjeni su sa maksimalnom preciznošću od šest piksela. Primenjeni algoritam spada u kategoriju proždrljivih algoritama. Usled te činjenice, šanse za propuštanje neke od minucija su minimalne, međutim među kandidatima za minucije nalazi se određen broj lažnih minucija. Usled toga neophodno je preduzeti korake kojima bi se ovi lažni kandidati eliminisali. Ti koraci uključuju uklanjanje ostrva, jezera, rupa, minucija u oblastima sa lošim kvalitetom, minucija sa strane, kuka, preklapanja, preširokih kao i preuskih minucija. Modul za poređenje otisaka prstiju često koristi dodatne informacije pored samih koordinata minucija. Pomoćne informacije ulkučuju pravac minucije, njen tip kao informacije vezane za njene susede. Međutim ne postoji standardizovan pristup skadištenju informacija o susedima. Različiti unimodalna rešenja koriste različite topologije i atribute. NBIS-ova šema za određivanje suseda nasleđena je direktno iz iz HO39. Jedan od ciljeva ovog softverskog paketa je da za svaku pronađenu minuciju da procenu kvaliteta/pouzdanosti. Čak i sa svim koracima koji su iskorišćeni za eliminisanje lažnih minucija, neka od njih može ostati u konačnim rezultatima. Robustan metod određivanja kvaliteta minucija može pomoći u tome da nepouzdanoj minuciji bude dodeljen lošiji kvalitet. Uz pomoć određivanja praga pouzdanosti, gde se vrši balans između želje za odbacivanjem lažnih minucija i potrebe da u tom procesu ne budu odbačene i prave, moguće je unaprediti performanse sistema. Po završetku procesa ekstrakcije minucija, MINDTCT upisuje rezultate u fajl. U slučaju da je ulaz u program bio ANSI/NIST fajl, MINDTCT mu dodaje dva nova zapisa i upisuje novi ANSI/NIST formatiran fajl na lokaciju <oroot>.mdt, gde je <oroot> parametar koji se prosleđuje programu MINDTCT. U slučaju da ulazni fajl nije u ANSI/NIST formatu, tada se rezultujuće minucije skladište u tekstualnom fajlu <oroot>.min, a rezultati binarizacije pamte u sirovom obliku. Pored ovoga, nezavisno od vrste ulaznog fajla, pronađene minucije se takođe upisuju u tekstualni fajl <oroot>.xyt koji je formatiran za korišćenje od strane BOZORTH3 modula za poređenje. Ovaj fajl sadrži informacije o koordinatama minucija, njihovom uglu kao i kvalitetu. Nekoliko pratećih fajlova takođe se dobijaju kao izlazi iz MINDTCT, konkretno tekstualni faljovi koji predstavljaju mapu oblasti sa slabim kontrastom visokom zakrivljenošću, neodređenim pravcom toka grebena, kao i mapa generalizovanog kvaliteta.

4.1.2 BOZORTH3 - alat za poređenje minucija

Dve ključne karakteristike ovog modula za poređenje su:

- Koristi isključivo informacije o minucijama i to njihovu lokaciju(x,y) kao i orijentaciju „t“, predstavljeno n-torkom {x,y,t}
- Algoritam je invarijantan u odnosu na rotacije i translacije

Algoritam se sastoji iz sledeća tri koraka:

- Obrazovanja tabela međuzavisnosti minucija unutar istih otisaka prsta (eng. Intra-Fingerprint Minutia Comparison Tables), za svaki od otisaka prstiju koji učestvuju u poređenju.
- Obrazovanja tabela međuzavisnosti minucija različitih otisaka prstiju (eng. Inter-Fingerprint Compatibility Table). Ova tabela se konstruiše na osnovu tabela dobijenih u prethodnom koraku.
- Prolazak kroz tabelu međuzavisnosti i povezivanje zapisa u tabeli u klastere. Potom se međusobno kompatibilni klasteri kombinuju i izračunava se skor poređenja.

Izlaz iz BOZORTH3 modula poređenje jesu skorovi poređenja, svaki skor u posebnom redu. U idealnom slučaju skor poređenja je visok ako upoređeni otisci pripadaju istom prstu iste osobe, a nizak ukoliko to nije slučaj. Takodje algoritam koji se koristi za prolazak kroz tabelu međuzavisnosti ne garantuje optimalno rešenje problema pronalaženja najduže putanje povezanih asocijacija, te samim tim ne garantuje ni potpuno tačno izračunavanje skora poređenja. Važno je i napomenuti da skor poređenja odslikava ali ne predstavlja broj identičnih minucija na različitim otiscima.

4.1.3 MARF - rešenje za rad sa glasovnim zapisom

Modularni okvir za prepoznavanje zvuka (eng. Modular Audio Recognition Frejmvork - MARF) [(26)], predstavlja kolekciju algoritama za procesiranje zvuka i govora implementiranih u JAVA programskom jeziku. Algoritmi koji se koriste u koracima preprocesiranja, izdvajanja karakteristika i klasifikacije. Organizovani su u jedan okvir kroz koji se mogu lako koristiti u svrhu programiranja sopstvenih sistema vezanih za prepoznavanje zvuka uopšte. MARF takođe predstavlja istraživačku platformu za različita merenja performansi implementiranih algoritama. MARF je projekat otvorenog koda koji je javno dostupan na SourceForge servisu. Cilj ovog projekta je kreiranje opšteg open source okvira koji bi omogućio istraživačima iz oblasti prepoznavanja zvuka da isprobaju ili primene različite metode i algoritme, kao i da ih iskoriste u sopstvenim aplikacijama. Kao dokaz ovog

koncepta razvijena je korisnička aplikacija za tekstualno nezavisnu identifikaciju govornika (SpeakerIdentApp program). Takođe, postoji i niz aplikacija za testiranje i aplikacija koje demonstriraju kako se koriste različiti aspekti modularnog okvira za prepoznavanje zvuka. Projekat je započet 2002. godine od strane četiri studenta sa Concordia Univerziteta u Montrealu, Kanada. Iako se često objavljuju različite radne verzije ovog projekta, poslednja zvanična verzija je objavljena 2007. godine.

MARF ima nekoliko aplikacija. Većina je vezana za pipeline za prepoznavanje, koji uključuje sledeće segmente:

- učitavanje uzoraka
- preprocesiranje
- izdvajanje karakteristika
- trening
- klasifikacija

U suštini, proces prepoznavanja počinje učitavanjem uzorka, zatim se uzorak podvrgava preprocesiranju i normalizaciji. Nakon toga se vrši ekstrakcija karakteristika i na samom kraju se trenira sistem ili se izvršava proces klasifikacije. Rezultat treninga su vektori karakteristika, a rezultat klasifikacije je kolekcija mogućih identifikacija subjekta, sortirana od najverovatnijeg do najmanje verovatnog pogotka. Aplikacija koja koristi okvir mora da odabere konkretnu konfiguraciju i podmodule za faze preprocesiranja, izdvajanja karakteristika i klasifikacije. Postoje dve varijante korišćenja MARF-a sa strane aplikacije:

- Trening – metoda train()
- Prepoznavanje – metoda recognize()

Trening se izvršava na čistoj instalaciji MARF-a kako bi u nju bili uneti neki trening podaci. Prepoznavanje je sam proces identifikacije uzorka na osnovu šablona prethodno sačuvanih u toku treninga.

4.1.4 Metode za skladištenje preprocesiranje, ekstrakciju karakteristika i klasifikaciju

Baza govornika definiše se u CSV datoteci, speakers.txt u okviru aplikacije. Ova datoteka ima sledeći format: <id\:int>,<ime:string>,<training-uzorci:list>,<testing-uzorci:list> Za upravljanje

modulima za skladištenje podataka definisan je StorageManager interfejs. Svaki modul za skladištenje će implementirati ovaj interfejs i njegove metode jer svaki modul treba da zna kako da se serijalizuje. Samim tim aplikaciju koja koristi MARF to ne treba da interesuje. Dakle, StorageManager je osnovna klasa sa apstraktnim metodama dump() i restore(). Ove metode bi uopštile serijalizaciju modela. Podaci korišćeni za trening se moraju na neki način sačuvati kako bi kasnije bili iskorišćeni u procesu klasifikacije. Zbog toga se šalju vektori karakteristika FFT i LPC kroz TrainingSet/TrainingSample klase, koje kao rezultat čuvaju srednje vektore za trening modele. Preprocessing je klasa koja sadrži metode za preprocesiranje: uklanjanje šuma, uklanjanje tišine, normalizaciju, isecanje audio snimka i slično. Ovu klasu nasleđuju i moduli za FFT filtriranje, pojačavanje visokih frekvencija i endpointing. Uklanjanje šuma, tišine i normalizacija su implementirani u okviru Preprocessing klase. FeatureExtraction je klasa koja je zadužena za proces izdvajanja karakteristika. Sadrži metodu getFeaturesArray() koja vraća vektor karakteristika. Konkretne implementacije algoritama za izdvajanje karakteristika nasleđuju ovu klasu i to su FFT, LPC, MinMaxAmplitude, F0, Cepstral, Segmentation i RandomFeatureExtraction. Classification je klasa zadužena za proces klasifikacije. Sadrži metode train(), dump() i restore(). Takođe, ovu klasu nasleđuju konkretni moduli za klasifikaciju. Moduli su kategorisani u Distance, NeuralNetwork i Stochastic grupe, a postoji i klasa koja bira metod za klasifikaciju nasumično – RandomClassification. Distance moduli koji su implementirani su EuclideanDistance, ChebyshevDistance, MarkowskiDistance, DiffDistance i MahalanobisDistance.

4.1.5 DigitalPersona - rešenje za rad sa otiskom prsta

The One Touch® for Windows SDK: Java Edition je softverski alat kompanije DigitalPersona za razvoj aplikacija koje omogućavaju integrisanje biometrije otiska prsta u široki spektar različitih tipova JAVA aplikacija. Ovaj SDK pruža standardne mogućnosti jednog biometrijskog sistema, kao što su akvizicija podataka, ekstrakcija biometrijskih karakteristika, njihovo skladištenje i kasnije poređenje. Korišćena verzija SDK omogućava rad isključivo sa Windows operativnim sistemom. U kasnijim verzijama dodata je i podrška za druge operativne sisteme, konkretno različite distribucije Linux i MacOS operativnih sistema. Pored Jave, podržani su i drugi programski jezici kao što su VisualBasic i C++. API je detaljno dokumentovan i priložene su aplikacije koje predstavljaju moguće primere

korišćenja ovog razvojnog alata. Naravno ovaj SDK dolazi u paketu sa senzorima ove kompanije i zahteva njihovo korišćenje.

5 Konceptualni model multibiometrijskog ekosistema za utvrđivanje identiteta

U ovom poglavlju će biti opisan i definisan softverski ekosistem. Potom će biti navedene raličite tehnologije koje će biti gradivni elementi multibiometrijskog ekosistema i ustanoviti veze između njih. Na kraju poglavlja biće predstavljen konceptualni model multibiometrijskog ekosistema.

5.1 Softverski ekosistem

Pojam ekosistema potiče iz ekologije i označava zajednicu živih organizama u sprezi sa neživim komponentama okruženja i uzajamnim delovanjima deluju kao sistem. Ljudski ekosistem sastoji se od učesnika u jednom okruženju, veza između učesnika, aktivnosti učesnika i transakcija u uspostavljenim vezama između njih a na koje utiču fizički i ne-fizički faktori. Ovim se mogu predstaviti komercijalni i socijalni ekosistemi. U komercijalnom ekosistemu učesnici su preduzeća, dobavljači i kupci, dok su faktori predstavljeni kroz robu i usluge a transakcije uključuju kako finasijske transakcije tako i informacije i razmenu znanja, podršku, kontakte itd. Socijalni ekosistem se sastoji od učesnika i njihovih socijalnih veza i razmene informacija u različitim formama. Softverski ekosistem se sastoji od skupa softverskih rešenja koja omogućavaju podršku i automatizaciju aktivnosti i transakcija među učesnicima u datom socijalnom ili poslovnom ekosistemu i organizacijama koje pružaju ova rešenja [83]. Primer navedenih ekosistema se može ilustrovati kroz softverske ekosisteme nastale oko operativnih sistema u ranim dvedesetim godinama prošlog veka koju su obeležili kompanije kao što su Microsoft i IBM. One su svojim delovanjem okupljale veliki broj različitih nezavisnih kompanija koje su razijale softver sa ciljem postizanja dominacije na tržištu desktop računara i operativnih sistema koji su se na njima izvršavali. Sličan primer se može videti i u Web 2.0 kontekstu gde softverski ekosistemi postoje već nekoliko decenija unzad (Web-orjentisana arhitektura, Social Web, Rich Internet Application, itd).

Softverski ekosistem je definisan kao skup uloga i pripadajućih aktera koji funkcionišu kao celina i u stalnoj su sprezi sa tržištem softvera i usluga podržavajući povezanost između njih [84]. Softverski ekosistemi su predmet rasprava o praksi softverskih kompanija koje imaju za cilj napredak i unapređenje softverskih ekosistema. Evidentirani su relevantni problemi sa stanovišta softverskog inženjerstva, ekonomskih apsekata softverskih ekosistema, vlasništva i transfera

Programiranje na nivou krajnjeg korsinika	MS Excell Calc Mathematica VHDL	Google Docs Yahoo! Pipes Micfrosoft PopFly	Trenutno ne postoji
applikacija	MS Office Libre Office	eBay Amazon	Trenutno ne postoji
Operativni sistem	MS Windows Linux Apple OS X	Google AppEngine Yahoo developer JBoss	Nokia Symbian, S60 Palm OS Android iPhone
Kategorija / Platforma	desktop	web	mobile

Slika 9. Kategorizacija ekosistema

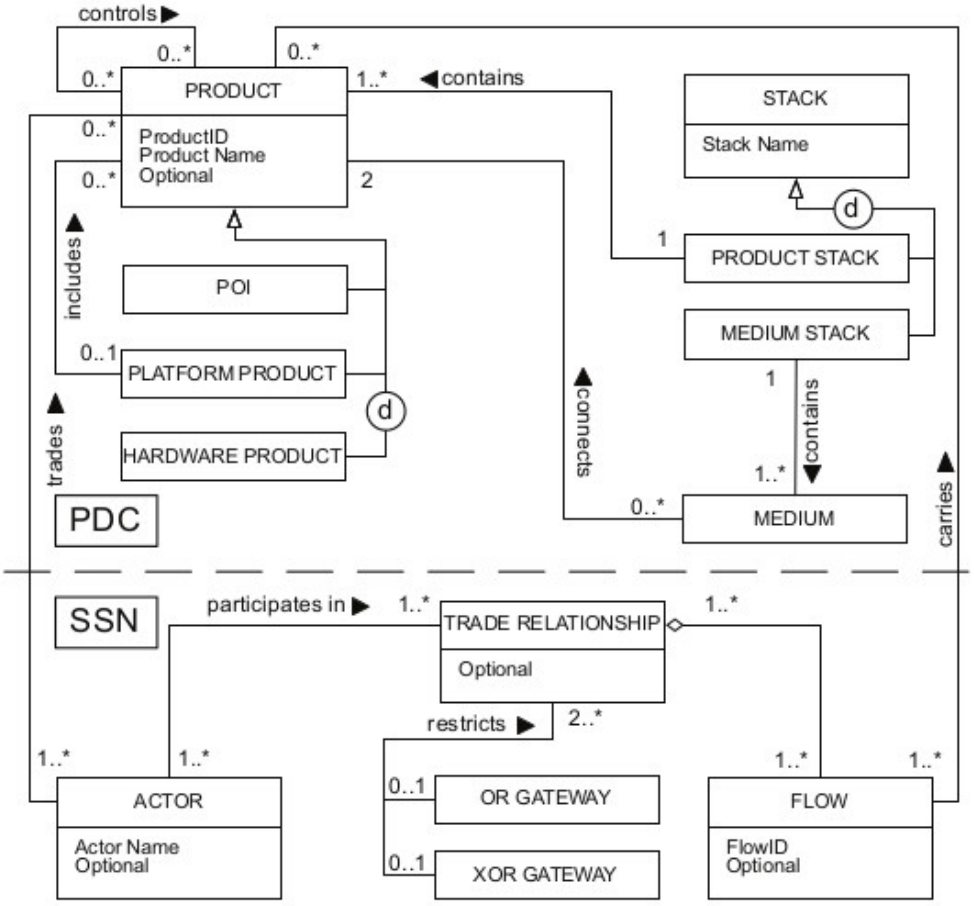
dobara i usluga, itd [85]. Razvoj modeliranja i metoda strateškog planiranja softverskih proizvoda i usluga se konstantno unapređuju. Kako se u softverskim ekosistemima često preklapaju softverski proizvodi i usluge, za validaciju metoda modelovanja softverskih ekosistema su uvedeni standardi [86] kao i metode za modelovanje i validaciju biznis modela u softverskoj industriji [87]. Ove [87, 86] metode modeliranja su uglavnom neformalne zbog njihove svestrane prirode. Ipak, potreban je jedan formalni standard za softverske ekosisteme tako da je moguće modelovanje i ekosistema i tog okruženja u kome softverski proizvodi i usluge funkcionišu. Metode modelovanje ekosistema su razvijene za specifične slučajeve u specifičnim ekosistemima i u drugim delatnostima sa fokusom na distribuciju zaliha i proizvodni menadžemnt. Takve metode i modeli nisu pogodni za softversku industriju zbog svoje specifične prirode.

Boucharas je u radu [3] formalizovao modelovanje softverskog ekosistema i pri tome je definisao dva nivoa :

- PDC (Product Deployement Context) - u kome se daje pregled arhitekture i međuzavisnosti softverskih proizvoda u radnom okruženju. Detalji u PDC nivou prikazuju hijerarhiju između različitih proizvoda i komponenti i opšti pogled na položaj različitih lokacija u odnosu na mrežu.
- SSN (Softvare Supplay Network) - predstavlja veze među softverom, hardverom i servisima koji sarađuju pri zadovoljavanju zahteva tržišta. SSN model omogućuje softverskim firmama da definišu svoj biznis model i

strateški nastup jer su u datom modelu prikazane veze između firmi odnosno njihovih proizvoda.

Na slici 10 je meta model softverskog ekosistema. Korišćenje ovog modela je pre svega vezano za jedan kokretan softverski proizvod u kome učestvuju proivođaći softverskih proizvoda i komponenti. Jedan od ciljeva multibiometrijskog ekosistema je da se iskoristi softver otvorenog koda za implemntaciju radnog okruženja. Time se izbegava vezivanje za proizvode jednog softverskog proizvođača.



Slika 10. Meta model softverskog ekosistema [3]

Platforma koja se koristi za izgradnju ekosistema obuhvata mnoge aspekte [84]. Sa inženjerskog stanovišta , softverski ekosistem pruža tehnologiju za implementaciju, okruženje za sveukupnu infrastrukturu softverskog projekta i metodologije razvoja. Pored tehnološkog aspekta treba uzeti u obzir socijalne, pravne i poslovne aspekte. Ekosistem se može posmatrati i kao biznis i upravljački model sa marketingom kao jednom od strateških prednosti. Softver otvorenog koda obezbeđuje rešenja sa svaki od navedenih aspekata potrebnih za razvoj i rast ekosistema. Iako se radi o

softveru otvorenog koda postoje različiti tipovi softvera otvorenog koda koji se razlikuju na osnovu principa, prakse, kulture i licenci [88]. Zajednički apspekti softvera otvorenog koda su transparentnost razvoja i sloboda u pravljenju kompleksnih sistema iz lako dostupnih delova. Oni pružaju jedinstveni način da se brzo izgradi ekosistem bez velikih početnih ulaganja [89]. Kompanije kao što su Sun Microsystems(sada Oracle), Nokia, Google su imale ili imaju ekosisteme otvorenog koda: Java, Symbian operativni sistem, Android operativni sistem, sa velikim potencijalom da privuku druge kompanije ili samostalne programere koji se mogu pridružiti naporima razvoja tih ekosistema.

Trend prebacivanja vlasničkog softvera u softver otvorenog koda je relativno nov te ima jako malo predloženih rešenja za održivi razvoj ekosistema otvorenog koda za vlasnički softver [90]. Iako softver otvorenog koda omogućava održivu platformu sa rast softverskih ekosistema iz više uglova - implementacione tehnologije, metodologije razvoja, poslovni model, organizaciona struktura i legalnost - još uvek ima malo kompanija koje se odlučuju da uđu u svet softverskih ekosistema otvorenog koda.

5.2 Računarstvo u oblaku

Računarstvo (*computing* eng.) je transformisano u model koji je sastavljen od usluga koje se pružaju korisniku na sličan način kao tradicionalne komunalne usluge kao što su voda, struja, gas, telefonija [91]. U takvom modelu, pristup servisima se zasniva na zahtevima korisnika bez obzira na to gde se nalazi pružaoc usluga ili kako se usluge isporučuju. Nekoliko računarskih paradigmi pruža usluge ovog tipa:

- Klaster računarstvo (*Cluster coumputing* eng.)
- *Grid* računarstvo (*Grid computing* eng.)
- Računarstvo u oblaku (*Cloud computing* eng.)

Pristup sadržajima Interneta je svakodnevna mogućnost koja je korisniku obezbeđena bez uticaja infrastrukture koju koristi pružaoc te usluge. Ova infrastruktura se sastoji od *Data* centara koji se održavaju i nadgledaju od strane provajdera sadržaja. Računarstvo u oblaku je proširenje pružanja usluge sadržaja, pri čemu su mogućnosti poslovnih aplikacija predstavljene kao sofisticirani servis koji je dostupan preko mreže. Napredak u polju tehnologija mikroporcesora omogućio je provajderima sadržaja da dodaju još jednu uslugu: iznajmljivanje virtualnog računara na zahtev korisnika. Ovakvom uslugom je povećana izolacija

poslovnih aplikacija koje se javljaju kao servis, jer korisnik u virtualnom računaru sada može da instalira operativni sistem na način koji zahteva poslovna palikacija u okruženju koje korisnik prilagađova svojim potrebama.

Grid računarstvo omogućuje deljenje, selekciju i agregaciju širokog spektra geografski razuđenih resursa. Ovi resursi uključuju superračunare, klastere raćunara, sisteme za skladištenje podataka, specijalizovane uređaje koji su u vlasništvu različitih organizacija [92]. P2P (*Peer-to-Peer*) računarstvo povećava decentralizaciju računarskih resursa, tako da ne postoji striktna razlika između klijenta i servera. Svi čvorovi su jednaki i istovremeno mogu biti i klijenti i serveri. P2P računarstvo omogućuje deljenje odnosno smanjenje troškova, interoperabilnost i agregregiranje resursa, povećanu skalabilnost i pouzdanost, povećanje autonomije, anonimnost ili privatnost i *ad-hoc* komunikaciju i kolaboraciju [93].

Do sada je bilo više pokušaja da se definišu Klaster računarstvo, Grid raćunarsvo i Računastvo u oblaku [94] :

- ”Klaster je tip paralelnog i distribuiranog sistema, koji se sastoji od kolekcije međusobno povezanih samorastalnih računara koji rade zajedno kao jedan računarski resurs”[95, 96]
- ”*Grid* je tip paralelnog i distribuiranog sistema koji omogućuje dinamičko deljenje, selekciju i agregaciju geografski distribuiranih autonomnih resursa u vreme izvršavanja a u zavisnosti od njihove raspoloživosti, mogućnosti, performanse, koštanja, korisničkog zahteva za kvalitetom servisa”[97]
- ”*Cloud* je tip paralelnog i distribuiranog sistema koji se sastoji od kolekcije međusobno povezanih virtualnih računara koji se dinamički obezbeđuju i predstavljaju kao jedan ili više unificiranih računarskih resursa zasnovanih na dogovoru na nivou usluga kroz pregovaranje između pružaoca usluge i korisnika”[94]

Platforma na kojoj je zasnovano Računarstvo u oblaku poseduje karateristike i Klastera i *Grid*-a, sa svojim specifičnim atributima i karakteristikama:

- Podrška u Virtualizaciji
- Dinamički sastavljeni servisi sa *Web* Servis interfejsima (SOAP i REST)
- Izgradnja novih servisa (*Cloud compute*, skladište, aplikacije)
- Kvalitet servisa na osnovu zahteva i dogovora

Tržišna podrška Računarstva u oblaku kao *Data* centara nove generacije se sastoji od sledećih entiteta:

- **Korisnici/Brokeri** : Korisnici ili posrednici koji rade u njihovo ime mogu zahtevati servise iz bilo kog mesta u svetu
- **SLA (Service Level Agreement) alokator resursa** : Deluje kao interfejs između korisnika ili posrednika i *Data* centra odnosno *Cloud* provajdera. Zahteva sledeće mehanizme :
  - **Service Request Examiner** : Po dolasku zahteva za servisom, SRE interpretira zahtev za kvalitetom servisa pre donošenja odluke o prihvatanju ili odbijanju zahteva.
  - **Naplata** : Mehanizam naplate odlučuje kako će zahtev za servisom biti naplaćen u skladu sa SLA.
  - **Monitor Virtualnog računara** : Vodi računa o raspoloživosti virtualnog računara iskorišćenosti resursa
  - **Dispečer** : Raspoređuje izvršavanje prihvaćenih zahteva za resursom i alocira virtualni računar
  - **Monitor zahteva za resursima** : Vodi evedenciju o stanju svakog pristiglog zahetva za resursom
- **Virtualni Računar** : Virtualni računari mogu biti startovani i zaustavljeni na zahtev u jednom fizičkom računaru kao odgovor na prihvaćeni servisni zahtev. Takođe omogućuje da više različitih virtualni računara pokreće različite operativne sisteme na jednom fizičkom računaru zahvaljući potpunoj izolaciji tih virtualnih računara.
- **Fizički računar** : *Data* centar sadrži više servera koji obezbeđuju resurse tako da mogu da ispune servisne zahteve.

Pored navedenih karakteristika NIST definiše sledeće osnovne karateristike [98] :

- ***On-demand self-service*** : Korisnik može jednostrano da automatski obezbedi potrebne računarske resurse, kao što su procesorsko vreme i mrežno skladište, bez zahteva za interakcijom sa čovekom
- ***Broad network access*** : Koristeći heterogene platforme tankog ili debelog klijenta i standardne mehanizme računarske mreže korisnik može da pristupi svojim resursima

- ***Resource pooling*** : Računarski resursi provajdera su udrženi tako da služe više korisnika koriteži model sa više štanara", sa različitim fizičkim i virtualnim resursima koji su dinamički dodeljeni i preusmereni prema zahtevu korisnika.
- ***Rapid elasticity*** : Mogućnost brzog skaliranja naviše i naniže upotrebe resursa na osnovu zahteva.
- ***Measured service*** : *Cloud* sistem automatski kontroliše i optimizuje upotrebu resursa uz pomoć sposobnosti merenja na određenom nivou apstrakcije koji odgovara tipu usluge (na pr. skladištenje, obradu, propusni opseg…)

Takođe u okviru definicije računarstva u oblaku NIST daje sledeće modele servisa [98] :

- **SaaS (*Software as a Service*)** : Korisniku se omogućuje korišćenje aplikacija pokrenutih na *cloud* infrastrukturi. Korisnik ne može da upravlja i kontroliše infrastrukturu uključujući mrežu, servere, operativne sisteme, skladišta niti ponašanje aplikacija osim mogućih limitiranih specifičnih korisničkih podešavanja
- **PaaS (*Platform as a Service*)** : Korisniku se omogućuje instalacija i konfiguracija u *cloud* infrastrukturi, dodeljenoj korisniku na osnovu zahteva, aplikacija kreiranih koristeći programske jezike, biblioteke, servise i alate podržanih od strane provajdera. Korisnik ne može da upravlja i kontroliše infrastrukturu uključujući mrežu, servere, operativne sisteme, skladišta već samo instalirane i konfigurisane aplikacije kao i podešavanja okruženja u kojima se one izvršavaju
- **IaaS (*Infrastrukture as a Service*)** : Korisniku je obezbeđena procesna obrada, skladišta, mrežna infrastruktura, i ostali fundamentalni računarski resursi nad kojima je korsnik u mogućnosti da pokreće, instalira i konfiguriše proizvoljan softver uključujući i operativni sistem i aplikacije. Korisnik ne može da upravlja i kontroliše infrastrukturu ali je u mogućnosti da ima kontrolu nad operativnim sistemom, skladištem i instaliranim i konfigurisanim aplikacijama uz moguću limitiranu kontrolu izabranih mrežnih komponenti.

5.3 Virtualizacija

Virtualizacija se obično definiše kao tehnologija koja uvodi softverski apstraktni nivo između hardvera sa jedne strane i operativnog sistema i aplikacija sa druge strane a koji se izvršavaju na tom istom hardveru. Osnovni zadatak tog apstraktnog nivoa je da sakrije fizičke resurse računara od operativnog sistema. Hardverska platfomra je particionisana na jednu ili više logičkih celina te je moguće pokrenuti više različitih operativnih sistema paralelno.

Takodje u u potrebi je i sledeća definicija u kojoj se virtualizacija definiše kao okvir ili metodologija podele resursa jednog računara na više izvršnih okruženja primenom jednog ili više koncepata ili tehnologija kao što su hardversko i softversko particionisanje, parcijalna ili kompletna simulacija računara, emulacija, kvalitet servisa i mnoge druge.

Virtualizacija je počela u šezdesetim godinama prošlog veka. IBM je na mainframe računarima omogućio podelu sistemskih resursa jednog računara na logičke resurse koji su postali deljivi među aplikacijama. Pre toga aplikacije koje su se izvršavale na računarima i koristile jedan od resursa na računaru su onemogućavale da neka druga aplikacija koristi taj isti resurs sve dok ona ne završi sa radom. Visoka cena računarskih sistema i izvršavanje samo jedne aplikacije u jednom trenutku vremena su učinile veoma teškim opravdanost ulaganja u takav sistem. Pojava relativno jeftinih računarskih sistema početkom devedesetih godina prošlog veka i adaptacija i stvaranje novih operativnih sistema koji su mogli da rade na tim računarima dovela je do transformacije centralizovanih aplikacija u distribuirane aplikacije zasnovane na klijent-server paradigmi.

Početkom ovoga veka tehnološka rešenja koja omogućuju virtualizaciju su implementirana u računarske sisteme zasnovane na Intel kompatibilnim procesorima. U prvoj dekadi ovoga veka razvijene su metode i tehnologije koje su omogućile da se privilegovane instrikcije procesora mogu izvršavati istovrenemo u više operativnih sistema na jednom računarskom sistemu.

Osnovne prednosti koje nudi virtualizacija su:

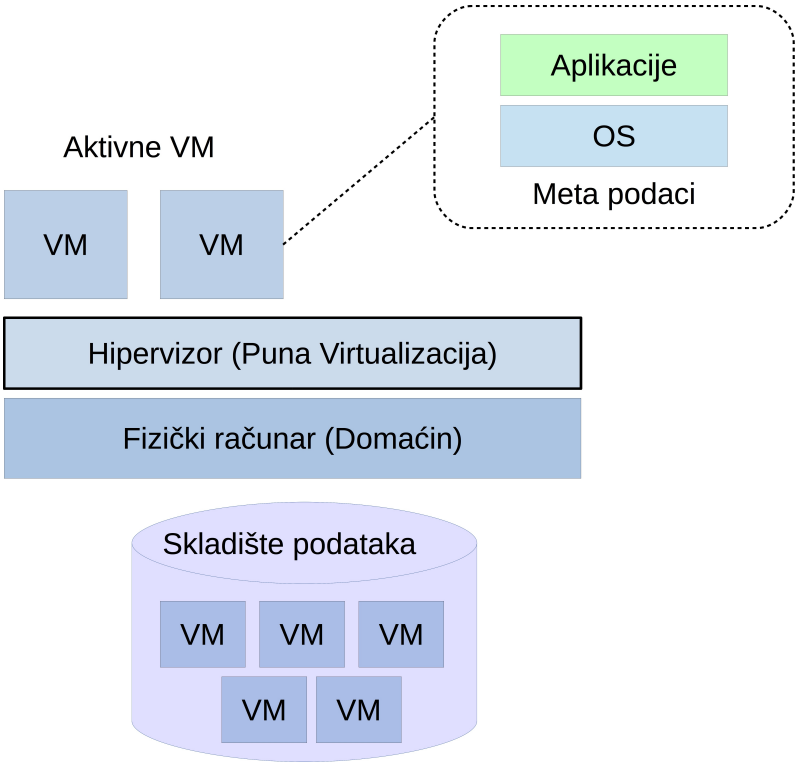
- **Deljenje resursa** - Za razliku od ne virtualizovanog okruženja gde su svi resursi posvećeni pokrenutim programima, u virtualizovanom okruženju vritualne mašine dele fizičke resurse računara kao što su procesor, memorija, disk i mrežni uređaji datog računarskog sistema.

- **Izolacija** - Jedno od ključnih pitanja u virtualizaciji je obezbeđivanje izolacije između virtualnih mašina koje se izvršavaju na istom fizičkom hardveru. Programi pokrenuti u jednoj virtualnom mašini ne vide programe pokenute u ostalim virtualnim mašinama.

Virtualizacija dozvoljava apstrakciju i izolaciju nižih funkcionalnih nivoa na pripadajućem hardveru računara. To omogućuje portabilnost viših funkcionalnih nivoa i deljenje i/ili agregaciju fizičkih resursa. Različiti pristupi virtualiziciji se mogu kategorisati na sledeći način:

- **Platforma za virtualizaciju** - Virtualizacija na nivou platforme omogućuje virtualizaciju i serverskog računara i desktop računara. "Platforma"u ovom kontekstu se odnosi na hardversku platformu i različite komponente koje se koriste na datoj hardverskoj platformi. Ovo ne uključuje samo interne komponente računarskog sistema već i moguće eksterne uređaje koji se priključuju na računar preko nekog od interfejsa, na primer USB, SCSI, Firewire, itd. Ključna komponenta koja omogućuje ovaj tip virtualizacije je hipervizor. On predstavlja komponentu koja virtualizuje platformu, omogućuje deljene fizilkih resursa jednog jednog računarskog sistema. Takođe u okviru njega su implementirane i polise koje omogućuju deljene između više entiteta. Svaki od tih entiteta predstavlja virtualnu mašinu koja predstavlja agregaciju operativnog sistema i skupa aplikacija. Postoje dva osnovna načina implementacije virtualizacije. Prvi način, tip-1 ili *bare-metal* hipervisor, se izvršava direktno nad hardverom i ima ulogu platforme. Drugi način tip-2 ili *hosted* hipervizor, je aplikacija koja se pokreće u istom kontekstu kao i operativni sistem nad tim računarom. Virtualna mašina mora biti konfigurisana gde, posmatrano sa strane hipervizora, konfiguracija predstavlja omotač u kome su opisani zahtevi i ograničenja te virtualne mašine. Na taj način virtualna mašina je hipervizoru predstavljena kao datoteka u nekom formatu, u kome je enkapsuliran i virtualni disk koji ta virtualna mašina koristi. Taj virtualni disk je takođe datoteka koja ima svoj format. Time se upravljanje nad virtualnim mašinama svodi na upravljanje datatokama čime je veoma pojednostavljeno upravljanje celom virtualnom platformom. Da bi upravljanje nad virtualnim mašinama bilo potpuno, na strani hipervizora je potrebno postojanje kataloga virtualnih mašina u kome se pamti stanje i status svake virtualne mašine i njenih komponenti kao i veze između virtualnih mašina. Tako da je moguće, na primer, definisati virtualnu mašinu koja će služiti kao šablon za kreiranje novih virtualnih mašina, saznati

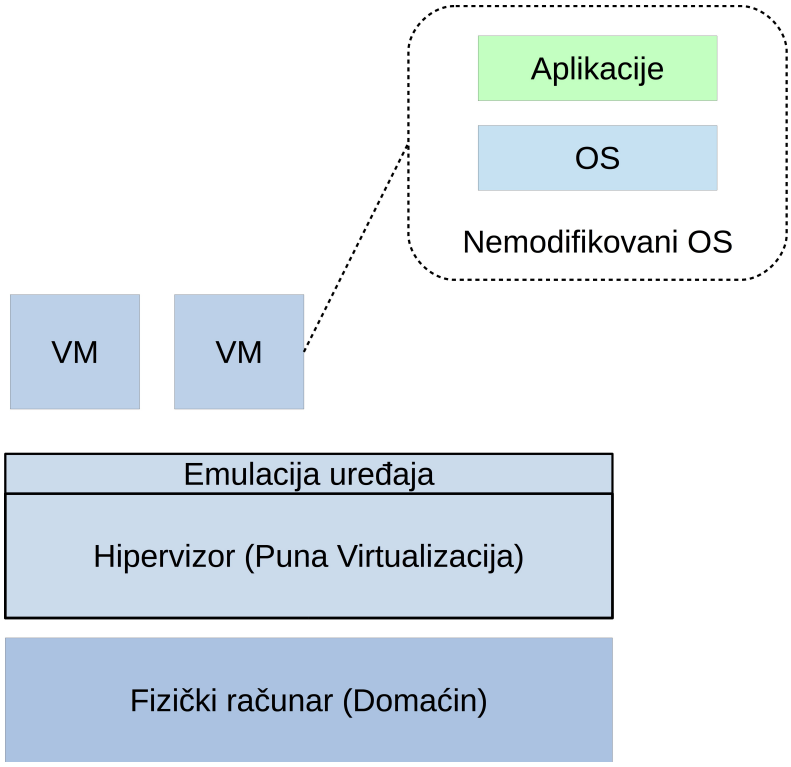
u kojoj datoteci se nalazi sigurnosna kopija virtualnog diska neke virtualne mašine, itd.



Slika 11. Platforma za virtualizaciju

- **Puna virtualizacija** - U ovom pristupu hipervizor se takođe zove i menadžer virtualnih mašina i pokrenut je povrh operativnog sistema računara, obično kao aplikacija koja se izvršava u korisničkom prostoru operativnog sistema. Kao rezultat ovog pistupa u virtualnom mašini aplikacije i gostujući operativni sistem se izvršavaju povrh virtualnog hardvera koji obezbeđuje hipervizor. Kada okruženje virtualne mašine obezbeđuje "dovoljnu reprezentaciju postojećeg hardvera računara dozvoljavajući gostujućem operativnom sistemu da se pokrene bez modifikacije može se smatrati da obezbeđuje "Punu virtualizaciju"". Kod ovog načuna podešavanja rada virtualne mašine dodeljeni U/I uređaji imitiraju stvarne, fizičke, uređaje u hipervizoru. Pristup ovim uređajima u virtualnom okruženju se ostvaruje preko drajvera gostujućeg operativnog sistema i prosleđuje fizičkim uređajima računara preko operativnog sistema domaćina ili drajvera virtualne mašine. Prednost ovakvog pristupa se sastoji u jednostavnosti korišćenja. Korisnik može da instalira softverski proizvod poput Oracle VirtualBox, VMware Workstation, Qemu/KVM, itd. kao bilo koji drugi softverski paket na izabranom operativnom sistemu. Unutar softvera za

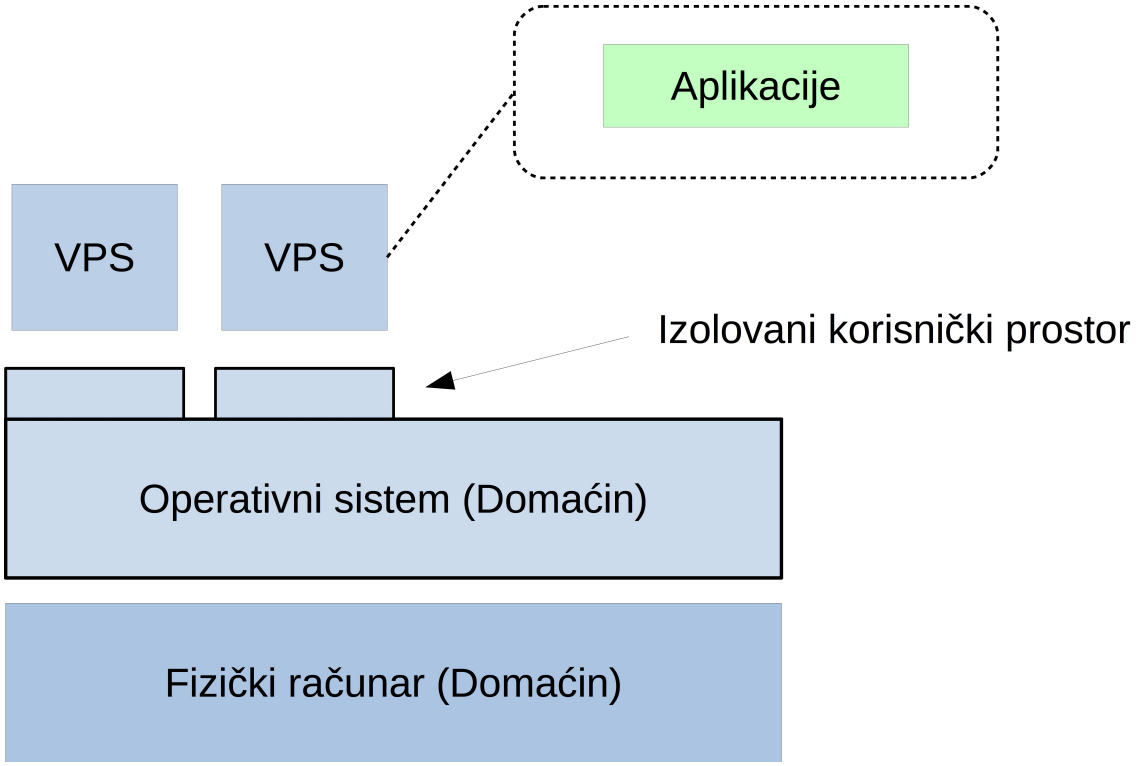
virtualizaciju, gostujući oepративni sistem će biti instaliran a potom i korišćen kao da je pokrenut direktno na hardveru nekog računara. Nedostatak ovakvog pristupa su slabije performanse gostujućeg operativnog sistema, nekada i do 30% u odnosu na izvršavanje na hardveru fizičkog računara.



Slika 12. Puna virtualizacija

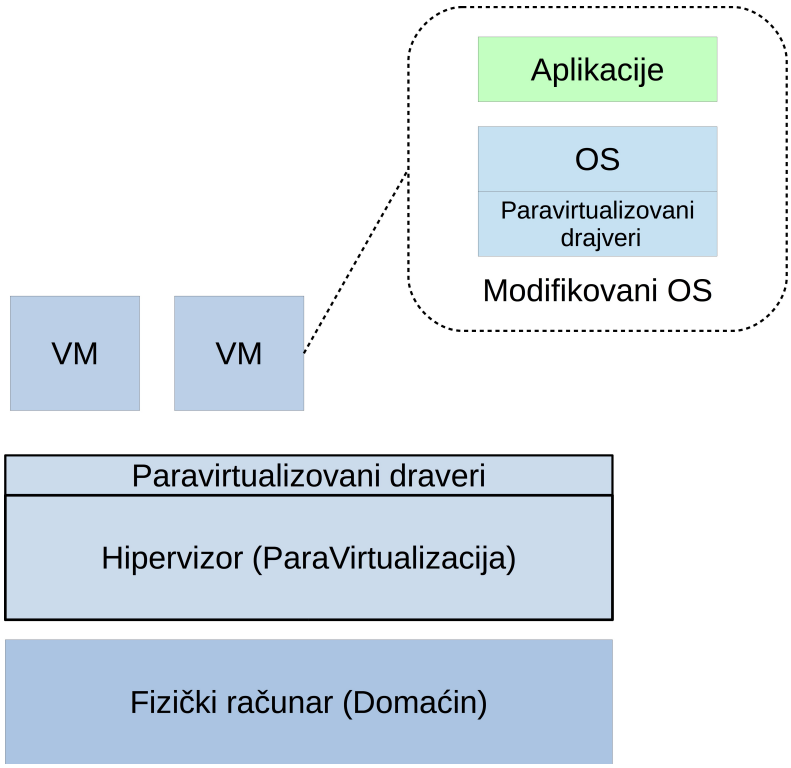
- **Virtualizacija na nivou operativnog sistema** - Kod ovog pristupa implementirana virstualizacija omogućuje paralelno pokretanje više instanci istog operativnog sistema. Ovaj pristup je poznat i pod imenima Single Kernel Image (SKI) ili virtualizacija zasnovana na kontejnerima. Ovaj tip virtualizacije ne virtualizuje hardver jednog računara već je virtualizovan operativni sistem računara koji će biti domaćin gostujućim virtualnim mašinama. To znači da će sve virtualne mašine isti virtualizovani operativni sistem. Taj virtualizovani operativni sistem se naziva virtualizacioni nivo. Takva arhitektura olakšava admnistraciju sistema virtualizacije. Administrator sistema može dodeliti resurse kao što su procesor, memorija i disk prostor, virtualnoj mašini dinamički u vreme izvršavanja. Za razliku od ostalih rešenja virtualizacije, dobra strana virtualizacije na nivou operativnog sistema je efikasnost sistema. Nedostatak ovog tipa virtualizacije je sledeći: pošto virtualne mašine koriste isti kernel kao i operativni sistem domaćina, gostujući operativni sistemi moraju da budu isti kao i osnovni operativni

sistem tog računara. Na primer, nije moguće pokrenuti Windows operativni sistem povrh Linux operativnog sistema.



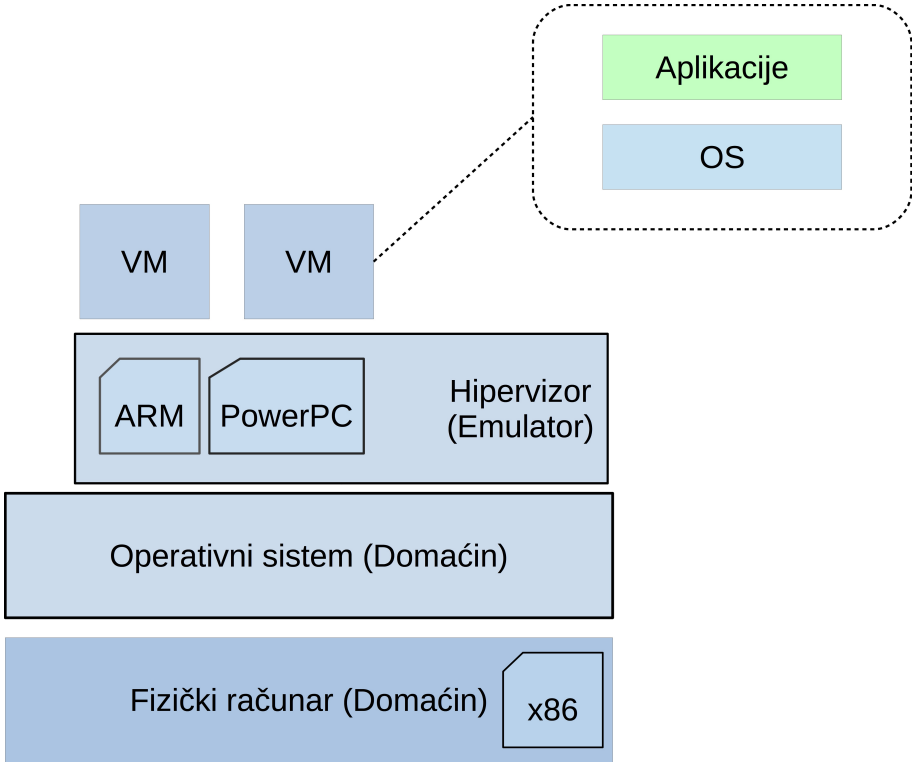
Slika 13. Virtualizacija na nivou operativnog sistema

- **Paravirtuelizacija** - Paravirtualizacija je tehnika koju je razvio Xen a koja obezbeđuje neznatno izmenjeni interfejs ka virtualnim mašinama. Izmena se sastoji u modifikaciji kopije postojećeg hardvera računara pri čemu se nevirtualizovani delovi originalnog Intel ia32 seta instrukcija zamenjuju njihovim virtualizovanim ekvivalentima. Kod ovog tipa virtualizacije gostujući operativni sistem je potrebno modifikovati da bi mogao da radi u takvom virtualnom okruženju. Paravirtualizacija je podskup virtualizacije servera koja obezbeđuje mali softverski interfejs između hardvera računara i modifikovanog gostujućeg operativnog sistema. Kao posledica tehnika paravirtualizacije gostujući modifikovani operativni sistem zna da se izvršava u virtualnom okruženju. Jedna od osnovnih karakteristika paravirtualizacione tehnologije je da hipervizor veoma jednostavan i omogućuje performanse veoma bliske nevirtaulizovanom hardveru. Pristup uređajima u paravirtualnom okruženju je veoma sličan pristupu uređajima u potpunom virtualnom okruženju. virtualni uređaji u paravitualnom okruženju se takođe oslanjaju na drajverima fizičkih uređaja na datom računaru.



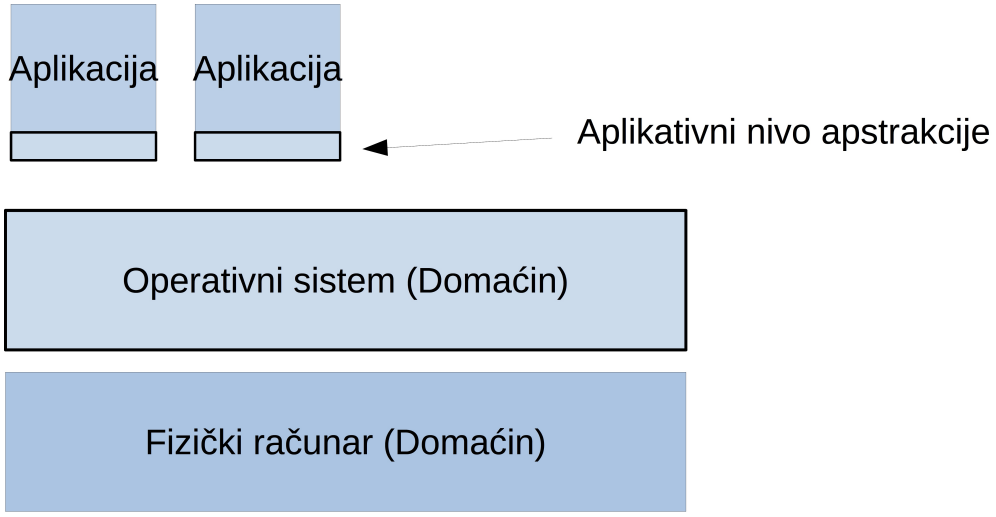
Slika 14. Paravirtualizacija

- **Emulacija** - Emulacija je proces u kome na računaru jedne hardverske arhitekture se emulira, odnosno imitira, računar neke druge hardverske arhitekture. Na emulatori se posmatraju kao neefikasan način virtualizacije jer tokom izvršavanja instrukcija gostujuće hardverske arhitekture moraju imitirati i komponente koje poseduje gostujuća hardverska platforma. Takođe, svaka instrukcija procesora gostujuće hardverske arhitekture se emulira kroz niz instrukcija procesora računara na kome se izvršava emulator. Emulacija instrukcija uključuje i imitaciju registara gostujućeg procesora, pojednotavljenu verziju memorijskog podsistema, kao i emulaciju uređaja. Ovaj tip virtualizacije prevazilazi jednostavne apstrakcije deljenja računarskih resusa jednog sistema već je potrebno kreirati nove karakteristike i funkcionalnosti. Na primer, slika , računar domaćin zasnovan na Intel ia32/ia32-64 arhitekturi, može da emulira hardverske platforme zasnovane na potputno drugačijim tipovima procesora, kao što su ARM, PowerPC.



Slika 15. Emulacija

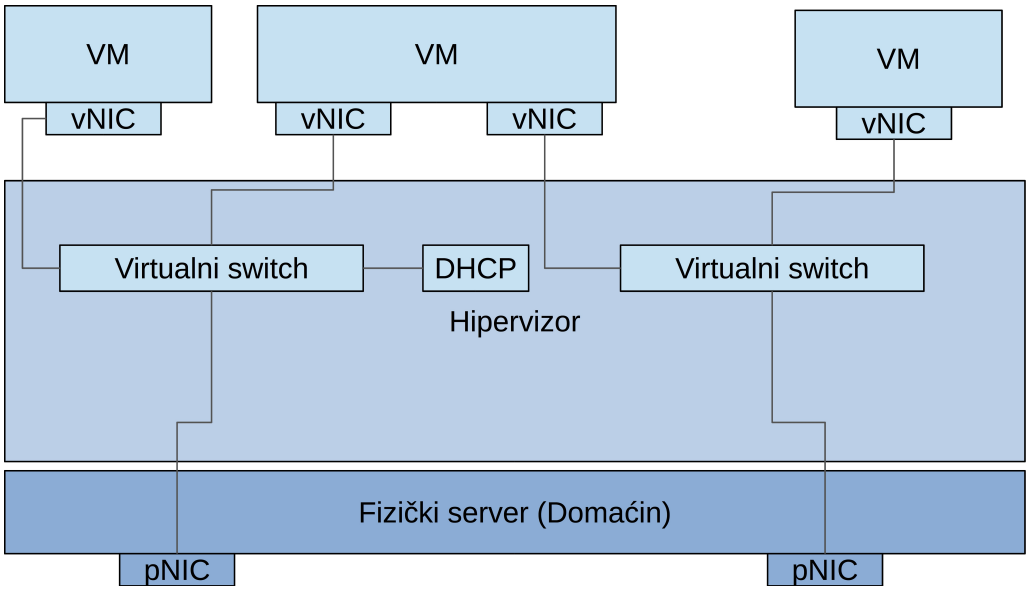
- **Virtualizacija aplikacija** - U ovom pristupu virtualizacije korisnik je u mogućnosti da pokrene serversku aplikaciju, odnosno servis, lokalno bez potrebe za kompleksnom instalacijom te aplikacije na operativnom sistemu računara. Takve virtualizovane aplikacije su osmišljene tako da se mogu pokrenuti u malom virtualnom okruženju koje sadrži samo one resurse neophodne za normalno izvršavanje aplikacije. Takođe u ovom virtualnom okruženju je moguće da svaki korisnik na sistemu može da pokrene aplikikaciju u izolovanom virtualnom okruženju. Ovakvo malo izolovano okruženje se ponaša kao izolacioni sloj između aplikacije i operativnog sistema domaćina. Ovakav tip virtualizacije se pkazao kao jako koristan posebno posmatrano iz perspektive razvoja novih tipova računara. Projektanti koji rade na razvoju nove hardverske platforme mogu da emuliraju ciljno okruženje bez da poseduju ciljano hardversko okruženje. U cilju poboljšanja performansi potpuno virtualizovane platforme u nekim rešenjima je implementirana dinamička translacija izvršnog koda radi minimizacije potrebnog broja translacija. Takođe takva rešenja u sebi sadrže delove kompajlera koji rade optimizaciju generisanja izvršnog koda za datu platformu, na primer QEMU.



Slika 16. Virtualizacija aplikacija

- **Virtualizacija resursa** - Virtualizacija specifičnih računarskih resursa kao što su: mrežni resursi, skladišta podataka, prostor imenovanih resursa, je poznat kao virtualizacija resursa. Postoji više različitih pristupa u virtualizaciji resursa. Neki od njih su:
  - Agregacija više pojedinačnih komponenti
  - Distribuirano računarstvo (Klaster ili Grid) u kome se više računara kombinuje u formu velikog super-računara sa ogromnim resursima: procesorskim, memorijskim, disk prostorom
  - Particionisanje jednog resursa, kao što je disk prostor, u manji broj resursa istog tipa: SAN (Storage Area Network), NAS (Network Attached Storage)
  - **Virtualizacija mreže** - U prethodno opisanim rešenjima i tehnikama virtualizacije može se reći da virtualne mašine čine agregaciju resursa na fizičkom računaru, ali i pored toga je čest slučaj da one ne mogu da komuniciraju među sobom. Zbog toga hipervizori imaju mogućnost mrežne virtualizacije sa ciljem optimizacije saobraćaja. Iako se u ovom slučaju radi o softverski zasnovanim mrežama efikasnost ovog rešenja je veoma velika. Virtualnim mašinama kao delu virtualizovane platforme je obezbeđen virtualnim mrežni adapter. Taj virtualni mrežni adapter je spojen na fizički mrežni adapter ili na virtualnu mrežnu infrastrukturu u samom hipervizoru. (slika). Takođe je moguće kreirati virtualne svičeve radi izolacije saobaćaja u sloju virtualnih mreža u hupervizoru. Na tim virtualnim svičevima je moguće spajati kako fizičke mrežne adaptere

tako i virtualne mrežne adaptere dodeljene virtualnim mašinama. Danas većina hipervizora poseduje sloj virtualnih mreža u nekom obliku. Trend koji je prisutan u poslednjih nekokoliko godina je da proizvođači mrežne opreme pored fizičkih svičeva imaju u ponudi i virtualne svičeve koji se instaliraju kao deo sloja mreže u hipervizorima. Pored komercijalnih proizvoda kao što su: Cisco Nexus 100V ili VMware Distributed Switch, značajno mesto zauzima i Open vSwitch koji je sastavni deo svih virtualizacionih platformi otvorenog koda : OpenStack, oVirt, RHEVM, VirtualBox, Xen, XenServer, itd.



Slika 17. Virtualizacija mreže

- **Virtualizacija skladišta** - Virtualizacija skladišta predstavlja apstrakciju skladišta koja se u logičkoj formi prezentuje korisniku i ona se može razlikovati od fizičke forme. Gledano iz ove perspektive, apstrakcija sakriva detalje na koji način je organizovano fizčko skladište u pozadini. Pored homogenizovanog fizičkog skladišta u pozadini moguće je koristiti i heterogeno fizičko skladište sa diskovima različitih karakteristika. Takođe je moguće kombinovati magistrale koje se koriste u komunikaciji sa diskovima kao što su: FiberChannel, InifiniBand, SAS, SATA, SCSI, itd. Takođe u virtualnom skladištu je moguće koristiti tehnologije zaštite podataka poput RAID-a na bilo kom nivou virtualizacije skladišta.

5.4 Komponente multibiometrijskog ekosistema

Cilj multibiometrijskog ekosistema je da obezbedi standardizovano okruženje u kome će dalji razvoj biometrijskih sistema u distribuiranom okruženju biti moguć. Takođe je potrebno da obezbedi jednostavno upravljanje nad podacima multibiometrijskog sistema kako u razvojnoj tako i produkcionoj fazi. Upravljanje hardverskim resursima je veoma važno u jednom multibiometrijskom sistemu. Raspoređivanje poslova u jednom distribuiranom okruženju može omogućiti jednostavniji razvoj algoritama distribuirane obrade podataka. Ovako koncipiran multibiometrijski sistem treba da omogući istraživačima implementaciju u bilo kom programskom okruženju. Razna programska okruženja imaju svoje specifičnosti koje definišu karakteristike okruženja u kome će biti pokrenuti. Agnostičnost na programske jezike i njihova izvršna okruženja je bitan činilac razvoja u naučnoj zajednici.

Multibiometrijski sistem se sastoji, kako je već navedeno, od niza modula koji se na programskom nivou reprezentuju kao servisi. Veze između modula multibiometrijskog sistema i programskih celina, odnosno servisa su prikazane u tabeli 2.

Modul multibiometrijskog sistema	Servis	Mesto na kome se izvršava
Biometrijski senzor	uzimanje biometrijskog uzorka	klijent sa senzorom
Izvlačenje karakteristika	obrada uzorka algoritmom za izračunavanje vektora karakteristika	MBio sistem
Poređenje	poređenje vektora karakteristika sa šablonom u bazi	MBio sistem
Odlučivanje	donošenje odluke na osnovu algoritma za fuziju rezultata poređenja	MBio sistem

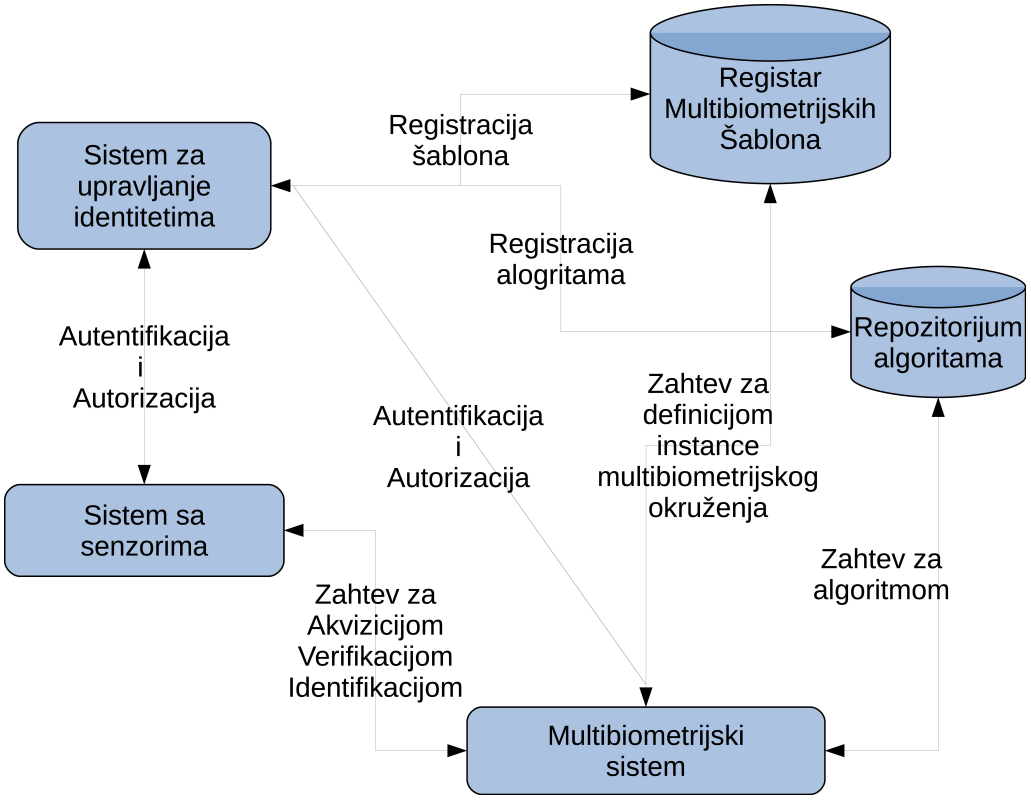
Tabela 2. Veze između modula multibiometrijskog sistema i servisa

Potrebno je napomenuti da modul za poređenje u trenutku izvršavanja treba da ima konekciju ka multibiometrijskoj bazi koja sadrži već unapred uzete uzorke korisnika datog multibiometrijskog sistema i/ili već sračunate šablone za korišćeni algoritam.

Takođe, trenutno ne postoji standardizovana arhitektura i shema multibiometrijske baze podataka te se koristi fajl sistem za čuvanje biometrijskih uzorka i podataka o njima.

Uzimanje biometrijskog uzorka je proces putem koga je multibiometrijski sistem u interakciji sa korisnikom. Taj proces se mora izvršavati na zasebnom računarskom sistemu koji je opremljen sa barem jednim biometrijskim senzorom. Da bi se uzeti biometrijski uzorak predao na dalju obradu potrebno je obezbediti povezanost tog računarskog sistema sa ostalim modulima koji će obraditi uzorak. U zavisnosti od složenosti procesa uzimanja biometrijskog uzorka potrebno je rasporediti senzore na više različitih računarskih sistema. Ovaj pristup nameće uvođenje servisa koji će autorizovati proces uzimanja i validaciju biometrijskog uzorka prilikom slanja ka ostalim modulima multibiometrijskog. Sistem za upravljanje identitima se nameće kao pogodan servis za autentifikaciju i autorizaciju. Identitet svakog računarskog sistema sa biometrijskim senzorom se može kreirati u sistemu za upravljanje identitetima. Time se autentifikacija i autorizacija uzimanja i slanja biometrijskog uzorka svodi na proveru identiteta računarskog sistema i njemu dodeljenih biometrijskih senzora nad kojima upravlja.

Moduli za izvlačenje karakteristika iz biometrijskog uzorka, poređenja i odlučivanja koriste različite algoritme putem kojih je definisan način rada svakog od tih modula. Mogućnost upravljanja nad algoritmima, njihovim implementacijama i radnim okruženjima je problem koji se rešava uvođenjem repozitorijuma algoritama. Za svaki algoritam je moguće specificirati programsko i radno okruženje u repozitorijumu. Za testiranje i unapređenje tih algoritama potrebno je obezbediti standardizovano radno okruženje koje će obezbediti jednostavno upravljanje. Trenutno tehnologije virtualizacije i računarstva u oblaku pružaju mogućnost kreiranja radnih okruženja korišćenjem šablona. Ti šabloni će biti registrovani u repozitorijumu algoritama koji će sadražati i specifikaciju potrebnih programskih paketa.



Slika 18. Konceptualni model multibiometrijskog ekosistema

Na osnovu mesta izvršavanja modula multibiometrijskog sistema prikazanih u tabeli 2 na slici 18 je predstavljen konceptualni model multibiometrijskog ekosistema. Osnovne komponente datog modela su:

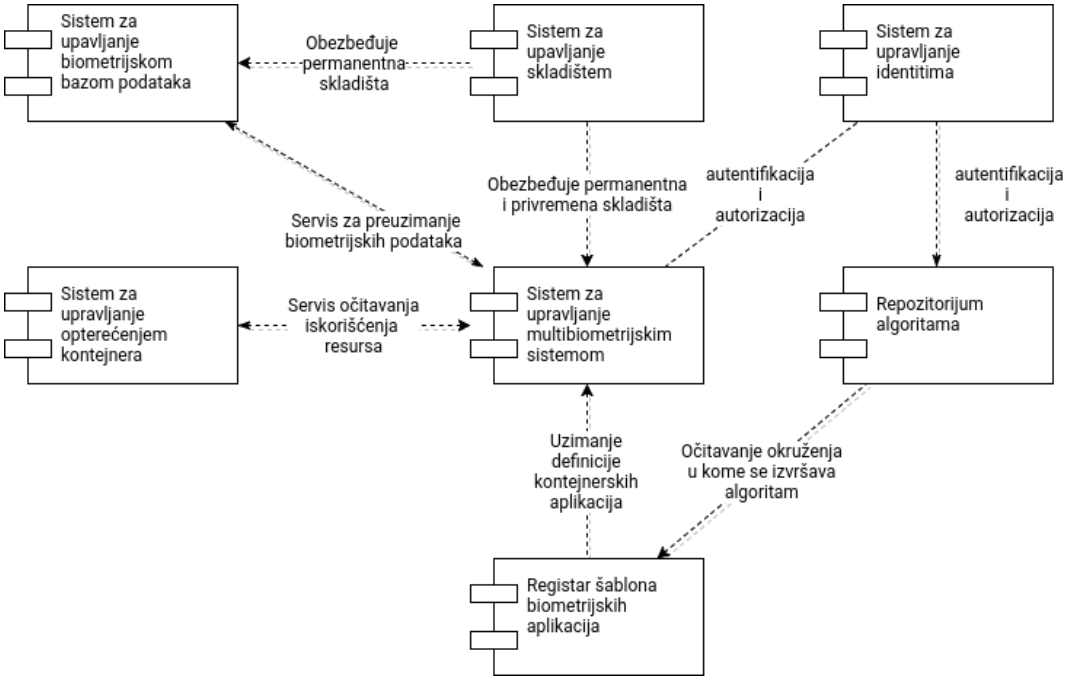
- **Sistem za upravljanje identitetima** - U sistemu za upravljanje identitema identitet ne mora biti samo identitet osobe. Hardver i softver takođe mogu dobiti svoj identitet u sistemu za upravljanjem identitema, kao što je već navedeno u drugom poglavlju. Pored autentifikacije koja će odrediti identitet hardvera, odnosno senzora, svaki od šablona multibiometrijskog sistema i algoritama koji se koriste u njima će dobit svoj identitet. Tokom autorizacije navedenih elemenata u ekosistemu se mogu opisati veze između pojedinačnih elemenata u svakoj instanci multibiometrijskog sistema. Takođe kroz sistem kontrole pristupa, koji je deo sistema za upravljanje identitetima se mogu definisati prava koja može imati svaki element ekosistema tokom izvršavanja jedne instance multibiometrijskog ekosistema.
- **Sistem sa senzorima** - Zbog specifičnosti akvizicije biometrijskih modaliteta ne postoji biometrijski senzor koji može očitati bilo koji biometrijski uzorak. Sistem sa senzorima predstavlja skup svih senzora raspoloživih multibiometrijskom sistemu, a oni mogu biti povezani na jedan ili više

različitih računarskih sistema. Svaki računarski sistem sa pripadajućim senzorima može biti autentifikovan od strane sistema za upravljanje identitetima a svaki njego senzor biti autorizavan za uzimanje biometrijskog uzorka za odgovarajući biometrijski modalitet za koji je on sposoban.

- **Registar multibiometrijskih šablona** - Sve instance multibiometrijskog sistema ne moraju biti iste u trenutku izvršavanja. U multibiometrijskom šablonu se mogu opisati specifičnosti jednog multibiometrijskog sistema. Na osnovu tih specifičnosti se može izgraditi identitet koji će jednoznačno autentifikovati tip multibiometrijskog sistema kao i autorizovati senzore neophodne za funkcionisanje te instance multibiometrijskog sistema, naravno ukoliko su zahtevani senzori dostupni u vremene izvršavanje te instance.
- **Repozitorijum alogritama** - svaki registrovani alogritam pored implementacije u nekom programskom jeziku će sadržati i specifikaciju izvršnog okruženja u kome će se izvršavati. Prilikom registracije algoritma potrebno je definisati modul multibiometrijskog sistema za koji je on namenjen. Ove informacije će biti relevantne prilikom autorizacije modula multibiometrijskog sistema koja će modulu dati spisak mogućih alogritama koji su raspoloživi u trenutku izvršavanja.
- **Multibiometrijski sistem** - predstavlja instancu jednog od multibiometrijskih sistema registrovanih šablona u registru. Svaka instanca može biti autentifikovana od strane sistema za upravljanje identitetima. U registrovanom šablonu se specificira izvršno okruženje kako sa strane potrebnih hardverskih resursa tako i nepohodnog skupa softverskih paketa i komponenti. U tom šablonu se takođe navodi distribuiranost modula multibiometrijskog sistema. Informacije o distribuiranosti koje su na taj način opisane će omogućiti izgradnju multibiometrijskog sistema i alocirati potrebne resurse na jednom ili više računarskih sistema i konfigurisati nepohodne veze među modulima te instance tog multibiometrijskog sistema.

6 Predlog generičke arhitekture multibiometrijskog sistema

Na osnovu konceptualnog modela predložena je generička arhitektura multibiometrijskog ekosistema data na slici 19.



Slika 19. Model generičke arhitekture multibiometrijskog ekosistema

Multibiometrijski sistem je izgrađen od sledećih komponenti:

- **Sistem za upravljanje identitetima** - Osnovna uloga ovog sistema je da vodi računa o korisnicima sistema, pripadnosti grupama, dodeljenim ulogama, polisama i pravima koja su im dodeljena za pristup resursima sistema. Korisnike multibiometrijskog sistema možemo podeliti na :
  - Administratorske korisnike : Oni imaju pravo raspolaganja na sve resurse sistema do maksimuma fizičke raspoloživosti svakog pojedinačnog resursa. Mogu da upravljaju nad svim identitema u sistemu, da upravljaju grupama, ulogama, polisama i pravima
  - Sistemske korisnike : Korisnici koji upravljaju nad pojedinačnim resursom multibiometrijskog sistema koji im je dodeljen. Radi bezbednosti u multibiometrijskom sistemu svaka komponenta ima svog korisnika pod kojim se startuje na operativnom sistemu

- Napredne korisnike : Oni mogu da koriste multibiometrijski sistem sa već predefinisanim resursima. Njihovi biometrijski podaci se mogu koristiti samo za njihovu ličnu autentifikaciju prilikom pristupa multibiometrijskom sistemu
- Eksterni Korisnici : Njihovi biometrijski podaci određuju identitet nad kojim multibiometrijski sistem radi

Pod resursima sistema se podrazumevaju:

- Skladište : na koja skladišta ili delove skladišta korisnik ima pravo, koji tip skladišta može da dobije, koja je veličina skladišta odobrena
- Biometrijska baza podataka : Prava čitanja, pisanja i brisanja nad podacima koji su smešteni u bazu podataka
- Repozitorijum algoritama : Na osnovu uloga u multibiometrijskom sistemu korisnik može da koristi već postojeće algoritme, da dodaje nove, da određuje prava pristupa nad algoritmima kojima je on vlasnik
- Registar šablona biometrijskih aplikacija : Na osnovu uloga u multibiometrijskom sistemu, kojim šablonima korisnik ima pravo da pristupi, da delove postojećih šablona iskoristi za pravljene sopstvenih, da definiše prava pristupa za sopstvene šablone
- Sistem za upravljanje opterećenjem kontejenera : Da li korisnik na osnovu uloga koje ima u multibiometrijskom sistemu može da koristi već predefinisane kvote nad resursima sistema ili da definiše kvote koje nisu veće od predefinisanih
- Akvizicioni modul sa senzorima : Da li korisnik može da koristi senzore za unos novih podataka u biometrijsku bazu podataka, ili će senzore koristiti za akvizicuju nekog od modaliteta sa kojima senzori rade te iskoristiti kroz neki od algoritama aplikaciji

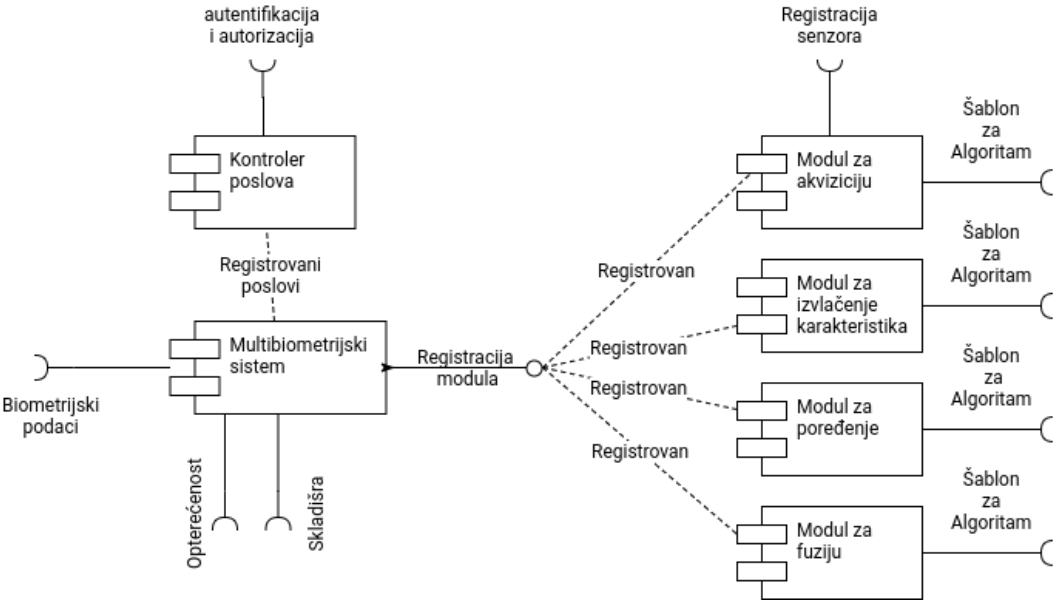
- **Sistem za upravljanje biometrijskom bazom podataka** - Sadrži biometrijske uzorke, kao i njihove vektore karakteristika razvrstane po algoritmima koji se nalaze u repozitorijumu algoritama. Biometrijska baza podataka sadži dva tipa informacija : *struktuirane* podatke koji se dobijaju od strane sistema za upravljanje identitetima, podacima o modalitetu uzorka, podacima o multimedijalnom formatu zapisa uzorka, referencu koju dobija od strane sistema za upravljanje skladištem i *nestruktuirane* podatke koji predstavljaju sadržaj multimedijalnog zapisa uzetog uzorka.

- **Sistem za upravljanje skladištem** - Skladišta se mogu podeliti na perzistentna i privremena. Perzistentna skladišta služe za podatke biometrijske baze, repozitorijuma algoritama, definicija šablona biometrijskih aplikacija, dok se privremena skladišta koriste za skladišta koja koriste kontejnerske aplikacije
- **Repozitorijum algoritama** - Predstavlja bazu podataka o algoritmima koji mogu da se koriste u kontejnerskm aplikacijama. Sadrži podatke tipu algoritma, radnom okruženju u kome algoritam radi, referenci na izvorni oblik alogritma, programski jezik u kome je pisan, zavisnosti ka programskim bibliotekama koje su neophodne prilikom kompajliranja, linkovanja i izvršavanja, komandama za njegovo prevođenje u izvršni kod, karakteristikama za dati tip algoritma.
- **Registar šablona biometrijskih aplikacija** - Predstavlja privatni registar šablona kontejnerskih aplikacija. Deli se na registar sistemskih kontejnera i registar šablona koji se koriste za biometrijske kontejnerske aplikacije. Registar sistemskih kontejnera je skup definicija pod-ova i pripadajućih kontejnera koji omogućuju instalaciju i konfiguraciju komponenti multibiometrijskog sistema:
  - Sistema za upravljanje identitetima
  - Sistema za upravljanjem biometrijskom bazom podataka
  - Sistema za upravljanjem skladištem
  - Repozitorijuma algoritama
  - Registra šablona biometrijskih aplikacija
  - Sistema za upravljanje opterećenjem kontejenra
  - Sistema za upravljanje multibiometrijskim sistemom

  Osnovni zadatak registra šablona biometrijskih aplikacija je da omogući Kubernetes-ima instalaciju i konfiguraciju multibiometrijskog sistema u kontejnerskom radnom okruženju.
- **Sistem za upravljanje opterećenjem kontejnera** - Koristeći mogućnosti imenskog prostora i kontrolnih grupa kernela operativnog sistema, sistem za upravljanjem opterećenjem kontejnera prijavljuje multibiometrijski sistem, tj. njegove kontejnere operativnom sistemu na kome se izvršavaju. Podešava kvote za CPU, RAM, disk operacije i mrežni podsistem. Na osnovu definisanih

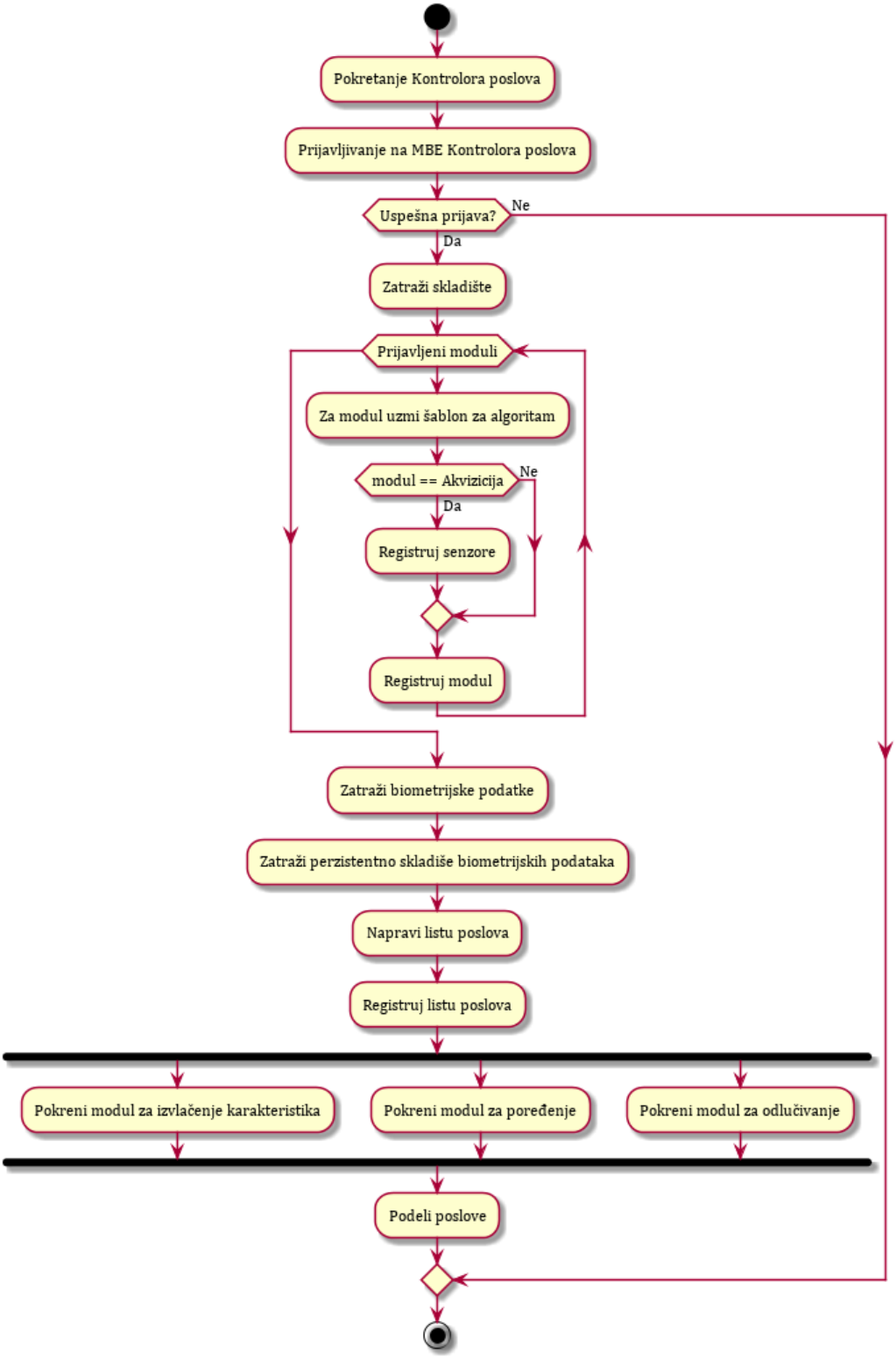
vrednosti, koristeći Kubernetes infrastrukturu, upravlja startovanjem, migracijom, zasustavljanjem, novih repliciranih pod-ova na računarima, bez obzira da li su to fizički računari ili virtualni računari, koji se nalaze u klasteru.

- **Sistem za upravljanje multibiometrijskim sistemom** - Multibiometrijski ekosistem se sastoji od osnovnih modula navedenih u tabeli 2. Stoga je potrebno obezbediti određeni skup *pod*-ova sa pripadajućim kontejnerima koji učitavaju i registrijuju module sa pripadajućim algoritmima, uspostavljaju neophodne veze ka ostalim komponentama multibiometrijskog sistema. U jednom distribuiranom sistemu moguće je istovremeno pokrenuti više različitih instanci multibiometrijskog sistema tako da svaka instanca radi sa različitim algoritmima u svakom biometrijskom modulu. Model jedne instance multibiometrijskog sistema je dat na slici 20.



Slika 20. Model instance multibiometrijskog sistema

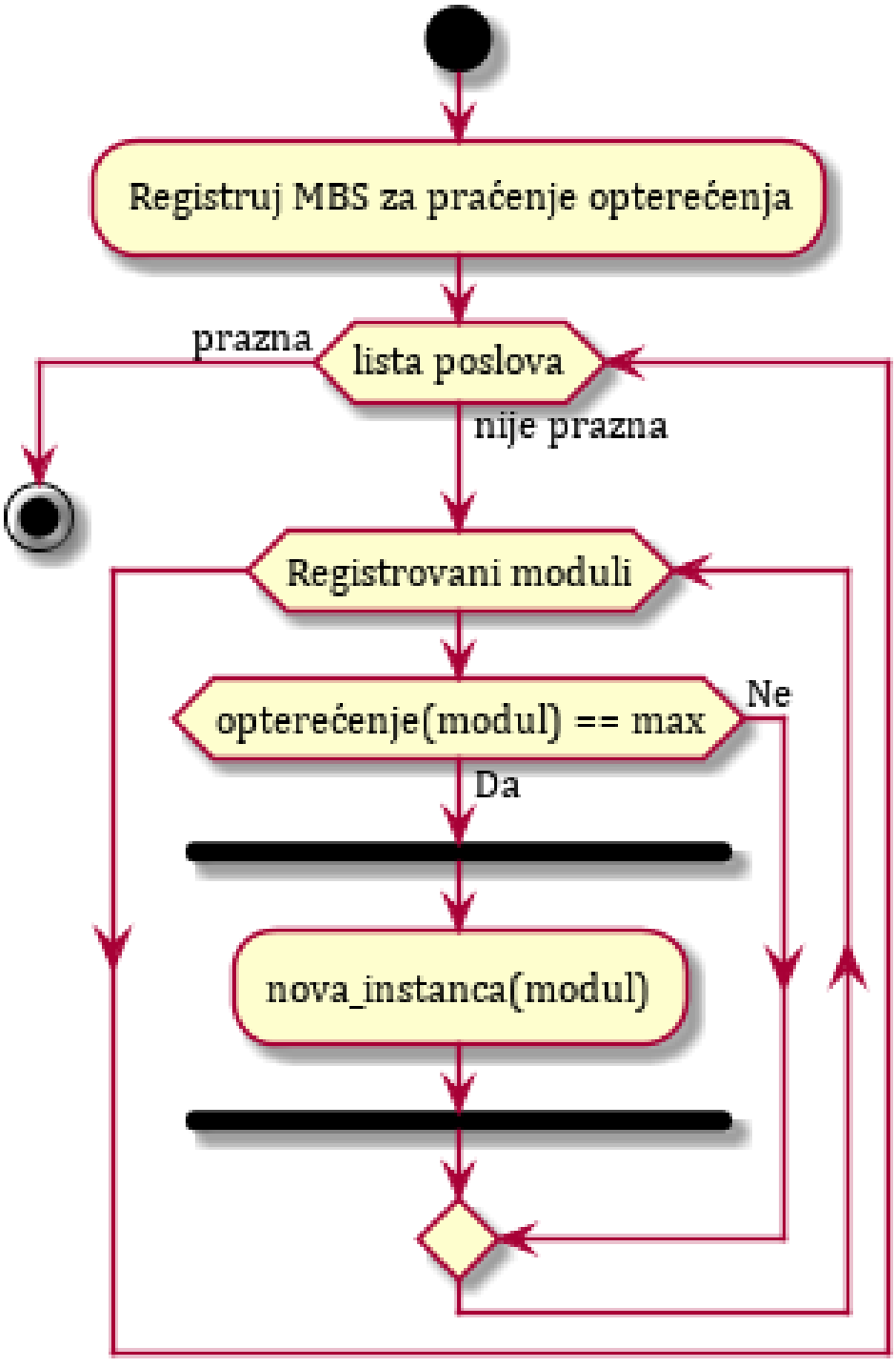
Aktivnosti koje multibiometrijski sistem treba da obavi za svaku instancu su prikazane na slici 21.



Slika 21. Aktivnosti sistema za upravljanje multibiometrijskim sistemom prilikom pokretanja jedne instance multibiometrijskog sistema

Stoga je potrebno registrovati svaki biometrijski modul sa odgvorajaćim šablonom algoritma koji će se koristiti za datu instancu. Multibiometrijski sistem je potrebno da obezbedi odgovarajuća skladišta:

- privremeno: za svaki registrovani biometrijski modul radi obrade biometrijskih uzoraka
- permanentno: sadrži fajlove u kome se nalazi multimedijalni sadržaj biometrijskih uzoraka



Slika 22. Aktivnosti sistema za upravljanje multibiometrijskim sistemom prilikom nadgledanja multibiometrijskog sistema

Pored toga potrebno je da obezbedi i biometrijske podatke, odnosno skup biometrijskih podataka nad koji će mutlibiometrijski sistem obrađivati. Na

osnovu tog skupa podataka definiše listu poslova i predaje ih kontroleru poslova, koji će poslove razvrstavati ka biometrijskim modulima. Svaki biometrijski modul u jednom trenutku vremena može da obradi samo jedan biometrijski uzorak. Radi postizanja maksimalnog opterećenja distribuiranog sistema, kada jedan biometrijski modul dostigne zadati utrošak barem jednog od dodeljenih mu resursa, multibiometrijski sistem će startrovati narednu kopiju tog modula u klasteru ukoliko ima dovoljno sistemskih resursa da alocira. Aktivnosti na nadgledanju opterećenja date su na slici 22.

Na ovom nivou apstahovan je računarski sistem u kome će se izvršavati aplikacije multibiometrijskog sistema. Time je dobijena sloboda u izboru potrebnih hardverskih komponenti koje će moći da se izaberu na osnovu parametara kao što su: cena koštanja hardverske platforme, performanse procesora, memorije, količine diks prostora za skladište podataka, itd. Takođe ovakva arhitektura omogućuje skalabilnost multibiometrijskog sistema kako horizontalno na nivou performasi računarskih sistema nad kojima se izvršavaju aplikacije multibiometrijskog sistema tako i vertikalno jer se može izabrati ciljno opterećenje sistema kroz upravljanje kvotama opterećenja procesora i količine memorije koju će dobiti aplikacije multibiometrijskog sistema. Ukoliko jedan računar, bio on fizički ili virtualni u nekom od izabranog sistema virtualizacije ili u oblaku, ne dostiže zahtevane performanse moguće je multibiometrijskom sistemu dodati barem još jedan računarski sistem bilo kog tipa.

7 Model radnog okruženja distribuiranog multibiometrijskog sistema i izgradnja odgovarajućeg repozitorijuma

Radno okruženje u kome će se izvršavati multibiometrijski sistem potrebno je da obezbedi pre svega distribuiranost, skalabilnost kako horizontalnu tako i vertikalnu. Skalabilnost se može definisati kao sposobnost nekog sistema da prilagodi problemu jer se obim tog problema povećava (broj elemenata ili predmeta, porast obima rada i/ili se očekuje povećanje obima) [99]. Mogućnost skaliranja sistema može da zavisi od njegovog dizajna, tipova podataka koji se koriste, algoritam ili komunikacionih mehanizama koji se koriste u komponentama tog sistema. Takođe postoji više različitih tipova skalabilnosti [99] :

- ***Load Scalability*** : Kada je sistem u mogućnosti da dobro iskoristi resurse pri različitim nivoima opterećenja. Faktori koji utiču na skalabilnost mogu biti: loše korišćenje paralelizma, neadekvatno raspoređivanje ili prekomerno preklapanje deljivih resursa.
- ***Space Skalability*** : Mogućnost sistetema da održi potrošnju sistemskih resursa na prihvatljivom nivou kada dođe do rasta opterećenja.
- ***Structural Scalability*** : Implementirani standardi u sistemu dozvoljavaju povećanje broja objekata nad kojima upravlja sistem ili će to učiniti u okviru zadatog vremenskog intervala.

Faktore koji utiču na povećanje ili smanjenje skalabilnosti veoma je teško identifikovati čak i u slučaju ciljane skalabilnosti. O skalabilnosti treba voditi računa još prilikom dizajniranja arhitekture sistema. Dobar dizajn arhitekture sistema treba da omogući:

- **Vertikalno skaliranje** : dodavanje veće računarske snage (više procera, memorije, propusne moći itd) u uređajima koji se koriste u sistemu.
- **Horizontalno skaliranje** : dodavanje više replika istog softvera ili hardverskih resursa.

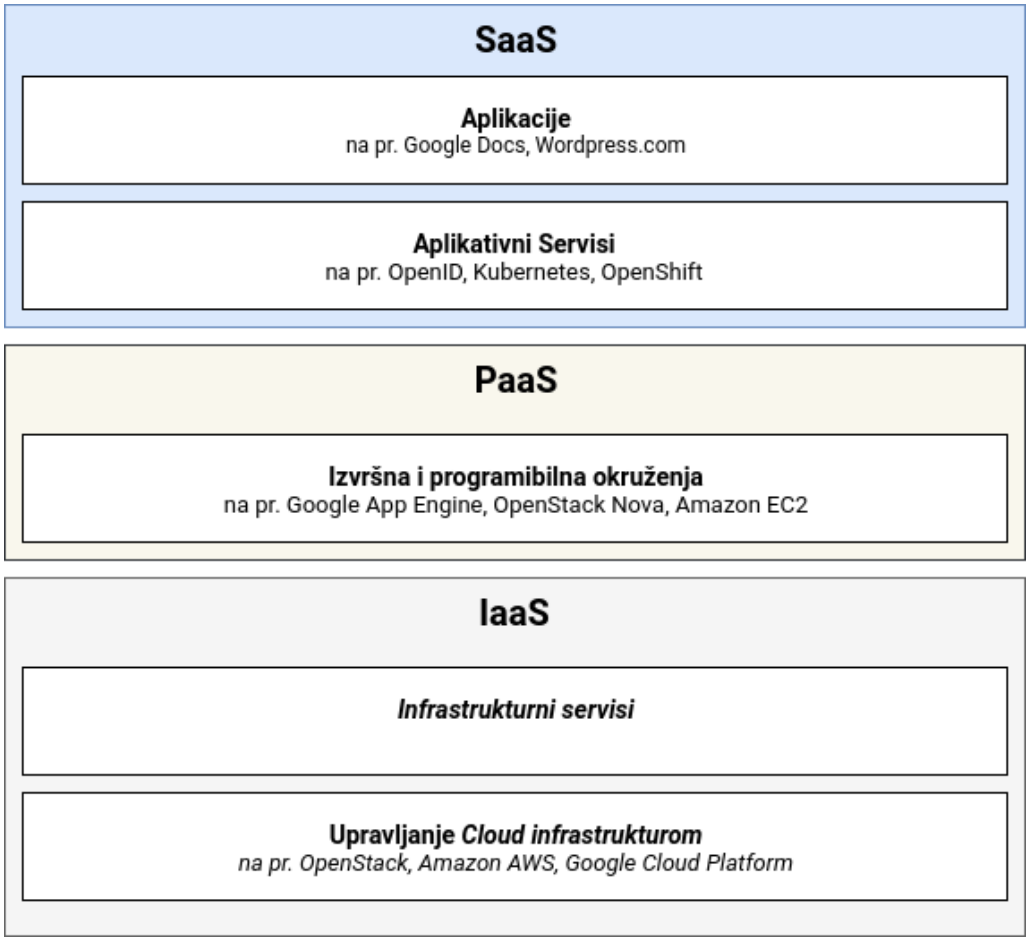
Skalabilnost *cloud* okruženja se zasniva na tri osnovne funkcionalnosti:

1. Virtualizacija : smanjuje kompleksnost sitema, standradizuje hardversku platformu i smanjuje troškove upravljanja resursima

2. Deljivost resursa : deljenje računarskih resursa između različitih aplikacija i/ili organizacija će omogućiti optimizaciju optimizaciju njihove uptrebe izbegavajući prazan hod

3. Dinamička alokacija : obezbeđivanje treba postaviti da radi na zahtev, takođe treba da bude automatsko u toku rada sistema. Ovakav način rada podrazumeva potrebu praćenja performansi usluga i automatizaciju odlučivanja i akcija kako bi se odgovorilo na povećanje odnosno smanjenje radnog opterećenja.

Do sada je predlagano više arhitektura, Buyya predlaže arhitekturu *cloud*-a zasnovanu na QoS-u [100]. Na osnovu te arhitekture Brandic daje kompletniju arhitekturu [101]. Iako se ove arhitekture pre svega oslanjaju QoS i SLA koji su značajni faktori u pogledu upliva kapitala u *Cloud* rešenja i prozvode, neke od osnovnih zajedničkih elemenata je idenitfikovao Lenk [102]. Uporednom analizom preloženih arhitektura se može doći do arhitekture *Cloud*-a predstavljene na slici 23



Slika 23. Arhitektura *Cloud*-a

7.1 Kontejneri

Kontejneri (*Containers* eng.) predstavljaju koncept zasnovan na imenskim prostorima (*namespaces* eng.) i kontrolnim grupama (*cgroup* eng.) u kernelu operativnog sistema. Time je omogućena virtualizacija na nivou operativnog sistema kroz laganu procesnu virtualizaciju (*lightweigth process virtualization* eng.) Ona daje privid korisnuku da se proces izvršava na operativnom sistemu sa sopstvenim kernelom. Moguće je pokrenuti mnoštvo takvih procesa koji imaju privid da se svaki od njih izvršava nad soptsvenim kernelom. Ustvari dele jedan te isti kernel koji je pokrenut na računaru.

Imenski prostori omogućuju izolaciju jednog ili više procesa tako da oni imaju drugačiji pogled na sistem od ostalih procesa. Paradigma izolacije procesa nije nova, postoji od 1992. godine u implementaciji Plan9 distribuiranog operativnog sistema nastala u Istraživačkom centru AT&T-a [103]. Postoji više imenskih prostora [104]:

- mnt (mount points, filesystems)
- pid (procesi)
- net (network stek)
- ipc (System V IPC)
- uts (hostname)
- user (UID i GID)
- security (selinux)
- device
- cgroup

Implementacija imenskih prostora se zasniva na odgovarajaćim sistemskim pozivima u kernelu operativnog sistema [105]:

- clone() - kreira novi proces i dodaje ga u novi ili već postojeći imenski prostor
- unshare() - kreira novi imenski prostor i dodaje tekući proces u njega
- setns() - omogućuje procesu da se pridruži novom imenskom prostoru

Kontrolne grupe predstavljaju mehanizam za hierahijsku organizaciju procesa i distribuciju resursa kroz tu hijerarhiju na kotrolisan i konfigurabilan način. Kontrolne grupe su sastavljene iz dva dela:

- jezgra (*core* eng.) koje je zaduženo za hijerarhijsku organizaciju procesa
- kontrolera (*controller* eng.) koji je zadužen za distribuciju specifičnog tipa sistemskog resursa

Kontrolne grupe formiraju stablo tako da svaki proces može pripadati jednoj i samo jednoj kontrolnoj grupi u jednom trenutku vremena. Sve niti koje proces kreira pripadaju istoj kontrolnog grupi u kojoj se nalazi proces u trenutku kreiranja. Proces može da migrira iz jedne u drugu kontrolnu grupu. Migracija procesa ne utiče na već kreirane procese potomke procesa koji se migrira. Poštujući odgovarajuća strukturna ograničenja kontroleri mogu biti za jednu kontrolnu grupu selektivno uključeni ili isključeni. Ako se kontroler uključi za jednu kontrolnu grupu to utiče na sve procese koji pripadaju toj kontrolnoj grupi uključujući i sva podstabla te kontrolne grupe.

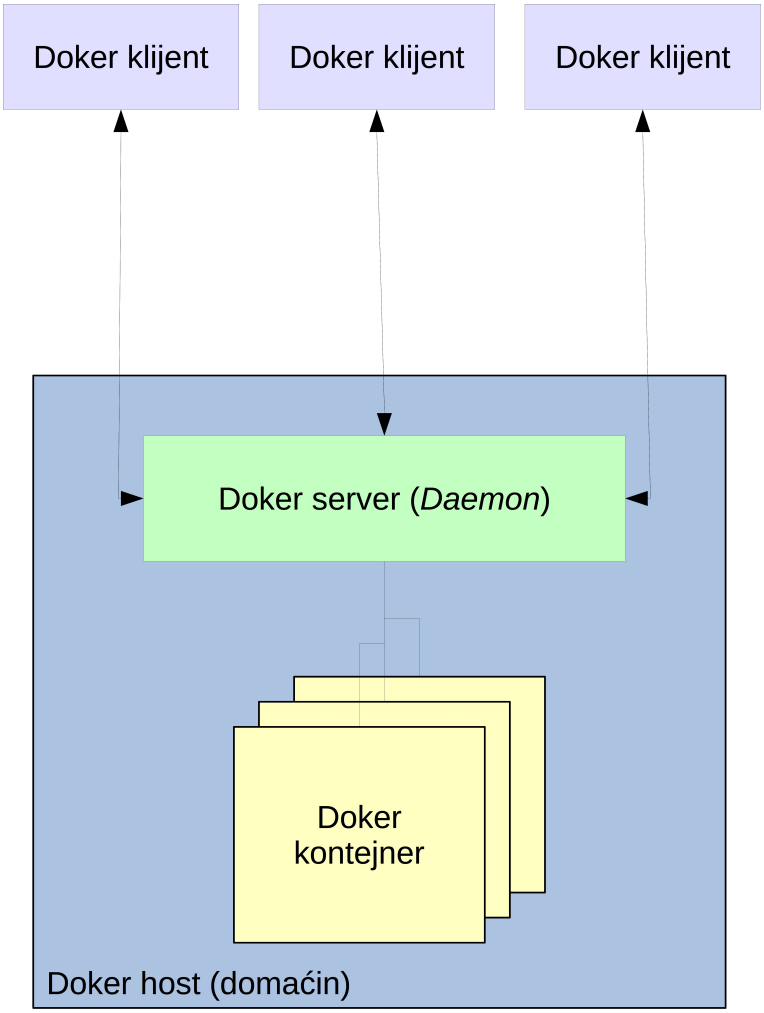
Zbog jednostavnije kontrole resursa procesa koji se pokreće u kontejneru potreban je skup alata kojima je moguće upravljati kontejnerom, sistemskim resursima i dodeljenim procesima. Do danas je razvijen veći broj alati ali nemaju svi alati kontrolu nad svim sistemskim resursima koje omogućuje imenski prostor kernela. Među njima se izdvajaju:

- Docker
- LXC
- OpenVZ
- Systemd-nspawn
- Linux-VServer
- Virtuozzo
- Kuberenetes

7.1.1 Doker

Doker (*Docker* eng.) predstavlja otvorenu virtualizacionu platformu zasnovanu na kontejnerima. On omogućuje automatizaciju instalacije aplikacija u kontejnere. Osnovne komponente su:

- **Doker klijent i server** : Doker je klijent server aplikacija. Doker kiljent komunicira preko RESTfull API-ja sa serverom. Klijent i server mogu biti instalirani na jednom računaru.



Slika 24. Arhitektura dokera

- **Doker šabloni** : Šabloni su osnovni gradivni elementi doker sveta. Kontejneri nastaju na osnovu šablona. Format šablona je nivojski i koriste neku od implementacija Union fajl sistema, koji omogućuju "Copy-On-Write" funkcionalnost. Time je omogućeno da se šablon gradi nivo po nivo koristeći seriju instrukcija. Na primer, doda se fajl; pokrene komanda; otvori TCP/IP port. Šabloni mogu biti javno objavljeni kao "izvorni kod" jer su portabilni, deljivi i nadogradivi.
- **Registri** : Doker čuva napravljene šablone u registrima. Postoji dva tipa registara: privatni i javni. Docker Inc upravlja javnim registrima šablona (*DockerHub*). Svaki registrovani korisnik može objaviti svoj šablon koji može biti deljen među korisnicima dokera. DockerHub ima registrovanih preko 10 000 šablona koje su korisnici napravili i podelili među sobom. U registru je moguće čuvati i privatne šablone koji sadrže privatne informacije zapisane u šablonu i time su zaštićene od neovlašćenog korišćenja i zloupotrebe. Za

privatne šablone može se koristiti DockeHub ili pokrenuti privatni registar u lokalnoj računarskoj mreži.

- **Doker kontejneri** : Kontejneri se kreiraju na osnovu šablona i mogu da sadrže jedan ili više procesa u vreme izvršavanja. Doker je pozajmio koncept kontejnera od špedicije gde se dobra transportuju globalno upakovani u kontejnere. U svom modelu kontejnera doker isporučuje softver. Doker kontejner se sastoji od:
  - Slike fajl sistema - Nastala je na osnovu šablona iz registra i sadrži neophodne fajlove koji omogućuju da se jedna i/ili više aplikacija izvršavaju u datom kontejneru.
  - Skup standardnih operacija - Doker podržava sledeće operacije nad kontejnerom: kreiranje, startovanje, zaustavljanje, restartovanje i brisanje. Doker ne zanima sadržaj kontejnera kada izvršava navedene operacije. Svaki kontejner se učitava na potpuno isti način bez obzira da li se radi o kontejneru koji u sebi ima web server, aplikativni server ili bazu podataka.
  - Izvršno okruženje - Doker ne zanima gde će se isporučiti kontejner. Na primer, kontejner se može napraviti na lokalnom računaru na osnovu javnih i/ili privatnih šablona, dograditi sopstvenim šablonom, a zatim snimiti taj šablon u registar. Takav registrovani šablon se može skinuti na fizički ili virtualni server, testirati u klasteru više računara u oblaku.

Za doker kontejner se može konstatovati da je on razmenljiv, portabilan, nadogradiv i generički koliko god je to moguće. Sa Dokerom je moguće brzo napraviti aplikativni server, web servis, server baze podataka, testirati sa nekim od standardnih alata za testiranje, na primer Jenkins, i pokrenuti onoliko puta koliko je to potrebno, na lokalnom računaru, serveru fizičkom i/ili virualnom, i iskoristiti pogodnosti virtalizovanih okruženja i računarstava u oblaku. Doker standardno koristi sleće tehnologije, tehnike i komponente:

- libcontainer - biblioteka u kojoj je opisan format kontejnera, incijalno nastala pod Linuks operativnim sistemom
- Linuks kernel imenski prostor, koji obezbeđuje izolaciju fajl sistema, procesa, memorije i mreže
- Izolaciju fajl sistema - svaki kontejner sadrži svoj *root* fajl sistem

- Izolaciju procesa - svaki kontejner sa svim svojim procesima se pokreće u svom okruženju
- Izolacija mreže - odvojeni virualni mrežni adapteri i IP adresni prostor između svakog kontejnera
- Izolacija resursa i grupisanje - resursi poput memorije i procesora se alociraju zasebno za svaki doker kontejner koristeći kontrolne grupe (*cgroups*) funkcionalnosti kernela
- Copy-On-Write - fajl sistem se kreira sa copy-on-write funkcionalnošću, što omogućeva više nivojsku organizaciju fajl sistema, koja se odlikuje brzinom i zahteva manji i ograničeni disk prostor
- Vođenje dnevnika (*logging*) - STDOUT, STDERR i STDIN iz kontejnera se sakupljaju i beleže u radni dnevnik koji je dostupan za analizu ili rešavanje problema nastalih tokom izvršavanja aplikacija u kontejneru
- Interaktivni shell - Moguće je kreirati pseudo terminal preko koga se može pristupiti direktno komandama koje kontejner vidi

Iako je Doker inicijalno nastao na Linuks operativnom sistemu danas je moguće instalirati Docker okruženje i pod MS Windows i OS X operativnim sistemima, međutim zbog nedostatka funkcionalnosti na nivou kernela navedenih operativnih sistema, Docker nema sve funckionalnosti koje ima na Linuks operativnim sistemima te se zbog toga ne preporučuje za produkciono okruženje.

7.2 Kuberneti

Kuberneti (*Kubernetes* eng. preuzeli iz grčkog κυβερνήτης - jednina: brodski pilot, guverner po Vuku bi bilo sa engleskog kubernetis a sa grčkog kuvernitis) su sistem otvorenog koda za automatsko raspoređivanje, skaliranje i upravljanje kontejnerskim aplikacijama u klasteru računara. Kuberneti (*Kubernetes* eng.) je prenosiva i proširiva platforma otvorenog koda za upravljanje kontejnerskim aplikacijama, njihovim radnim opterećenjem i servisima, koja olakšava deklarativnu konfiguraciju i automatizaciju. Kuberneti definišu skup gradivnih blokova (”primitiva”) koji zajedno obezbeđuju mehanizam za upravljanje i skaliranje aplikacija. Komponente koje ulaze u skup primitiva su dizajnirane kao slabo spojene (”loosely coupled”) i proširive kako bi omogućile širok spektar različitih radnih opterećenja. Proširivost je obezbeđena kroz API. Time oni spadaju u PaaS (*Platform*

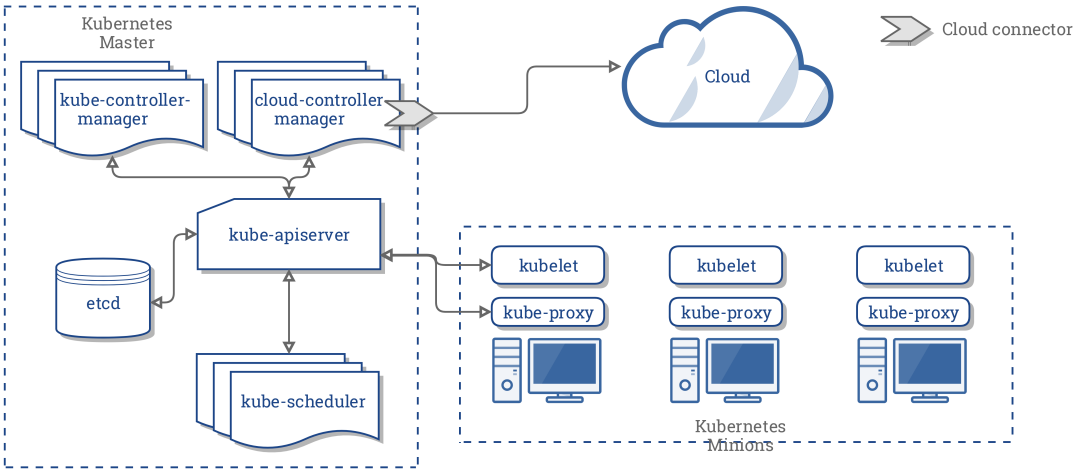
as a Service eng.) kategoriju računarstva u oblaku (čloud computing"bolji prevod?). Primitive od kojih se gradi su (TODO referenca na dokumentaciju kuberneta):

- **Pod** : apstrakcija skupa kontejnera koji su čvrsto vezani ("thigly coupled") preko deljivih resursa: mrežnog interfejsa i sistema skladišta. Kroz ovu apstrakciju daje se perzistentnost prilikom raspoređivanja pojedinačnog kontejnera sa sledećim ograničenjima:
  - pod se izvršava na jednom računaru sa svim kontejnerima koji mu pripadaju
  - pod dobija svoju lokalnu IP adresu u klaster mreži i svi kontejneri dele mrežne portove koji su dodeljeni toj IP adresi. Implikacija ovog ograničenja je da nije moguće da se u jednom podu nađu kontejneri u kojima se nalaze servisi koji slušaju na istom portu
  - pod se može replicirati na više računara da bi se obezbedila skalabilnost i tolerancija na grešku
- **Labele i selektori** : predstavljaju osnovni mehanizam grupisanja koji se koristi za određivanje nad kojim komponentama se određena operacija primenjuje
- **Kontroleri** : Upravljaju stanjem klastera podova. Sastoji se od više tipova kontrolera:
  - *Kontroler replikacije* : obezbeđuje skaliranje i replikaciju kroz klaster tako da se u njemu uvek nalazi tačno naznačen broj kopija jednog poda
  - *Kontroler poslova* : Upravlja izvršavanjem podova i aplikacija u njegovim kontejnerima do njihovogzavršetka
  - *ĐaemonSet Controller"* : Obezbeđuje da se na svakom računaru u nekom podskupu računara u klasteru izvršava samo jedan pod
- **Servisi** : Skup podova koji rade zajedno. Taj skup je definisan kao servis putem selektora labela. Rutiranje i otkrivanje (*service discovery* eng.) se obezbeđuju dodeljivanje fiksne IP adrese i DNS imenaza dati servis, balansiranje mrežnog saobraćaja između repliciranih kopija poda čak i slučaju kada se dogodi otkaz računara a pod je migriran na drugi računar u klasteru

Kuberneti su zasnovani na master-slave arhitekturi. Komponente se mogu podeliti na one upravljaju pojedinačnim čvorom, računarom u klasteru, i na one koje su deo kontrolne oblasti. Mogu se izdvojiti sledeće komponente koje pripadaju arhitekturi:

- **Master** : Predstavlja osnovnu jedinicu koja upravlja klasterom, njegovim opterećenjem i direktno komunicira sa računarima klastera. Kontrolna oblast Mastera se sastoji od više različitih komponenti gde svaka od njih ima svoj jedinstveni proces nas sistemu koji može da se izvršava na jednom master čvoru ili na više master čvorova u slučaju klastera visoke raspoloživosti (*high-availability* eng.)
- **etcd** : Predstavlja perzistentno distribuirano skladište u kome su zapisani konfiguracioni podaci klastera, kao i podaci o stanju klastera u datom trenutku vremena.
- **API server** : Ključna komponenta u arhitekturi kuberneta. Realizovan je kao RESTfull Web servis preko koga se upravlja podacima koji se čuvaju u etcd komponenti
- **Planer** : Prati stanje i upotrebu resursa i obezbeđuje izvršavanje poda na čvoru u klasteru na osvnovu dostupnosti resursa. Zbog toga planer mora da zna zahteve za resursima svakog poda koji je potrebno izvršiti u datom klasteru.
- **Menadžer kotrolera** : Jeste proces preko koga kontroleri komuniciraju putem API-ja radi kreiranja, ažuriranja i brisanja resursa nad kojima upravljaju.
- **Čvor** : Ili Minion predstavlja računar na kome su kontejneri raspoređeni. Svaki čvor u klasteru je potrebno da pokrene neki od alata u kojima je moguće izvršavati kontejner. Svi čvorovi moraju da imaju pokrenut isti alat. Takođe se na njemu izvršavaju i sve komponente koje omogućavaju mrežnu komunikaciju sa masterom.
- **Kubelet** : Odgovoran je za stanje svakog čvora obezbeđujući da su svi kontejneri u ispravnom stanju. Takođe je zadužen za pokretanje, zaustavljanje, i upravljanje nad kontejnerskim aplikacijama. Prati stanje poda i ukoliko nije u odgovarajućem stanju, pod će biti preraspoređen na isti ili neki od čvorova klastera koji poseduje odgovarajuće resurse za njegovo izvršavanje.
- **Kontejner** : je najniži nivo mikro-servisa koji obezbeđuje izvršavanje aplikacije, odgovarajuće biblioteke i njihove zavisnosti ka drugim aplikacijama ili bibliotekama.

- **cAdvisor** : Agent koji nadgleda i sakuplja podatke o preformansi i utrošku resursa kao što su CPU, memorija, iskorišćenost fajl sistema i mreže jednog kontejnera na svakom čvoru ponaosob.



Slika 25. Arhitektura Kuberneta

7.3 Radno okruženje multibiometrijskog sistema

Prilikom dizajna arhitekture radnog okruženja sa ciljem da se osigura vertikalana i horizontalna skalabilnost multibiometrijskog sistema izbor je sveden na korišćenje virtualnih računara i/ili kontejnera. Osnovni problem koji je trebalo rešiti prilikom dizajna navedene arhitekture je kako osigurati izolaciju configuracija servisa i aplikacija. Deljenje resursa na na nivou operativnog sistema predstavlja rizik jer svaki mehanizam deljenja resursa otvara mogući put preko koga mogu da procure informacije čak i među aplikacijama različitih korisnika i mora biti dizajniran sa velikom pažnjom da se ne naruši sigurnost podataka, aplikacija i sistema [106]. Kada se instaliraju različite aplikacije na jednom operativnom sistemu cena koštanja administracije može veoma lako da prevaziđe cenu koštanja samog softvera.

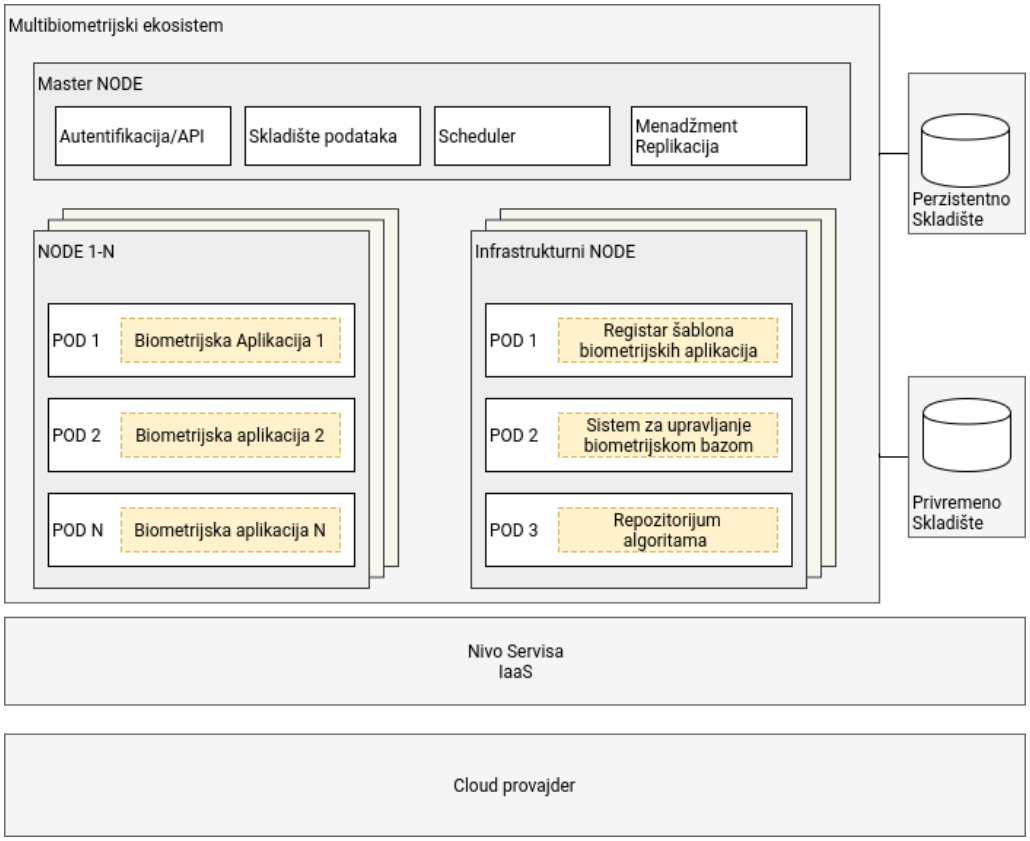
Ove slabosti u današnjim serverskim operativnim sistemima su dovele administratore i programere do toga da se uprosti instalacija i konfiguracija tako da se za svaku aplikaciju koristi zasebna kopija operativnog sistema u kome će se ona izvršavati, bez obzira da li se radi o namenskom serveru ili virtualnom računaru. Ovakav tip izolacije dovodi do toga da se bilo koji kod, podaci ili konfiguracije moraju eksplicitno konfigurisati prilikom njihovog deljenja.

Prilikom konsolidacije servera i servisa preduzeća, infrastrukura i opterećenje sistema pripadaju istoj kompaniji. U *aaS okruženju veza između provajdera i korisnika je na dugom štapu, tako da postaje teško uočiti i ispraviti anomalije koje mogu nastati tokom rada. Sistem virtualizacije treba da reši u *aaS okruženjima izolaciju resursa a da ne ugrozi performanse resursa koji se daju krajnjem korisniku na korišćenje.

S obzirom na dostupnost softvera za virtualizaciju, kao i na mogućnosti, instalaciju i konfigurabilnost izbor je sledeći :

- **KVM** : *Kernel Virtual Machine* pripada tipu virtualizacije pune virtualizacije koja dozvoljava da se *hypervisor* ponaša kao tip 1 [107]. Poseduje mogućnost vertikalne sklabilnosti, a uz odgavarajuće sistemske aplikacije i servise i horizontalnu sklabilnosti (OpenStack, oVirt ...:). Omogućuje statičku dodelu broja virtualnih procesora (vCPU) kao i dodeljivanje minimalne i maksimalne količine memorije koji će kreirani virtualni računar dobiti od strane operativnog sistema. Kako je virtualni računar samo jedan od procesa koji se izvršava u operativnom sistemu, mogu se dodeliti kontrolne grupe i imenski prostori tako da je izolacija procesa u kome se izvršava virtualni računar sa svojim operativnim sistemom izolovan od ostalih procesa u sistemu. Među ostalim procesima sistema može biti i više različitih virtualnih računara i/ili replika istog virtualnog računara.
- **Kontejneri** : Pripadaju grupi virtualizacije na nivou operativnog sistema. Karakteristike kontejnera su već opisane prethodno.

Korišćenje oba tipa virtualizacije omogućuje da arhitektura radnog okruženja ima skalabilnost i vertikalnu i horizontalnu, a takođe omogućuje da u okviru *Cloud*-a formiranog nad fizičkim računarima bude pokrenuto više nezavisnih multibometrijskih sistema (slika 26).



Slika 26. Arhitektura radnog okruženja

U ovoj arhitekturi multibiometrijskog sistema potrebna su, minimalno, tri virtualna računara koje obezbeđuje IaaS:

- **Master NODE** : pokreće osnovne servise koji omogućuju konfiguraciju *Kubernetes* klastera. U njemu će funkcionisati sledeće komponente multibiometrijskog sistema: Sistem za upravljanje identitetima, Sistem za upravljanje skladištem, Sistem za upravljanje opterećenjem kontejnerima.
- **Infrastrukturni NODE** : Kroz servise na ovom virtualnom računaru se pokreću kontejneri koji će sadržati sledeće komponente: Sistem za upravljanje biometrijskom bazom podataka, Registar šablona biometrijskih aplikacija, Repozitorijum algoritama. Takođe će u njemu biti pristupna tačka preko koje će multibiometrijski sistem biti povezan ka svim ostalim mrežama (mreža putem koje se ostvaruje pristup računaru na kojem su povezani biometrijski senzori, Internet, itd).
- **NODE 1-N** : je skup virtualnih računara, a minimalno jedan, na kome će biti pokrenute komponente mutlibiometrijskog sistema: Kontrolor poslova,

Multibiometrijski sistem, Moduli sa algoritmima za: izvlačenje karakteristika, poređenje, fuziju.

7.3.1 Konfiguracija radog okruženja

Kreiranje Master noda je deo standardne instalacije i konfiguracije *Kubernetes*-a. Master je računar, fizički ili virtualni, koji sadrži komponente date u tabeli :

Komponente	Opis
API Server	*Kubernetes* API server radi validaciju i konfiguraciju podataka *pod*-ova, servisa i kontrolera replikacije. Takođe dodeljuje *pod*-ove nodovima i radi sinhornizaciju informacija o *pod*-u sa konfiguracionim servisom
etcd	etcd čuva perzistentne podatke o stanju Mastera dok ostale komponente posmatraju promene u etcd-u i sebe dovode u željeno stanje.
Controller Manager Server	*Controller manager server* posmatraju promene u kontroleru replikacije u etcd a tada koristi API da bi sever doveo željeno stanje.

Tabela 3. Tabela komponenti Master *Node*-a

Kubernetes karakteriše deskriptivna konfiguracija te je potrebno napraviti za Master *node* odgovarajući fajl.

```
apiLevels:
- v1beta3
- v1
apiVersion: v1
assetConfig:
  logoutURL: ""
  masterPublicURL: https://10.10.42.1:8443
  publicURL: https://10.10.42.1:8443/console/
  servingInfo:
    bindAddress: 0.0.0.0:8443
    certFile: master.server.crt
    clientCA: ""
    keyFile: master.server.key
    maxRequestsInFlight: 0
    requestTimeoutSeconds: 0
controllers: '*'
corsAllowedOrigins:
- 127.0.0.1
- 10.10.42.1:8443
- localhost
dnsConfig:
  bindAddress: 0.0.0.0:53
etcdClientInfo:
  ca: ca.crt
  certFile: master.etcd-client.crt
  keyFile: master.etcd-client.key
  urls:
  - https://10.10.42.1:4001
etcdConfig:
  address: 10.10.42.1:4001
  peerAddress: 10.10.42.1:7001
  peerServingInfo:
    bindAddress: 0.0.0.0:7001
    certFile: etcd.server.crt
```

```
    clientCA: ca.crt
    keyFile: etcd.server.key
  servingInfo:
    bindAddress: 0.0.0.0:4001
    certFile: etcd.server.crt
    clientCA: ca.crt
    keyFile: etcd.server.key
  storageDirectory: /var/lib/openshift/openshift.local.etcd
etcdStorageConfig:
  kubernetesStoragePrefix: kubernetes.io
  kubernetesStorageVersion: v1
  openShiftStoragePrefix: openshift.io
  openShiftStorageVersion: v1
imageConfig:
  format: openshift/origin-${component}:${version}
  latest: false
kind: MasterConfig
kubeletClientInfo:
  ca: ca.crt
  certFile: master.kubelet-client.crt
  keyFile: master.kubelet-client.key
  port: 10250
kubernetesMasterConfig:
  apiLevels:
  - v1beta3
  - v1
  apiServerArguments: null
  controllerArguments: null
  masterCount: 1
  masterIP: 10.10.42.1
  podEvictionTimeout: 5m
  schedulerConfigFile: ""
  servicesNodePortRange: 30000-32767
  servicesSubnet: 10.30.0.0/16
  staticNodeNames:
  - 127.0.0.1
masterClients:
  externalKubernetesKubeConfig: ""
  openshiftLoopbackKubeConfig: openshift-master.kubeconfig
masterPublicURL: https://10.10.42.1:8443
networkConfig:
  clusterNetworkCIDR: 10.1.0.0/16
  hostSubnetLength: 8
  networkPluginName: ""
oauthConfig:
  assetPublicURL: https://10.10.42.1:8443/console/
  grantConfig:
    method: auto
  identityProviders:
  - challenge: true
    login: true
    name: anypassword
    provider:
      apiVersion: v1
      kind: AllowAllPasswordIdentityProvider
  masterPublicURL: https://10.10.42.1:8443
  masterURL: https://10.10.42.1:8443
  sessionConfig:
    sessionMaxAgeSeconds: 300
    sessionName: ssn
    sessionSecretsFile: ""
  tokenConfig:
    accessTokenMaxAgeSeconds: 86400
    authorizeTokenMaxAgeSeconds: 300
policyConfig:
  bootstrapPolicyFile: policy.json
  openshiftInfrastructureNamespace: openshift-infra
  openshiftSharedResourcesNamespace: openshift
projectConfig:
  defaultNodeSelector: ""
  projectRequestMessage: ""
  projectRequestTemplate: ""
  securityAllocator:
    mcsAllocatorRange: s0:/2
    mcsLabelsPerProject: 5
    uidAllocatorRange: 1000000000-1999999999/10000
routingConfig:
```

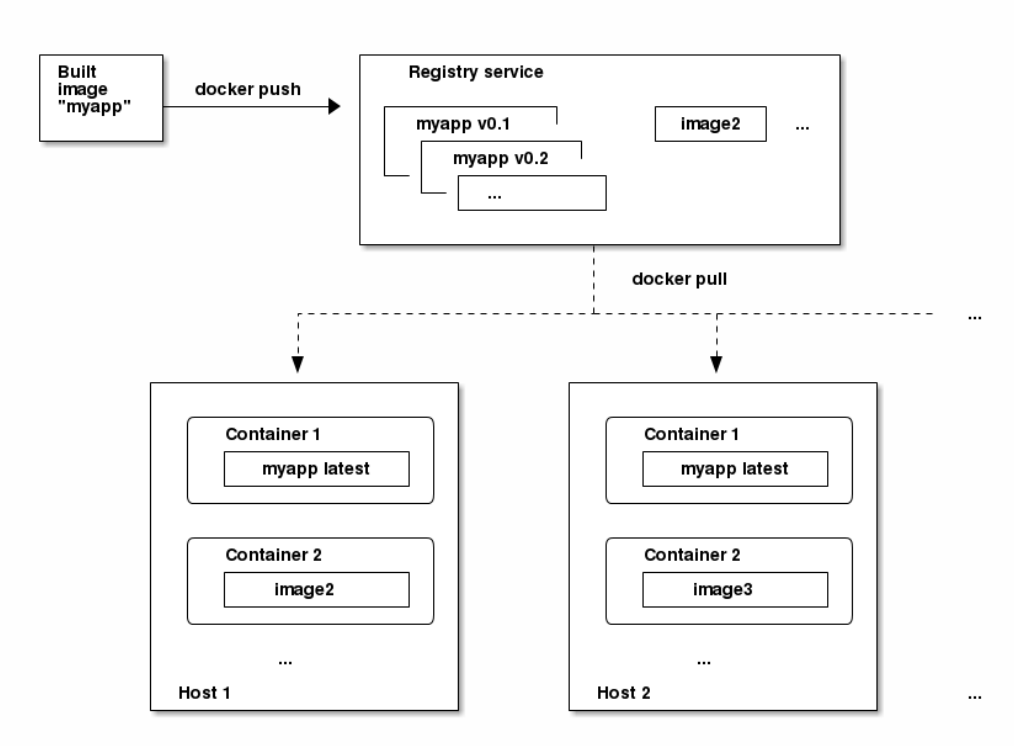
```
  subdomain: router.default.local
serviceAccountConfig:
  managedNames:
  - default
  - builder
  - deployer
  privateKeyFile: serviceaccounts.private.key
  publicKeyFiles:
  - serviceaccounts.public.key
servingInfo:
  bindAddress: 0.0.0.0:8443
  certFile: master.server.crt
  clientCA: ca.crt
  keyFile: master.server.key
  maxRequestsInFlight: 500
  requestTimeoutSeconds: 3600
oauthConfig:
  identityProviders:
  - name: "MBS_identity_provider"
    challenge: true
    login: true
    provider:
      apiVersion: v1
      kind: LDAPPasswordIdentityProvider
      attributes:
        id:
        - uid=admin,ou=Users,dc=mbs,dc=mmklab,dc=local
        email:
        - admin@mbs.mmklab.local
        name:
        - admin
        preferredUsername:
        - admin
      bindDN: "cn=admin"
      bindPassword: "Probni-pass"
      ca: mbs-ldap-ca-bundle.crt
      insecure: false
      url: "ldaps://ldap.mbs.mmklab.local/ou=Users,dc=mbs,dc=mmklab,dc=local?uid" (10)
```

Dok se inicijalna konfiguracija za svaki *node*, uključujući i Infrastrukturni *node*, kreira na osnovu sledeće osnovne konfiguracije:

```
allowDisabledDocker: false
apiVersion: v1
dnsDomain: multibio.cluster.local
dnsIP: 127.0.0.1
dockerConfig:
  execHandlerName: native
imageConfig:
  format: openshift/origin-${component}:${version}
  latest: false
kind: NodeConfig
masterKubeConfig: node.kubeconfig
networkPluginName: ""
nodeName: 127.0.0.1
podManifestConfig: null
servingInfo:
  bindAddress: 0.0.0.0:10250
  certFile: server.crt
  clientCA: node-client-ca.crt
  keyFile: server.key
volumeDirectory: /tmp/cfg/openshift.local.volumes
```

Registar šablona biometrijskih aplikacija je registar pripremljenih kontejnera. Sadrži ne samo kontejnere biometrijskih aplikacija već i sitemske kontejnere potrebne za regularan rad multibiometrijskog sistema.

Za kreiranje registra šablona biometrijskih aplikacija se može iskoristiti već postojeća infrastruktura kroz izgradnju privatnog registra. Funkcionalni model za igradnju privatnog registra šablona biometrijskih aplikacija je prikazan na slici 27.



Slika 27. Model izgradnje privatnog registra šablona biometrijskih aplikacija

Registar je jedan od sistemskih kontejnera a već je predefinisan na sledeći način:

```
FROM alpine:3.4

RUN set -ex \
    && apk add --no-cache ca-certificates apache2-utils

COPY ./registry/registry /bin/registry
COPY ./registry/regitry-config.yml /etc/docker/registry/config.yml

VOLUME ["/var/lib/registry"]
EXPOSE 5000

COPY docker-entrypoint.sh /entrypoint.sh
ENTRYPOINT ["/entrypoint.sh"]

CMD ["/etc/docker/registry/config.yml"]
```

Listing 1: Docker file definicija kontejnera za registar

```
version: 0.1
log:
  fields:
    service: registry
storage:
  cache:
    blobdescriptor: inmemory
  filesystem:
    rootdirectory: /var/lib/registry
```

```
http:
  addr: :5000
  headers:
    X-Content-Type-Options: [nosniff]
health:
  storagedriver:
    enabled: true
    interval: 10s
    threshold: 3
```

Listing 2: Konfiguracija Registry kontejnera

Skladište podataka u modelu radnog okruženja je definisano sa dva tipa skladišta: perzistentno i privremeno. Perzistentno skladište se koristi za sve podatke koji se koji se trajno čuvaju u multibiometrijskom sistemu: podaci biometrijske baze podataka, skladište biometrijskih uzoraka koji su uneti u biometrijsku bazu podataka, skladište podataka Repozitorijuma algoritama, skladište podataka u koje se smeštaju fajlovi implementacije algoritama, skladište u kojima se smeštaju šabloni kontejnera koji se koriste u multibiometrijskom sistemu. Privremeno skladište podataka se koristi za čuvanje podataka koji nastaju tokom životnog ciklusa kontejnera. Karakteristika oba tipa skladišta podataka je da prostor koji se koristi treaba da obezbedi IaaS *Cloud* provajdera. Koristeći mehanizam za dinamičko obezbeđivanje omogućujemo IaaS-u da optimalno raspoređuje podatake na skladištu koji pruža kao uslugu na zahtev. Takođe to omogućuje i manji broj skladišta koja će multibiometrijski sistem tražiti od IaaS provajdera. U skladu sa navedenim klaster multibiometrijskog sistema će koristiti dva bloka skladišta koja će dobiti od IaaS provajdera, a u internom korištenju će se označiti da jedan blok kao perzistentno skladište a drugi kao privremeno skladište sa dinamičkom alokacijom resursa u odnosu na zahteve koji će kontejneri u radu.

Ovakvim pristupom korišćenja skladišta podataka ono postaje merljiv resurs nad kojim se može pratiti stanje, odnosno postaviti kvote na količinu prostora koji se koristi od strane pojedinačne komponente sistema. Da bi se skladište uvelo kao meljiv resurs u multibiometrijski sistem potrebno je konfigurisati *StorageClass resource* objekat koji opisuje i klasifikuje skladište koje se može zahtevati i omogućiti deljenje istok skladišnog prostora među više različitih instanci jednog ili više kontejnera. Osnovna struktura *StorageClass* objekta je definisana na sledeći način:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: dynamic_storage
  annotations:
     ...
provisioner: kubernetes.io/plug-in-type
parameters:
  param1: value
  ...
```

```
  paramN: value
```

gde je ***provisioner*** atribut koji može imati jednu od sledećih vrednosti: NFS, iSCSI, Fibre Channel, GlusteFS, OpenStack Cinder, Ceph RBD, AWS Elastic Block Store, GCE Persistent Disk, VMWare vSphere, LocalVolume. Postavljanje atributa ***annotations*** na *storageclass.kubernetes.io/is-default-class: "true"* omogućuje da se konfigurisano skladište koristi za dinamičko obezbeđivanje blok skladišta na zahtev. Naravno moguće je koristiti više različitih provajdera skladišta istovremeno pri čemu je neophodno u konfiguraciji naglasiti koji provajder je osnovni.

Provajder identiteta je konfigurisan tokom instalacije Master *Node*-a a parametri se već nalaze u konfigracionim fajlu.

Sistem za praćenje opterećenja se oslanja na ugrađeni mehanizam kontrole procesora, memorije i skladišta u *Kubernetes*-u. Mehanizam koristi imenski prostor za upravljanje nad navedenim resursima. U predloženom modelu generičke arhitekture, radi regularnog rada multibiometrijskog sistema Master *node* treba da alocira onoliko procesorskih i memorijskih resursa koliko poseduje računar na kome se izvršava. Infrastrukturni *node* takođe obezbeđuje servise koji mogu da utiču na rad multibiometrijskog sistema te za njega važe ista pravila upotrebe resursa kao i za Master *node*. Kod *node*-ova na kojima će se izvršavati kontejneri biometrijskih algoritama može se ograničiti upotreba navedenih resursa.

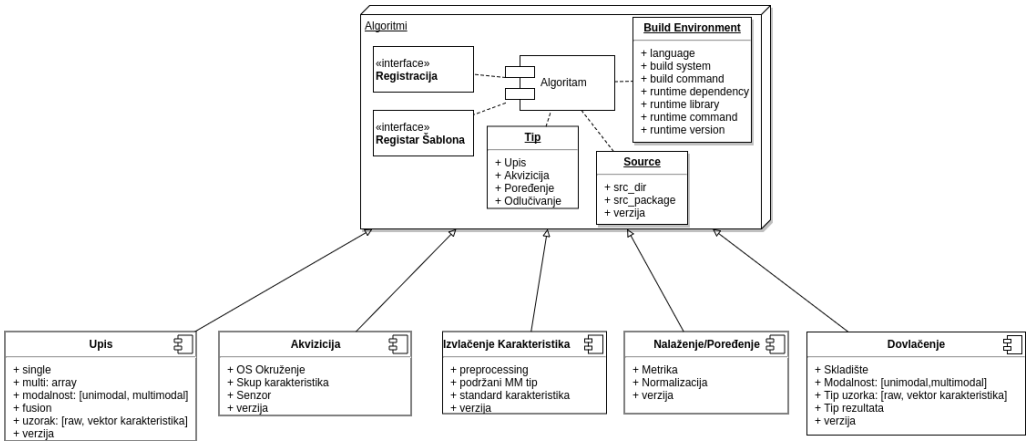
```
apiVersion: v1
kind: LimitRange
metadata:
  name: Algo-resources
spec:
  limits:
  - max:
      cpu: "1"
      memory: "500Mi"
    min:
      cpu: "500m"
      memory: "100Mi"
    type: Container
```

Da bi navedena ograničenja mogla da se primene potrebno je kreirati imenski prostor **Algo-resources**. Prilikom kreiranja *pod*-a sa algoritmom dodeliti ograničenja.

7.4 Repozitorijum algoritama

Repozitorijum algoritama je složena komponenta multibiometrijskog sistema. Osnovni zadatak je da čuva i održava bazu algoritama koji se mogu koristiti u kontejnerima biometrijskih aplikacija. Informacije o algoritmu koje se čuvaju u repozitorijumu su dovoljne da se na osnovu njih konstruiše šablon kontejnera koji

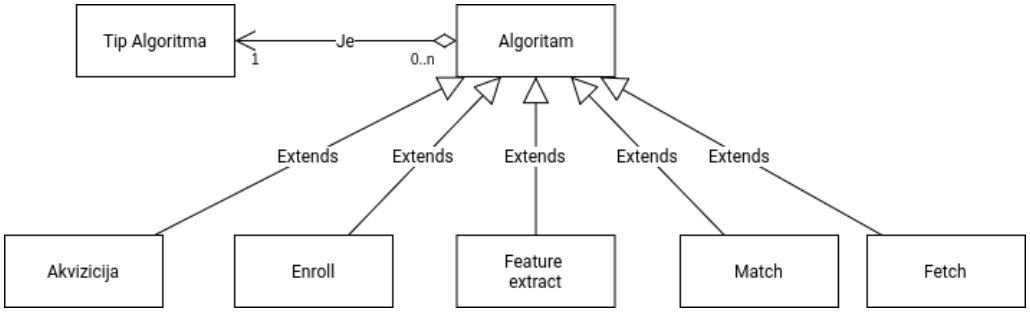
će biti sačuvan u registru šablona biometrijskih aplikacija. Model komponenti repozitorijuma algoritama je dan ta slici 28.



Slika 28. Model komponenti repozitorijuma algoritama

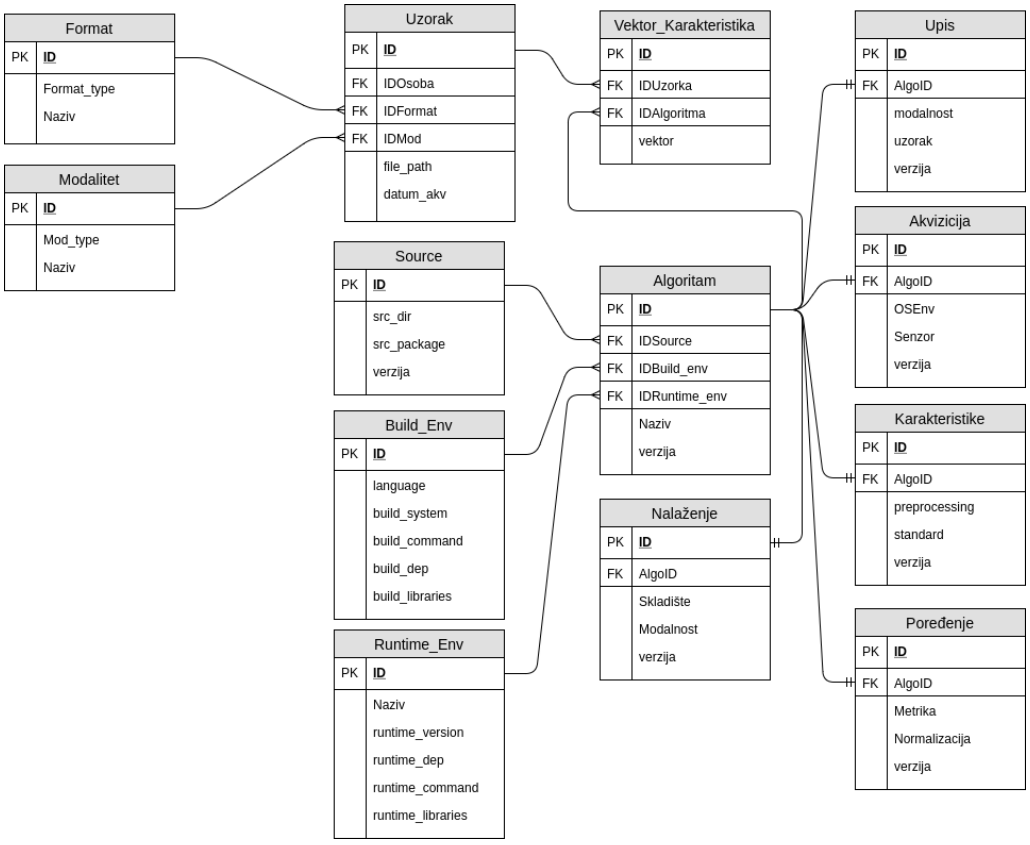
Jedan algoritam je predstavlja jedan ili više fajlova koji u sebi sadrže neki programski kod u nekom programskom jeziku i oni se nalaze u nekom direktorijumu. Ukoliko fajlovi implementacije zahtevaju kompjaliranje programskog koda, potrebno je da se algoritmu konfigurišu koji *build* sistem koristi i komandu sa kojom se pokreće kompajliranje. Takođe ukoliko se koriste eksterne biblioteke tokom kompjaliranja potrebno ih je navesti u konfiguraciji kroz parametar *build-dependency*. *Build* sistem pored pravljenja izvršne verzije alogritma može vratiti status kompajliranja, te na osnovu njega se može algoritam proglasiti aktivnim u slučaju da je kompajliranje uspešno, odnosno neaktivnim ukoliko kompjaliranje nije uspešno. Kod interpreterskih jezika proces kompajliranja nije obavezan, ali ukoliko postoji kompajler za dati interpreterski jezik potrebno ga je koristiti zbog provere leksičke i semantičke ispravnosti kao i zbog provere valjanosti biblioteka koju algoritam zahteva pri izvršavanju.

Na osnovu modela komponenti može se igraditi objektni model koji opisuje međuzavisnosti objekata u aplikaciji koja će održavati repozitorijum algoritama (slika 29).



Slika 29. Dijagram objekata repozitorijuma algoritama

Algoritmi za svaki pojedini tip su specijalzacija algoritma pa je omogućeno da svaki od algoritama poseduje svoje sebi svojstvene atribute. Na slici 30 je predstavljen model baze podataka koji podržava predloženi repozitorijum algoritama.



Slika 30. Model baze podataka repozitorijuma algoritama

Sada je moguće na osnovu podataka koji se nalaze u repozitorijumu algoritama generisati *Dockerfile* sa *runtime* okruženjem u kome će se algoritam izvršavati u kontejneru. Dobijeni fajl sa izvršnim kodom u slučaju kompajlerski orjetisanih jezika sa biliotekama koje koristi za izvršavanje su deo izvršnog okruženja koje

predstavlja osnovu za pripremu kontejnera u registru šablona biometrijskih aplikacija.

```
FROM centos:centos7
LABEL algoritam.tip="tip␣algoritma"
LABEL algoritam.id="id␣algoritma"
LABEL algoritam.version="verzija␣algoritma"

RUN yum -y update && yum clean all

ADD . /runtime

COPY ./runtime/algoritam-id /bin/algoritam-id

VOLUME ["/var/lib/algoritam/"]
EXPOSE 80

COPY algoritam-entrypoint.sh /entrypoint.sh
ENTRYPOINT ["/entrypoint.sh"]

CMD ["/bin/bash", "/runtime/bin/algoritam-start"]
```

Listing 3: *Dockerfile* definicija kontejnera za algoritam

U *Dockerfajl*-u su dodate labele koje jednoznačno određuju o kom algoritmu iz repozitorijuma se radi.

7.5 Infrastrukturni *Node*

Definicija pod-a za Infrastrukturni Node! kontejneri za postgres, registar šablona, repozitorijum algoritama!

Servisi koji se izvršavaju u ovom *node*-u postavljaju i održavaju infrastrukuru multibiometrijskog sistema i svaki od njih je kontejner za sebe. Za ove servise je potrebno obezbediti i persitentno skladište u kome će se nalazati podaci neophodni za regularan rad ovih servisa. Prikaz configuracije *pod*-a za infrastrukturni *node*:

```
apiVersion: v1
kind: Pod
metadata:
  name: infrastruktura
spec:
  volumes:
  - name: sgtInfrastruktura
    emptyDir: {}
  containers:
  - name: registar
    image: mbsregistry
    volumeMounts:
    - name: stg_mbsregistry
      mountPath: /var/lib/registry
  - name: mbs_baza
    image: "postgres:10.1"
    ports:
    - containerPort: 5432
      name: postgres
    volumeMounts:
      - name: stg_postgres
        mountPath: /var/lib/pg/data
  - name: repoAlgo
    image: "repoAlgo:0.1"
```

```
    ports:
    - containerPort: 80
      name: repoAlgo
    volumeMounts:
      - name: stg_repoAlgo
        mountPath: /var/lib/repoAlgo
```

7.6 *Node* sa biometrijskim aplikacijama

Na osnovu arhitekture biometrijskog sistema definisan je redosled akcija koje je potrebno izvršiti kao i redosled veza među modulima sistema (slika 2). Multibiometrijski sistem radi nad podacima koji se čuvaju u biometrijskoj bazi podataka na osnovu zadatih kriterijuma pretrage potrebno je napraviti listu poslova, odnosno listu uzoraka koju multibiometrijski sitem treba da obradi. Distribuirana obrada sa podelom poslova po pojedinačnim čvorovima klastera kontejnera zahteva upotrebu brokera poruka gde će se svaki biometrijski uzorak koji treba obraditi naći u jednoj poruci. Zbog toga je u multibiometrijski sistem uvršten i kontroler poslova pa je kontejner koji ga pokreće kao servis :

```
apiVersion: v1
kind: ReplicationController
metadata:
  labels:
    component: rabbitmq
  name: rabbitmq-controller
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: taskQueue
        component: rabbitmq
    spec:
      containers:
      - image: rabbitmq
        name: rabbitmq
        ports:
- containerPort: 5672
```

Multibiometrijski sistem će se sastojati od niza kontejnera sa izabranim algoritmom po svakom modulu njegova definicija će zavisiti od konkretnog slučaja korišćenja.

8 Korisnički zahtevi i model logičke arhitekture konkretnog obradnog multibiometrijskog sistema

U skladu sa operacijama koje se mogu zahtevati od multibiometrijskog sistema možemo definisati četiri različita slučaja korišćenja:

1. Akvizicija - upis u bazu podatka
2. Identifikacija
3. Verifikacija

8.1 Akvizicija - upis u bazu podataka

Tokom akvizicije se uzima biometrijski uzorak osobe. Tom prilikom osoba dolazi u interakciju sa senzorom. Multibiometrijski sistem se nalazi kod *Cloud* provajdera. Potrebno je uspostaviti vezu između senzora i akvizicionog modula mutlibiometrijskog sistema. Načini za uspostavu te veze mogu biti:

- Lokalna veza : kada je biometrijski senzor direktno priključen na računar. Tada je potrebno da taj računar poseduje neki mrežni adapter preko koga može da pristupi multibiometrijskom sistemu, preuzme kontejner sa akvizicionim modulom i odgovarajćum algoritmom za akviziciju. Uzorak koji uzme će obraditi kod sebe a potom isti poslati kroz mrežu do ostatka multibiometrijskog sistema. Takođe je moguće da taj računar bude tanki klijent na kome će se prikazivati interfejs virtualnog desktop računara, pri čemu će se biometrijski uređaj koji je priključen deliti sa virtualnim desktop računarom kroz komunikacioni kanal koji je već ostvario tanki klijent za prikaz interfejsa udaljenog virtualnog računara. Pri ovakvoj komunikaciji se ne prenosti akvizicioni modul do računara sa senzorom već se on izvršava na udaljenom računaru. Pri ovom tipu veze je potrebno da komunikacioni kanal između senzora i akvizicionog module bude pozdana i sigurna [108].
- Udaljena veza : kada biometrijski senzor ima ugrađen softver sa mogućnošću pružanja usluge web servisa preko koga se može dobiti biometrijski otisak. Akvizicioni modul dobija uzeti ostisak preko web servisa [109].

Uzeti uzorak se upisuje u biometrijsku bazu podataka i vezuje za odgovarajučeg korisnika u sistemu. Ukoliko je to prvi biometrijski uzorak za datog korisnika potrebno je kreirati korisnika sa parametrima o njegovom identitetu.

8.2 Verifikacija

U verifikaciji se za uzeti uzorak proverava da li on pripada korisniku sa zadatim ID-jem. Ukoliko je senzor multimodalni proverava se svaki uzeti modalitet. Uzeti uzorak ne mora da nastane u interakciji korisnika sa senzorom već se uzorak može pročitati iz biometrijske isprave koju nosi korisnik sa sobom. Pri tome, uređaj koji očitava uzorak je priključen na računar a tip veze između uređaja i računara je isti kao i kod akvizicije. Ova aktivnost neće generisati veće opterećenje multibiometrijsokg sistema jer se iz biometrijske baze podataka povlače samo uzorci koji su vezani za korisnika sa već određenim ID-jem.

8.3 Identifikacija

Identifikacija je aktivnost koja generiše najveće opterećenje multibiometrijskog sistema. U ovoj aktivnosti biometrijski uzorak je potrebno uporediti sa svim postojećim uzorcima tog modaliteta svih registrovanih korisnika u datom multibiometrijskom sistemu. Ukoliko u biometrijskoj bazi za izabrani algoritam modula za izvlačenje karakteristika postoje karakteristični vektori onda modul za poređenje upoređuje samo vektore karakteristika. U protivnom za svaki biometrijski uzorak u multibiometrijskom sistemu se izvlači karakteristčni vektor sa izabranim algoritmom, a dobijeni karakteristični vektori upisuju u biometrijsku bazu podataka za svaki uzorak koji odgovara modalitetu uzetog biometrijskog uzorka. Modul za poređenje sa izabranim algoritmom radi fuziju u zavisnosti od nivoa fuzije koji je prikazan na slici 6. Kako je cilj rada multibiometrijskog sistema da za što kraće vreme dođe do rezultata, to će tokom ove aktivnosti biti iskorišćeni svi raspoloživi računarski resursi multibiometrijskog sistema u datom trenutku.

8.4 Model logičke arhitekture

Multibiometrijski sistem (slika 26) obezbeđuje svoje resurse od IaaS *Cloud* provajdera. Modalitet rada multibiometrijskog sistema se može optimizovati po osnovu minimizacije utrošenih resursa ili maksimizacije računarske snage. Automatsko skaliranje virtualnih računara na nivou IaaS-a omogućuje da broj angažovanih virtualnih računara u jednom trenutku vremena može da zivisi od obradnog opterećenja *Kubernetes* klastera.

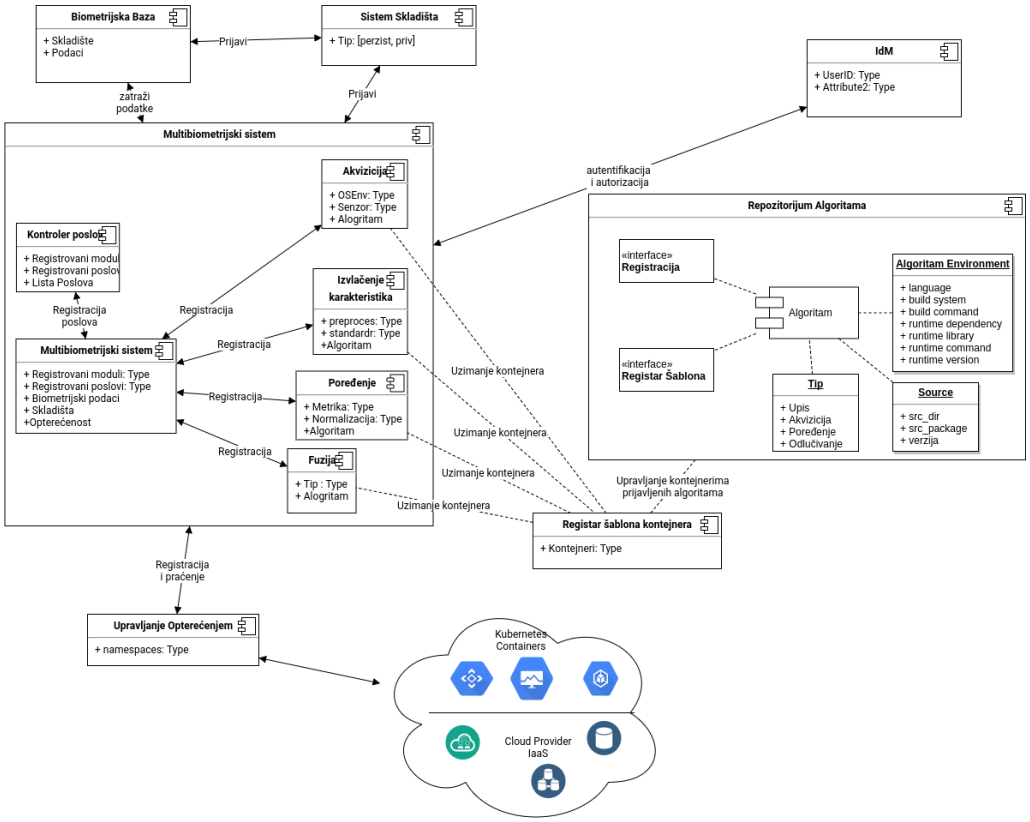
U strukturi logičke arhitekture multibiometrijskog sistema se izdvajaju dve celine koje predstavljaju obradni sistem (slika 31). To su :

- Multibiometrijski sistem : koji održava infrastrukturu neophodnu za obavljanje aktivnosti akvizicije, verifikacije i identifikacije. Upravlja kontejnerima biometrijskih algoritama i određuje proces rada u instanci multibiometrijskog sistema na osnovu specifičnih zahteva obrade uzorka. Registracijom algoritama se dobija specifičan redosled prolaska datog biometrijskog uzorka kroz sistem. Uz pomoć sistema za upravljanje opterećenjem može da dinamički alocira i dealocira resurse u *kubernetes* klasteru a i brojem virtualnih računara koji se dobijaju od *cloud* provajdera koji ne može da pređe zadatu granicu koja se postavlja administracijom IaaS-a. *Kubernetes* klaster takođe omogućuje upravljanje fizičkim računarima ukoliko se oni dodele klasteru. Time se može pored balansiranja opterećenja dobiti i mogućnost svojevrsnog bekap sistema u slučaju da *cloud* provajder se suoči sa nepredviđenim otkazima njegove infrastrukture.

- Repozitorijum algoritama : održava prijavljene algoritme na onsovu njihove namene. Ti algoritmi mogu da imaju svoje verzije što omogućuje istraživačima da isprobavaju i usavršavaju nove algoritme za bilo koji biometrijski modalitet. Na osnovu provere valjanosti algoritma, njegovim kompajliranjem moguće je kreirati kontejner koji će se čuvati u registru šablona kontejnera. Privatnost algoritama je obezbeđena kroz sistem upravljanja identitetima koji pomoću prava, praravila i uloga može da omogući tačno jednom korisniku autorizaciju nad alogritmom. Kontejnerska tehnologija omogućuje i minimalni prostor potreban za čuvanje različitih verzija jednog algoritma. Nova verzija kontejnera se pravi na osnovu prethodne tako što pamte samo promene između stare i nove verzije.

Biometrijska baza podataka predstavlja sponu između struktuiranih podataka o biometrijskim uzorcima kao što su na primer modalitet, multimedijalni format zapisa, vektor karakteristika za primenjeni algoritam, putanju u fajl sistemu koja pokazuje na uzorak itd sa multimedijalnim sadržajem uzorka. Ona takođe čuva podatke o algoritmima kao i putanje do fajlova čiji sadržaj određuje algoritam. Kroz sistem prava, pravila i uloga, koji su sastavni deo sistema za upravljane bazama podataka, može se ograničiti pristup svim uzorcima i algoritmima ili selektivno dozvoliti pristup. U sistemu je omogućeno da postoji više instanci multibiometrijskog sistema pa je tako moguće definisati do kojih podataka o uzorcima i algoritmima može svaka instanca da pristupi.

Skladište je otpravnik zahteva *kubernetes* klastera ka celokupnom skladištu koje je sistem dobio od strane IaaS-a. Dinamički alocira i dealocira prostor i obezbeđuje optimizaciju pristupa skladištu. Ukoliko se fizički računar doda *kubernetes* klasteru njegov lokalni disk može da služi samo kao privremeno skladište za sve kontejere koji se izvršavaju na tom računaru.



Slika 31. Model logičke arhitekture obradnog multibiometrijskog sistema

9 Model upravljanja performansama distribuiranog multibiometrijskog sistema

Glavni cilj modeliranja je donošenje odluke o kompletnoj konfiguraciji distribuiranog sistema i strategiji njegovog rada koja će ektrimizirati određene karakteristike pri zadatom skupu ograničenja. Tako se na primer minimiziraju ukupni investicioni troškovi, troškovi eksploatacije, vreme odziva, vreme pristupa a maksimiziraju se karakteristike kao što su brzina izvršavanja i pouzdanost sistema. Ograničenja u modelu mogu predstavljati količinu sredstava koja stoje na raspolaganju za projektovanje distribuiranog sistema, minimalnu ili maksimalnu vrednost nekog parametra sistema kao što je pouzdanost, vreme odziva ili slično.

Svaki model predstavlja uprošćenu sliku realnog sistema na kojoj se može eksperimentisati u cilju dobijanja željenih rezultata. Težnja svih modelara je da model dovedu do matematičke formulacije, odnosno do nivoa na kome je moguće dati njegov matematički opis. Time se postupak rešavanja problema svodi na rešavanje sistema jednačina i nejednačina kojima je opisan distribuirani sistem u cilju pronalaženja vrednosti upravljačkih promenljivih koje će dovesti sistem u optimalno stanje. Pri formiranju matemaičkog modela distribuiranog sistema u razmatranje treba uzeti sve njegove elemente i karakteristike koji mogu uticati na izbor optimalne konfiguracije i strukture sistema. Pri tome je potrebno naći načina za iskaivanje “dobrote” sistema, odnosno kriterijume preko kojima će se odlučiti kakve vrednosti upravljačkih promenljivih traba izabrati da bi sistem bio optimalan. Kriterijumi po kojima se bira optimalan sistem mogu biti različiti, kao što su investicioni troškovi, troškovi održavanja i eksploatacije, dostupnost podataka, vreme pristupa i vreme odziva, vreme izvršavanja, i vreme prenosa, protok mreže i slično. Prilikom modeliranja se u obzir ne uzimaju svi kriterijumi već se samo jedan ili nekoliko njih koji su od najveće važnosti modelaru.

Aplikacije multibiometrijskog sistema se izvršavaju isključivo na čvorovima. Ukoliko na čvoru ne postoji aplikacija ona se instalira na osnovu šablona u Registru. Pretpostavlja se da se prilikom startovanja čita cela aplikacija, odnosno da nema čitanja delova aplikacije. Ako su u toku rada aplikacije potrebni podaci oni se čitaju sa dodeljenog perzistentnog skladišta koji može biti lokalni disk ili mrežni disk prostor na nekom udaljenom skladištu podataka. Takođe treba uzeti u obzir da da je pre modeliranja uzet u obzir verovatnoće bezotkaznog rada komponenti računara u planiranom vremenskom interavalu na čijim osnovama će se izračunati

verovatnoće dostupnosti datoteka i aplikacija. Zadovoljenjem polaznih pretpostavki može se kreirati i konačni višekriterijumski linearni matematički model.

9.1 Notacija modela

N = skup čvorova

n = indeks čvora

P = skup procesora

D = skup diskova

M = skup memorije

G = skup grafičkih interfejsa

A = skup aplikacija

a = indeks aplikacije

F = skup datoteka

f = indeks datoteke

$PPRICE_n$ = cena procesora na čvoru n

$DPRICE_n$ = cena diska na čvoru n

$MPRICE_n$ = cena memorije na čvoru n

$GPRICE_n$ = cena grafičkog interfejsa na čvoru n

$PLIFE_n$ = vek trajanja procesora na čvoru n

$DLIFE_n$ = vek trajanja diska na čvoru n

$MLIFE_n$ = vek trajanja memorije na čvoru n

$GLIFE_n$ = vek trajanja grafičkog interfejsa na čvoru n

$PRELI_n$ = pouzdanost procesora na čvoru n

$DRELI_n$ = pouzdanost diska na čvoru n

$MRELI_n$ = pouzdanost memorije na čvoru n

$GRELI_n$ = pouzdanost grafičkog instefejsa na čvoru n

$NRELI_n$ = pouzdanost čvora n, verovatnoća da čvor neće biti neoperativan iz razloga koji nisu obuhvaćeni pozdanošću pomenutih resursa

$MIPS_n$ = brzina procesora na čvoru n

RAM_n = veličina RAM-a na čvoru n

$DISK_n$ = kapacitet diska na čvoru n

JOB_n = prosečan broj poslova koji se obave u toku određenog perioda (dnevno) na čvoru n

$ASIZE_a$ = veličina palikacije a

$ARAM_a$ = minimalna veličina RAM-a potrebna da bi aplikacija mogla da se izvršava

$AGRAPH_a$ = minimalan nivo grafičkog interfejsa koji je potreban da bi aplikacija a mogla da se izvršava

$AMIPS_a$ = minimalna brzina procesora potrebna da bi aplikacija a mogla da se izvršava

$ADISK_a$ = ukupna veličina aplikacije a na disku

$FSIZE_f$ = veličina datateke f (u KB)

ANA_{an} = binarna matrica alokacija aplikacija po čvorovima

FNA_{fn} = binarna matrica alokacija datoreka po čvorovima

ANP_{an} = matrica verovatnoće korišćenja aplikacija po čvorovima

FAU_{fn} = matrica prosečnog obima korišćenja datotetka po jednom startovanju aplikacije

TSPEED = brzina prenosa podataka u mreži

9.2 Kriterijumi u modelu

Investicioni troškovi (IC) prestavljaju zbir svih cena resursa alociranih u sistemu.

Investicioni troškovi po jedinici vremena (IC') prestavljaju troškove opreme izražene po određenom vremenskom periodu

Procenat redudanse datoteka (RP) izražava meru u kojoj su podaci replicirani.

Iskorišćenost kapaciteta diskova (DU) versno odslikava koliko je dobar izbor diskova.

Prosečna količina prenetih podataka (AT) po jednom korišćenju aplikacije u mreži pokazuje koliko se podataka prenese preko mreže prilikom jednog startovanja aplikacije.

Prosečno vreme transfera (ATT) je vreme u kojem se izvrši transfer prosečne količine podataka po jednom startovanju aplikacije.

Dostupnost datoteka (FA) izražava verovatnoću sa kojom će datoteka biti dostupna u odereñenom vrmenskom instervalu

Dostupnost aplikacije (AA) izražava verovatnoću sa kojom će aplikacija biti dostupna u određenom vremenskom instervalu.

$$IC = \sum_{n \epsilon N} (PPRICE_n + DPRICE_n + MPRICE_n + GPRICE_n) \tag{3}$$

$$IC' = \sum_{n \epsilon N} \left(\frac{PPRICE_n}{PLIFE_n} + \frac{DRPICE_n}{DLIFE_n} + \frac{MPRICE_n}{MLIFE_n} + \frac{GPRICE_n}{GLIFE_n} \right) \tag{4}$$

$$RP = \sum_{n \epsilon N} \left(\frac{\sum_{n \epsilon N} \sum_{f \epsilon F} \left(FNA_{fn} \cdot FSIZE_f \right)}{\sum_{f \epsilon F} FSIZE_f} - 1 \right) \cdot 100\% \tag{5}$$

$$DU = \frac{\sum_{n \epsilon N} \sum_{f \epsilon F} \left(FNA_{fn} \cdot FSIZE_f \right) + \sum_{n \epsilon N} \sum_{a \epsilon A} (ANA_{an} \cdot ASIZE_a)}{\sum_{n \epsilon N} DISK_n} \tag{6}$$

$$AT = \frac{\sum_{n \epsilon N} \sum_{a \epsilon A} \left(JOB_n \cdot ANP_{an} \cdot \left(ASIZE_a \cdot (1 - ANA_{an}) + \sum \left(FAU_{fa} \cdot \left(1 - FNA_{fn} \right) \right) \right) \right)}{\sum_{n \epsilon N} JOB_n} \tag{7}$$

$$ATT = \frac{AT}{TSPEED} \tag{8}$$

$$FA = \frac{1}{F} \cdot \sum_{f \epsilon F} \left(1 - \prod_{n \epsilon N} \left(1 - FNA_{fn} \cdot NRELI_n \cdot PRELI_n \cdot DRELI_n \cdot MRELI_n \cdot GRELI_n \right) \right) \tag{9}$$

$$AA = \frac{1}{A} \cdot \sum_{a \epsilon A} \left(1 - \prod_{n \epsilon N} (1 - ANA_{an} \cdot NRELI_n \cdot PRELI_n \cdot DRELI_n \cdot MRELI_n \cdot GRELI_n) \right) \tag{10}$$

9.3 Ograničenja u modelu

- Ograničenje kapaciteta diska

$$\sum_{f \epsilon F} FNA_{fn} \cdot FSIZE_f + \sum_{a \epsilon A} ANA_{an} \cdot ADISK_a \leq DISK_n \tag{11}$$

Ovo ograničenje važi u slučajevima korišćenja kada se sve potrebne datoteke, u slučaju multibiometrijskog sistema su to datoteke u kojima se nalaze biometrijski uzorci za obradu na čvoru n. Međutim, ako se koristi samo jedan čvor tokom rada mutlibiometrijskog sistema, kapacitet diska na čvoru nesme biti manji od kapaciteta diska na kome se čuvaju svi biometrijski uzorci svih korisnika sistema ukoliko se radi o identifikaciji. To može predstavljati nepremostiv problem za velike baze podataka. S toga je primerenije koristiti mrežni disk koristeći neku od thenologija virtualizacije skladišta.

- Sve datoteke moraju biti alocirane

$$\sum_{n \epsilon N} FNA_{fn} > 0 \tag{12}$$

Svaki čvor distribuiranog multibiometrijskog sistema, a u zavisnosti od modusa rada, verifikacija ili identifikacija, pre početka rada aplikacija potrebno je da ima dostupne biometrijske uzorke nad kojima će biti vršena obrada. Bez obzira da li se kao skladište koristi lokalni ili mrežni disk.

- Sve aplikacije moraju biti alocirane

$$\sum_{n \epsilon N} ANA_{an} > 0 \tag{13}$$

Svi moduli multibiometrijskog sistema trebaju biti raspoređeni minimalno na jednom čvoru. Maksimalan broj čvorova će zavisiti od opterećenja prilikom obrade multibiometrijskih uzoraka. Samo opterećenje pojedinačnog čvora u sistemu zavisi od primenjenog algoritma za otređeni biometrijski modalitet. Kako se obrada biometrijskih uzoraka svodi na izvršavanje više različitih unimodalnih aplikacija u kojima se mogu primeniti različiti algoritmi za obradu biometrijskog modaliteta opterećenje čvora direktno zavisi od kompleksnosti algoritma. Ovo ograničenje obezbeđuje da multibiometrijski sistem prilikom pokretanja ne može da dostigne radnu funkiconalnost bez pokretanja minimalnog skupa modula koji nesme da bude prazan.

- Grafički interfejs čvora odgovara potrebama aplikacije

$$GRAPH_n \geq AGRAPH_a \tag{14}$$

U zavisnosti od potreba algoritma moguće je alocirati procesor grafičkog adaptera sa ciljem povećanja računarske snage po jednom čvoru. Ovo ograničenje direktno utiče na izbor tipa računarskog sistema. Drugim rečima ukoliko aplikacija modula multibiometrijskog sistema zahteva prisustvo grafičkog interfejsa na čvoru ona se može izvršavati samo na fizčkom računarskom sistemu. U svakom mutlibiometrijskom sistemu postojaće barem jedan fizički računarski sistem koji će alocirati ovo ograničenje a to je sistem sa senzorima.

- RAM memorija čvora odgovara potrebama aplikacije

$$RAM_n \geq ARAM_a \tag{15}$$

Memorija dodeljena čvoru na kome će se aplikacija izvršavati se dinamički alocira ali ne može da pređe maksimalnu dozvoljenu veličinu određenu kvotom za taj čvor. Ovo ograničenje direktno utiče na dizajn repozitorijma alogritama jer pored osnovnih informacija o algoritmu zahteva i podatak o tome koliko memorije je potrebno obezbediti u radnom okruženju.

- Snaga procesora odgovara potrebama aplikacije

$$MIPS_n \geq AMIPS_a \tag{16}$$

Procesorska snaga dodeljena svakom čvoru multibiometrijskog sistema zavisi od konfiguracije tipa računarskog sistema na kome će se izvršavati aplikacija u čvoru. Ukoliko se radi o virtualnom računarskom sistemu procesorska snaga je određena kvotom koju zadaje administrator virtualnog okruženja. Za tip fizičkog računarskog sistema podešavanja procesorske snage takođe zadaje administrator tog sistema.

- Zbir verovatnoća korišćenja aplikacije mora biti jednaka 1 na svakom čvoru

$$\sum_{a \epsilon A} ANP_{an} = 1 \tag{17}$$

Ovo ograničenje definiše da svaki čvor mora biti sposoban da izvrši aplikaciju bilo kog modula multibiometrijskog sistema. Posledica ovog ograničenja je da

svakom čvoru mora biti dostupno skladište u kome će biti mutlibiometrijska baza podataka.

Pre svega, neki od kriterijuma se mogu ograničiti i time prevesti u ograničenja. Očigledan primer za to je kriterijum investicionih troškova. Vrlo je čest, pa čak i uobičajen slučaj, da su sredstva raspoloživa za projektovanje sistema organičena, pa je i time logično da ovaj kriterijum bude preveden u ograničenje. Isto tako se može postupiti i sa ostalim kriterijumima, ukoliko je potrebno.

Osim toga, prilikom projektovanja ovog tipa distribuiranog sistema uzećemo u razmatranje samo nekoliko mogućih rešenja. Pre svih, od interesa za posmatranje, se izdvajaju promenljive koje se odnose na procesorsku snagu, količinu memorije i veličinu skladišta podataka bez obzira da li se radi o lokalnom i mrežnom skladištu.

Rešavanje ovog sistema jednačina je vremenski dug proces da bi se njime dinamički upravljalo performansama sitema. Najveći deo opterećenja će upravo praviti kontejneri biometrijskih algoritama. Procesorska snaga i količina memorije koju će algoritam koristiti u toku rada direktno utiče od multimedijalnog tipa i sadržaja uzorka, načina implementacije, izbora programskog jezika, te je vreme izvršavanja i utrošak resursa nemerljivo pre izvršavanja.

Polazeći od funkcija cilja: minimizacije operativnih troškova sistema, minimizacije investicionih troškova, maksimizacije raspoloživosti podataka i minimizacije vremena odgovora sistema, u dizajnu ovakvog tipa distribuiranog sistema možemo konstatovati da se veličina sistema treba skalirati u zavisnosti od opterećenja pojedinačnog računarskog sistema koji ulazi u sastav distribuiranog multibiometrijskog sistema. Količina podataka, kako je već napomenuto, zavisi od načina rada. U verifikaciji se radi sa malim skupom podataka te opterećenje će biti manje. Minimalna veličina distribuiranog sistema se onda može dobiti rešavanjem predloženog modela sa navedenim funkcijama cilja. Prilikom indetifikacije, maksimalna veličina distribuiranog sistema, zavisi od količine biometrijskih uzoraka koje treba obraditi. Uz fiskni broj korisnika sistema rešenje ovog modela može dati maksimalnu veličinu obradnog distribuiranog multibiometrijskog sistema, pri čemu na minimizaciju vremena potrebnog za dobijanje odgovora od strane multibiometrijskog sistema direktno utiče minimizacija operativnih i investicionih troškova.

Skalabilnost koju nudi računarstvo u oblaku, daje mogućnost dinamičkog povećanja ili smanjenja broja čvorova u zavisnosti od opterećenja inicijanlno projektovanog sistema i navedenog obradnog sistema. Rešavanje navedenog modela optimizacije

performansi je mnogo duže u odnosu na vreme potrebno distribuiranom sistemu da donese odluku u realnom vremenu. Rezultat odlučivanja u upravljanju u dinamilkom distribuiranom sistemu treba da kaže Master modulu za koliko čvorova treba povećati ili smanjiti broj obradnih čvorova.

9.4 Heuristika

Skalabilnosti predloženog distribuiranog multibiometrijskog ekosistema u realnom vremenu se može postići uvođenjem monitora. Monitoring ovako složenog distribuiranog sistema je ključni korak u upravljanju performanasama. Monitor je alat koji koristi za posmatranje aktivnosti na sistemu. U opštem slučaju, monitori posmatraju performanse sistema, sakupljaju statistiku performansi, analiziraju podatke i prikazuju rezultat. Takođe mogu da identifikuju oblast gde nastaju problemi i da predlože način rešavanja problema.

Monitore ne koriste samo analitičari performanse sistema već takođe ih mogu koristiti programeri i administratori sistema. Sledeći razlozi mogu biti dobar razlog za nadgledanje sistema [110]:

- Sistemski programeri mogu koristiti monitor za nalaženje često korišćenih segmenata softvera koji se izvršava i da na osnovu analize prikupljeni podataka urade optimizaciju njihovih performansi
- Administratori sistema mogu koristiti monitor za merenje iskorišćenosti resursa i nađu uska grla koja utiču na performansu sistema.
- Administratori sistema takođe mogu koristiti monitor za rekonfiguraciju sistema. Parametri sistema se mogu podesiti sa ciljem poboljšanja performansi.
- Sistem analitičari mogu koristiti monitore za nalaženje parametara modela, validaciju modela i kreiraju uzlaze modela. Na osnovu toga mogu unaprediti model u cilju dobijanja bolji performansi sistema.

Monitori se mogu klasifikovati na osnovu različitih karakteristika, kao što su nivoi implenmentacije, načina pokretanja, i mogućnosti prikaza rezultata. Na osnovu nivoa implementacije mogu se podeliti na softverske monitore, hardverske monitore ili hibridne monitore. Na osnovu mehanizma pokretanja, monitori se mogu podeliti na monitore zasnovane na događajima ili na vremenskom intervalu. Monitori zasnovani na događajima se aktiviraju sa svakom pojavom tog događaja na sistemu. Monitori zasnovani na vremenskom intervalu se aktiviraju u fiksnim

vremenskim intervalima na osnovu prekida prouzrokovanih sistemskim časovnikom. Na osnovu mogućnosti prikaza rezultata mogu se podeliti na monitore u realnom vremenu i *batch* monitore.

Većina računarskih sistema su delovi distriburanih sistema i sastoje od nekog broja hardverskih i softverskih komponenti koje rade konkurentno. Praćenje distribuiranog sistema je teže nego praćenje centralizovanog sistema. Posmatrano sa strane monitora on je taj koji mora biti distribuiran kroz sistem a time će se sastojati od više komponenti koje moraju da rade konkurentno.

10 Određivanje konkretne fizičke arhitekture distribuiranog multibiometrijskog sistema

Pri određivanju konkretne fizičke arhitekture nad kojom će se izvršavati multibiometrijski sistem prvo je potrebno sagledati dimenzije podataka nad kojim će on raditi. Pod dimenzijama podataka se posmatraju:

- ukupan broj direktorijuma
- ukupan broj fajlova
- prosečna veličina sadržaja fajlova
- dubina stabla direktorijuma u kome su smešteni fajlovi

Navedeni parametri direktno utiču na performansu sistema jer je potrebno obezbediti što je moguće brži podsistem diskova na računarima. Naravno pod uslovom da ukupna količina zauzetog prostora na disku omogućuje da se za svaki kontejner odvoji dovoljno veliki privremeni prostor na skladištu u kome će biti smešteni svi uzorci. Ovakav pristup alokaciji resursa može da dovede do povećanja vremena kreiranja odnosno startovanja pojedinačnog kontejnera kome treba da budu dostuopni biometrijski uzorci. Ne predstavja problem moguća situacija da se na jednom računararu startuje više kontejnera algoritma za izvlačenje karakteristika jer se skladištni prostor može deliti između kontejnera. Privremeni skladišni prostor dobijen od strane sistema za svaki kontejner će se mapirati na perzistentni skladištni prostor koji je dati računar dobio od sistema. Biće mapirani onoliko puta koliko je to potrebno, odnosno do granice kada svi startovani kontejneri na tom računaru ne utroše ostale sistemske resurse, što se pre svega odnosi na količinu upotrebljene memorije račnunara i opterećenja procesora.

Prostor za bolju performansu, odnosno smanjeni utrošak sistemskih resursa se može potražiti u načinu na koji algoritmi pristupaju uzorcima. Osim algoritama koji rade akviziciju i upis u biometrijsku bazu podataka, algoritmi svih ostalih kontejnera samo čitaju fajlove. Ovo pruža mogućnost da mapiranje bude u *ReadOnly* modu što će rezultirati manjom potrošnjom za memorijom računara jer neće biti odvajani baferi za smeštanje novih podataka.

Sledeći aspekt koji treba posmatrati je utrošak memorije po pojedinačnom kontejneru. Utrošak memeorije zavisi od veličine multimedijalno sadržaja koji raspakovan zauzme u memoriji. Takođe način na koji je implementiran algoritam, odnosno da li strukture podataka koje se koriste u algoritmu su implementirane na

optimalan način. S obzirom da se ovi podaci mogu saznati samo aktivnim praćenjem utroška memorijskog resursa tokom rada i beležiti u nekom od log fajlova ili smeštati u zasebnu bazu podataka te nakon toga izvlačiti statistiku o utrošku memorije. Tako dobijeni rezultati se mogu upotrebiti u definiciji posmatranog kontejnera i redefinisati maksimalnu i minimalnu količinu memorije koju taj algoritam koristi u toku rada.

Iskorišćenost procesorske snage je veoma važan parametar u sistemu. Čak i ako posmatramo kroz cenu komponenti računara procesorska snaga često prevazilazi 50% ukupne cene. Brzina, efikasnost i efektivnost procesora utiče na brzinu izvršavanja svake operacije u i na sistemu. Osnovni cilj je da iskorištenost procesora bude što bliža maksimumu kapaciteta tog tipa resursa. Distribuiranost rešenje sama po sebi ne garantuje da će svi procesori u svim računarima uvek raditi na maksimumu snage. Često nije optimalno rešenje u igradnji klastera koji treba da opsužuje neku *Cloud* infrastrukturu da svi procesori budu iste snage i kapaciteta. Snaga procesora se ogleda u brzini njegovor rada, dok kapacitet predstavlja mogućnost paralelizacije izvršavanja instrukcija, odnosno broj jezgara koji poseduje. Procesori koji opslužuju sistem skladišta ne moraja da budu iste snage i kapaciteta kao i procesori nad kojima će se izvršavati virtualni računari i kontejneri. Međutim iskorišćenost procesora zavisi od programskog jezika u kome je implementiran algoritam, od implementacije algoritma, načina iskorišćenja ostalih resursa u sistemu, načina obrade multimedijalnog sadržaja uzoraka. Još uvek ne postoji metodologija i tehnologija koja može da unapred, pre izvršavanja da prikaže koji i kakav procesor je adekvatan za izvršavanje nekog izvršnog koda. Kao i kod utroška memorijskih resursa, može se pratiti i beležiti iskorišćenost procesorskih resursa u distribuiranom okruženju tokom rada. Na osnovu statističke obrade tih podataka mogu se može se planirati raspodela tih resursa pri pokretanju novih poslova. Koja može biti sa kriterijumom većeg angažovanja procesorske snage ili kriterijumom ravnoravnije preraspodele opterećenja procesora kroz klaster koji čini *Cloud* implementaciju.

Pri određivanju fizičke arhitekture, s toga, postoji mogućnost da inicijanlno rešenje bude predimenzionirano ili podimenzionirano. Model upravljanja perofrmansama omogućuje da inicijanlna fizička arhitektura bude što je moguće optimalnija za planiranu obradu. Heuristike će nam omogućiti da se obrada rasporedi po nekom od kriterijuma, recimo maksimalnog iskorišćenja resursa ili kriterijuma koji će omogućiti da se ravnomernije rasporedi obrada po računarima sistema. Skalabilnost *Cloud* tehnologija, vertikalana i horizontalna omogućuju da se nadopunjuju ili smanjuju resursi celokupnog sistema prema potrebama obrade koje

se mogu saznati na osnovu statistike prikupljene tokom rada sistema i na taj način otklanjati uska grla. Na osnovu logičkog modela arhitekture opisanog u osmom poglavlju a prikazanog na slici 31, moguće je dati minimalnu fizičku arhitekturu na kojoj će moći da se regularno izvršava multibiometrijski sistem.

Za minimalnu fizičku arhitekturu potrebno je obezbediti tri virtualna računara u kojima će moći da se instalira *Kubernetes* klaster. U tom distribuiranom okruženju minimalno jedan virtualni računar će biti slobodan za pokretanje kontejnera algoritama. Daljom konfiguracijom ovog okruženja sa kriterijumom maksimalne iskorišćenosti kapaciteta biće angažovani svi preostali resursi ovog distribuiranog okruženja. Od strane IaaS provajdera skladište podataka će biti objedinjeno i na osnovu zahteva će *Kubernetes* klaster da preraspodeljuje slobodan prostor pokrenutim kontejnerima. Ovo minimalno okruženje ne obezbeđuje kriterijum najbržeg dostizanja cilja, odnosno da se za najkraće vreme identifikuje identitet vlasnika biometrijskog uzorka. Kriterijum najbrže identifikacije zahteva od IaaS provajdera da mogućnost da *Kubernetes* klaster zatraži nove virtualne računare koje će priključiti klasteru na njegov zahtev koji će proizaći na osnovu monitoringa opterećenja biometrijskog sistema. Takođe ukoliko je potrebno, zbog veličine skladišnog protstora da u *Kubernetes* klaster se dodata eksterno, u odnosu na IaaS, skladište sa biometrijskim uzorcima koje obezbeđuje dovoljno smeštajnog kapaciteta.

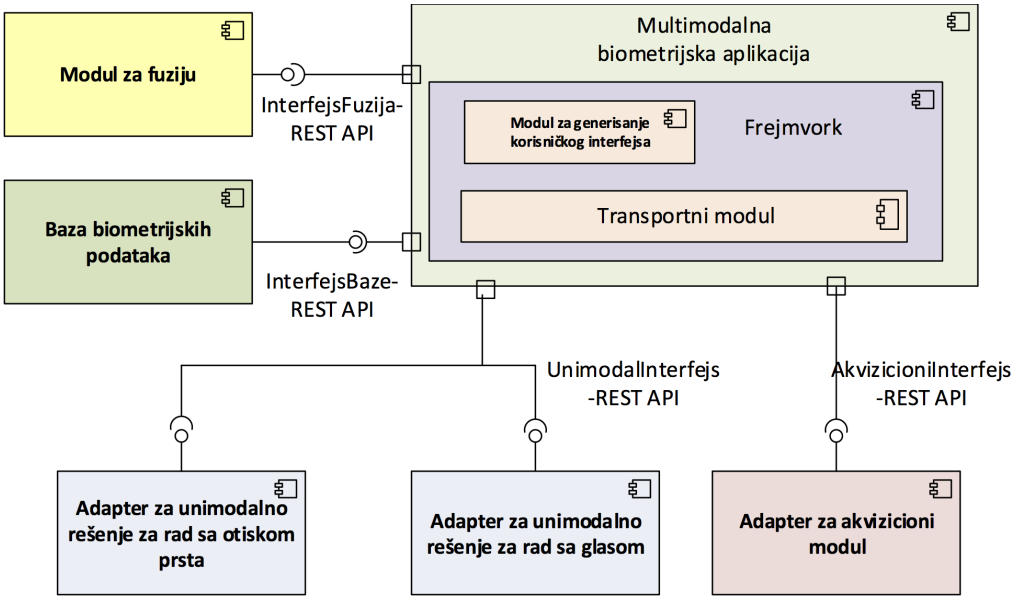
Logička arhitektura obradnog multibiometrijskog sistema deli fizičku arhitekturu na dva nivoa :

- Nivo virtualnih računara i skladišta
- Nivo IaaS-a

Ovom nivojskom podelom je moguća horizontalna i vertikalna skalbilnost svakog pojedinačnog nivoa. Takođe omogućuje da se model upravljanja performansama primeni na svakom nivou nezavisno jer se odgovarajućim heuristikama može pratiti opterećenje sistema na svakom nivou ponaosob.

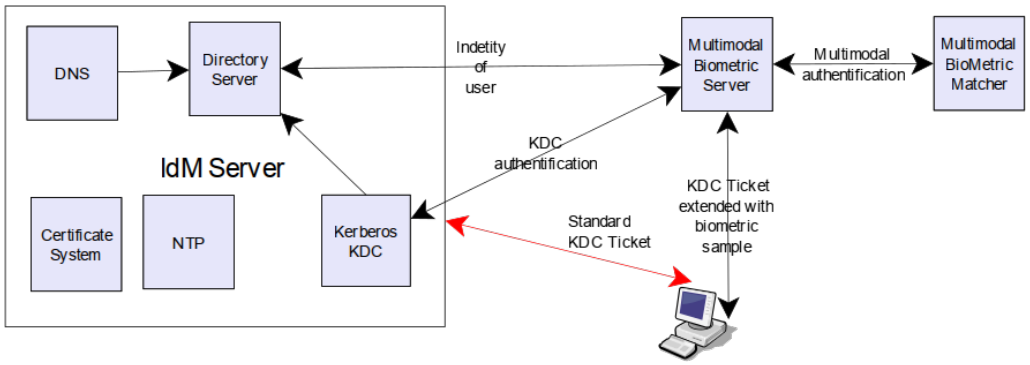
11 Studija slučaja

Verifikacija predloženog rešenja će biti primenjena na Multimodalnu aplikaciju, MMBIO, nastalu na projektu TR-32013. Aplikacija je napravljena sa arhitekturom (slika 32) koja za komunikaciju između modula koristi REST pristup.



Slika 32. Arhitektura multimodalne biometrijske aplikacije realizovane nad MMBIO *framework*-om [4]

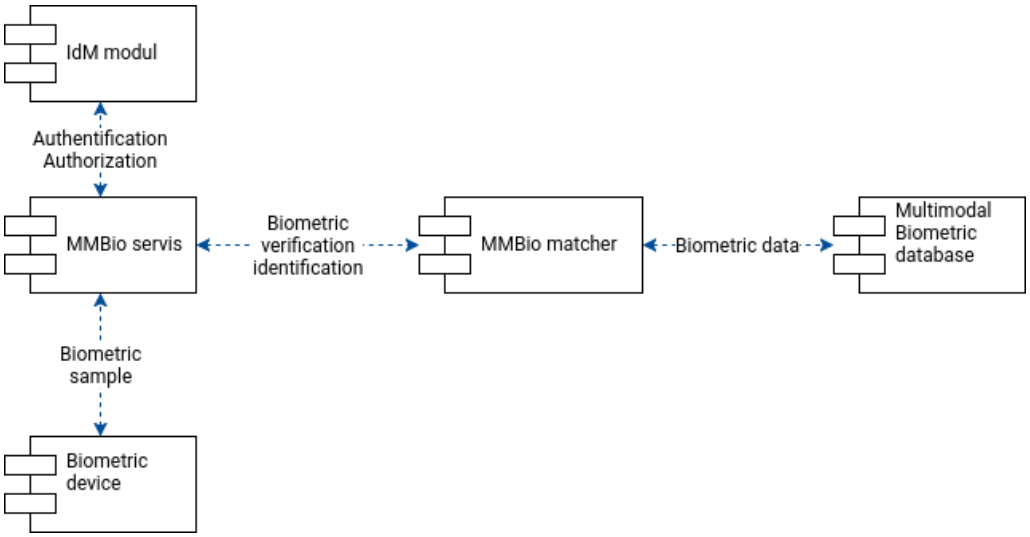
U prvom koraku je arhitektura proširena tako da podržava rad sa identitetima za koje su vezani biometrijski uzorci. Tako je dobijena arhitektura koja je predstavljena u radu [111] (slika 33).



Slika 33. MMBio server arhitektura

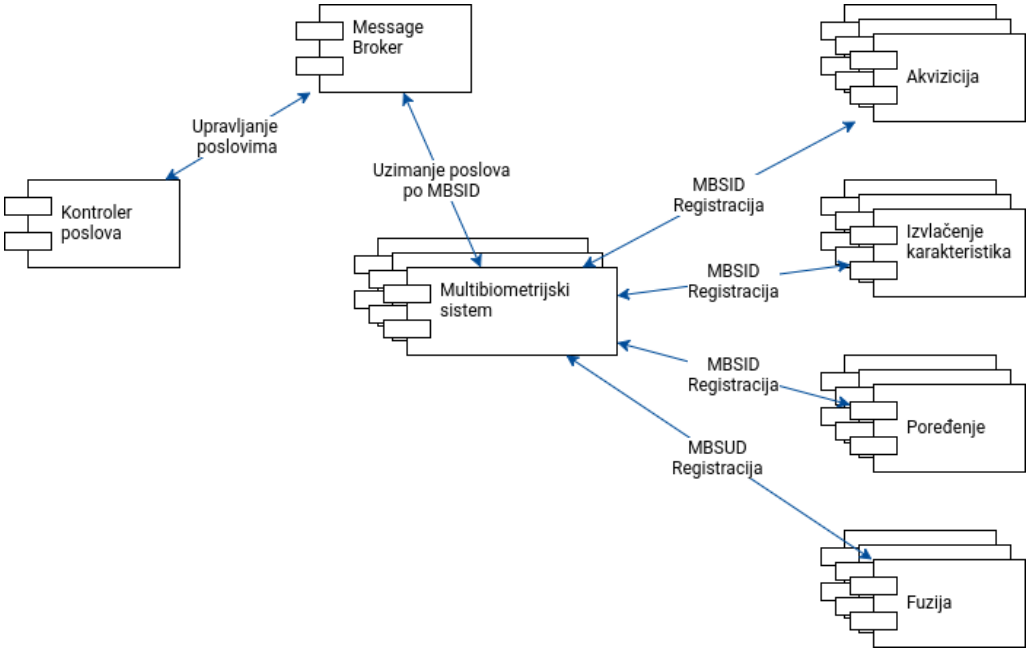
U radu [112] je predstavljen prošireni model transformisane arhitekture multibiometrijskog sistema MMBio (slika 34). Zadržan je isti komunikacioni

protokol u komunikaciji između modula. Proširenje je izvedeno u MMBio servisu tako da povezuje identitet sa biometriskim uzorkom u slučaju unimodalnog senzora ili biometrijskim uzorcima u slučaju multimodalnog senzora. MMBio servis od pristiglih zahteva generiše listu poslova a potom šalje serijalozovane zahteve ka MMBio Matcher-u. MMBio Matcher na osnovu modaliteta biometrijskog uzorka u šalje zahteve za podacima ka multimodalnoj biometrijskoj bazi.



Slika 34. Prošireni model MMBio arhitekture

Kako MMBio servis radi serijalizaciju poslova u jednom trenutku vremena moguće je obraditi samo jedan uzorak. Sledeći korak u transformaciji MMBio aplikacije je da se generisana lista poslova MMBio servisa preda kontroleru poslova koji će distribuirati obradu. U MMBio arhitekturi je dodat kontroler poslova koji prihvata zahteve od MMBio servisa pravi nezavisne liste liste poslova i šalje ih Message Brokeru koji upravlja listama poslova. Lista poslova dobija svoj jedinstveni identifikator. Svaki multibiometrijski sistem sa svojim jedinstvenim identifikatorom (MBSID) konkuriše za prvu slobodnu listu poslova. Svaka instanca multibiometrijskog sistema registruje procesne module na osnovu zahteva iz liste poslova (slika 35).

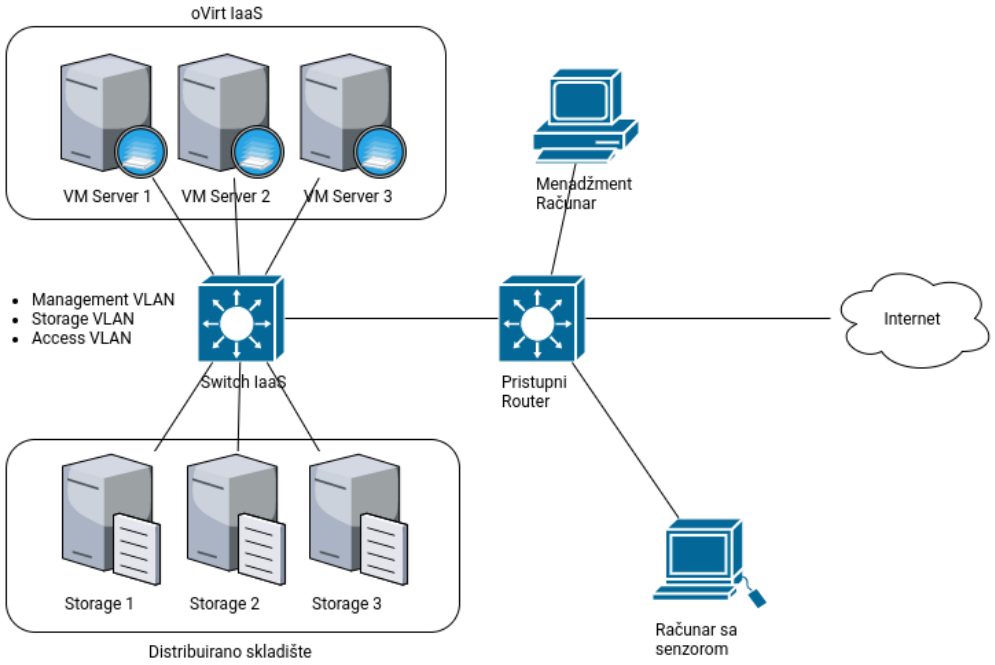


Slika 35. Distribuirana MMBio arhitektura

Kako je MMBio aplikacija podeljena transformacijom modula na web servise. Cilj transformacije je bio da se moduli aplikacije prilagode konfiguraciji kontejnera i izvršavanju u *Kubernetes* okruženju. Iz unimodalnih adpatera su izvučeni algoritmi tako da poštuju specifikaciju dodavanja u repozitorijum kao što je dato na slici 28.

Pri određivanju fizičke arhitekture distribuiranog multibiometrijskog sistema pošlo se od sledećih činjenica:

- Broj biometrijskih uzoraka: Prikupljeno je oko 20000 otisaka prstiju od 500 ljudi iz CASIA Fingerprint baze u verziji 5. Takođe su priključeni uzorci koji su prikupljeni na projektu TR-32013 ”Multimodalna biometrija u upravljanju identitetima” oko 1000 različitih uzoraka od oko 100 ljudi.
- Iskorišteno je oVirt okruženje koje je predstavljalo IaaS. Fizička arhitektura je data na slici 36.
- Primenom modela za upravljanje performansama sa datim brojem otisaka određena je minimalan broj virtualnih računara koji se koriste za *Kubernetes* klaster



Slika 36. Fizička aritektura IaaS-a

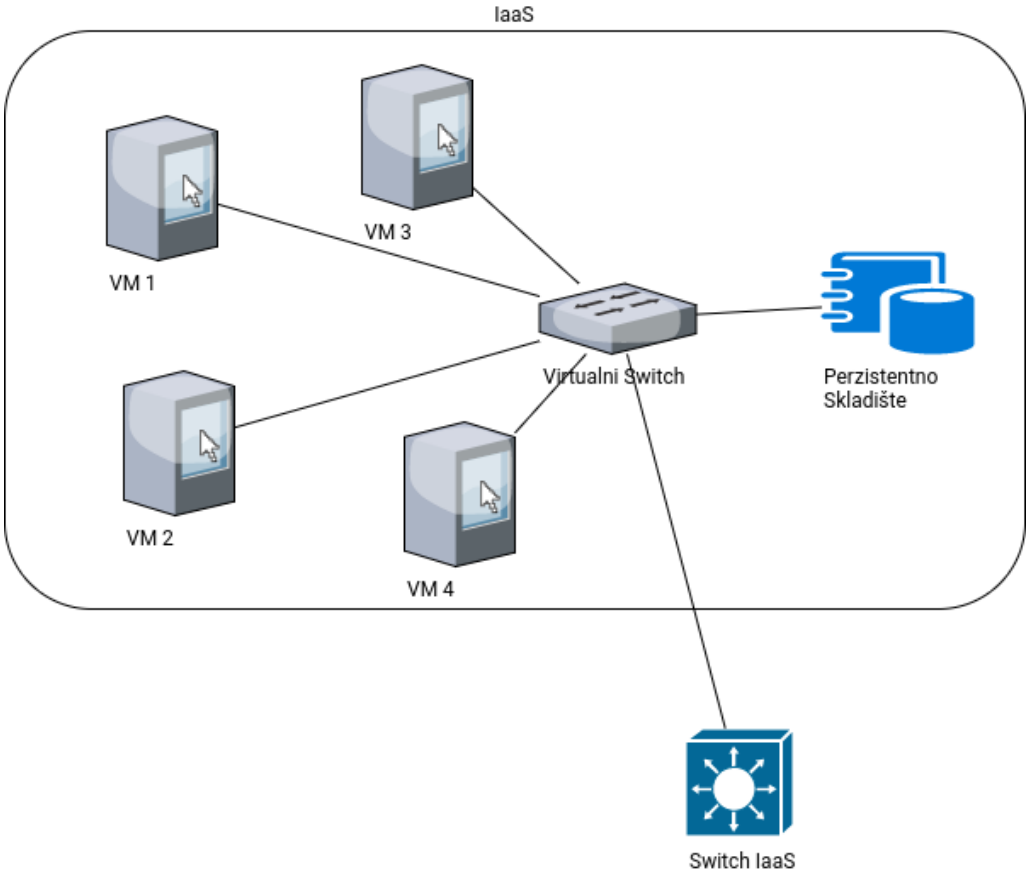
Na slici 36 je predstavljena fizička arhitektura distribuiranog multibiometrijskog sistema. U konkretnom fizičkom okruženju je korišćen oVirt virtualizaciona platforma kao IaaS *Cloud* provajder. Sastoji se od tri računara koji obezbeđuju distribuirano skladište podataka, i tri računara koji obezbeđuju virtualizaciju. Povezani su preko sviča koji podržava virtualizaciju mreže. U mreži su odvojene tri celine:

- Management VLAN: Obezbeđuje rad klastera za virtualizaciju i klastera skladišta. Kroz taj VLAN se pristupa softveru za kontrolu IaaS-a.
- Storage VLAN: Omogućuje sinhronizaciju sadržaja diskova na serverima za distribuirano skladište koji rade u režimu replikacije sadržaja. Kroz isti VLAN serveri za virtualizaciju dobijaju deljeno skladište kako bi obezbedili mogućnost migracije virtualnih računara bez prekida rada
- Access VLAN: povezuje IaaS sa ostatkom mreža u okruženju a preko pristupnog rutera i sa Internetom.

Menadžent računar preko veb aplikacije upravlja IaaS okruženjem, kreira i upravlja virtualnim računarima i deljivim skladištem. Takođe može da prati opterećenje sistema, te da raspoređuje ručno virtualne računare kroz servera za virtualizaciju. Administrator sistema može da konfiguriše i automatsku preraspoređivanje opterećenja na osnovu postvaljenih kvota utrošaka resursa na računarima za virtualizaciju.

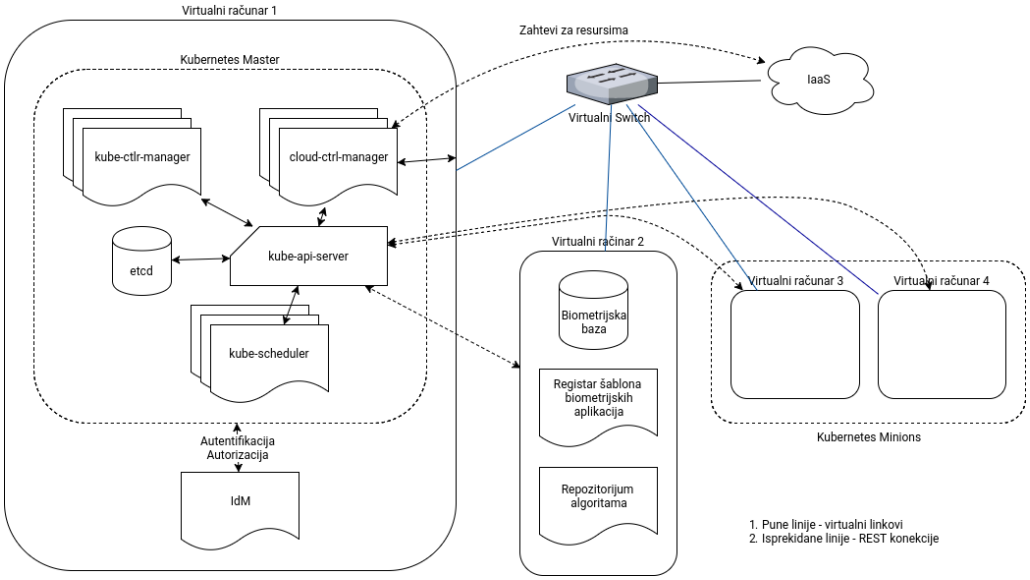
Računaru sa senzorom je omogućen pristup do virtualnih računara preko pristupnog rutera. Pristupni ruter pored rutiranja mrežnog saobraćaja ima ulogu i *firewall*-a sakrivajući mrežnu infrastrukturu IaaS od nedozvoljenog pristupa.

Na slici 37 su prikazane veze među virtualnim računarima i perzistetnim skladištem obezbeđenim od strane IaaS-a. Virtualni računari su raspoređeni od strane IaaS menadžmenta na jednom ili više fizičkih računara u zavisnosti od radnog opterećenja servera. Svako premeštanje virtualnog računara premešta i uspostavljene vitualne linkove ka virtualnom *switch*-u preko koga se uspostavljaju virtualni komunikacioni kanali. Servis virtualne računarske mreže u IaaS-u obezbeđuje potpunu izolaciju saobraćaja kroz fizičku mrežnu infrastrukturu. Skladišni prostor koji koriste virtualni računari je servis IaaS-a koji mapira zahteve virtualnog skladišta ka fizičkom. Kako ukupna količna skladišnog prostora za biometrijske uzorke iznosi oko 3GB nije bilo potrebe za angažovanjem dodatnih eksternih resursa skladištnog prostora.



Slika 37. Fizička aritektura *Kubernetes* virtualnih računara

Instalacijom i konfiguracijom *Kubernetes* servisa na jedan virtualni računar, obradno opterećenje je dostiglo limite memorijskih resursa dodeljenih tom računaru.



Slika 38. Logička aritektura *Kubernetes* virtualnih računara i servisa

Evaluacija rešenja je rađena za tri različita okruženja :

- Native : Na fizičkom računaru je pokretan MMBio aplikacija u izvornom obliku
- Kontejner : MMBio aplikacija transormisana u *Kubernetes* okružnju
- Virtualni računar : MMBio aplikacija pokrenuta u virtualnom računaru

Računarski resursi dodeljeni pojedinačnom virutalnom računaru su bili sledeći :

- CPU : 4 vCPU
- RAM : 8 GB
- DISK : 10 GB

Veličina perzistentnog skladišta obezbeđenog od strane IaaS-a je 500GB.

Repozitorijum algoritama je zauzeo 200 MB perzistentnog prostora na skladištu, a prilikom kompjaliranja algoritama privremeno skladište je zauzimalo i do 300 MB.

Registar šablona biometrijskih kontejnera je zauzeo 6 GB prostora gde je svaki šablon kontejnera zauzeo 1GB.

Biometrijska baza podataka je dobila perzitetno skladipte od 1GB.

Tokom rada transformisana MMBio aplikacija je radila samo u režimu rada identifikacije. Definisana su dva skupa resursa koji su dodeljivani kontejnerima. Razlika između ova dva skupa je samo u broju procesora koje može da dobije

kontejner. U prvom slučaju maksimalno 1 procesor a u drugom slučaju maksimalno 2.

```
apiVersion: v1
kind: LimitRange
metadata:
  name: Algo-resources-1vcpu
spec:
  limits:
  - max:
      cpu: "1"
      memory: "500Mi"
    min:
      cpu: "500m"
      memory: "100Mi"
    type: Container
```

Listing 4: Maksimalno 1 procesor

```
apiVersion: v1
kind: LimitRange
metadata:
  name: Algo-resources-2vcpu
spec:
  limits:
  - max:
      cpu: "2"
      memory: "500Mi"
    min:
      cpu: "500m"
      memory: "100Mi"
    type: Container
```

Listing 5: Maksimalno 2 procesora

Prilikom evaluacije rada multibiometrijskog sistema na fizičkom računaru procesu u kome se izvršavao multibiometrijski sistem primenjena je kontrola korišćenih računarskih resursa kroz isti mehanizam korišćenja kontrolnih grupa i imenskih prostora.

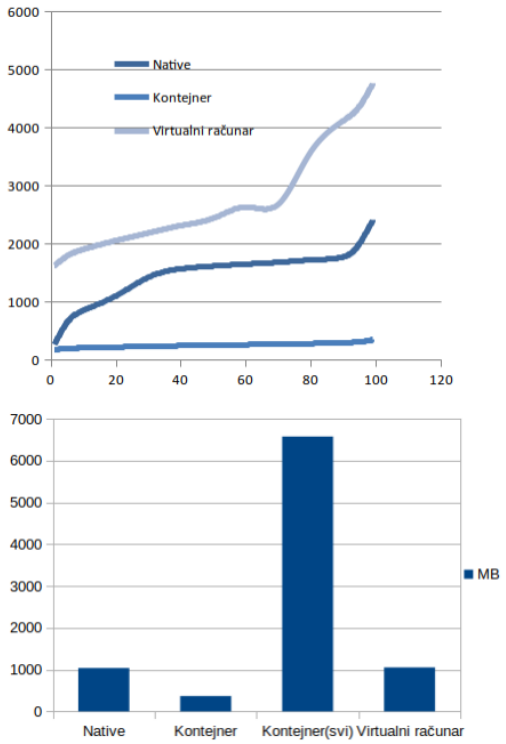
Tokom evaluacije multimibometrijski sistem je pokretan po 100 puta u svakom od okruženja sa oba skupa ograničenja računarskih resursa.

Sledeće ograničenje koje je primenjeno u evaluaciji je to da je pokretana samo jedna instanca multibiometrijskog sistema, iako distribuirana arhitektura omogućuje pokretanje više instanci multibiometrijskog sistema.

Mereni su sledeći resursi:

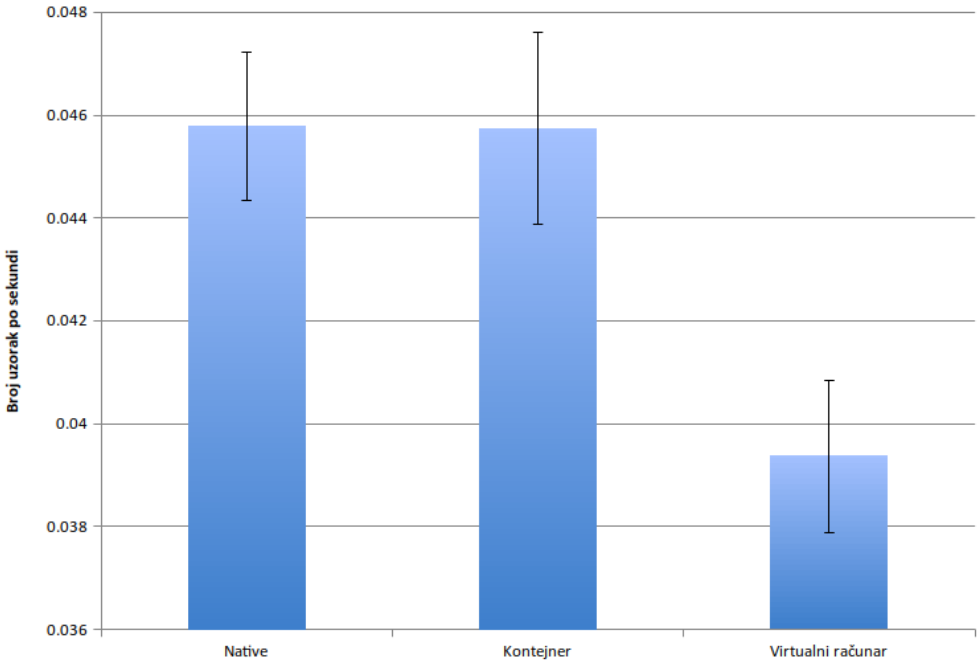
- Pristup disku
- Opterećenje procesora
- Količina utrošene memorije

Pristup disku i količina utrošene memorije:

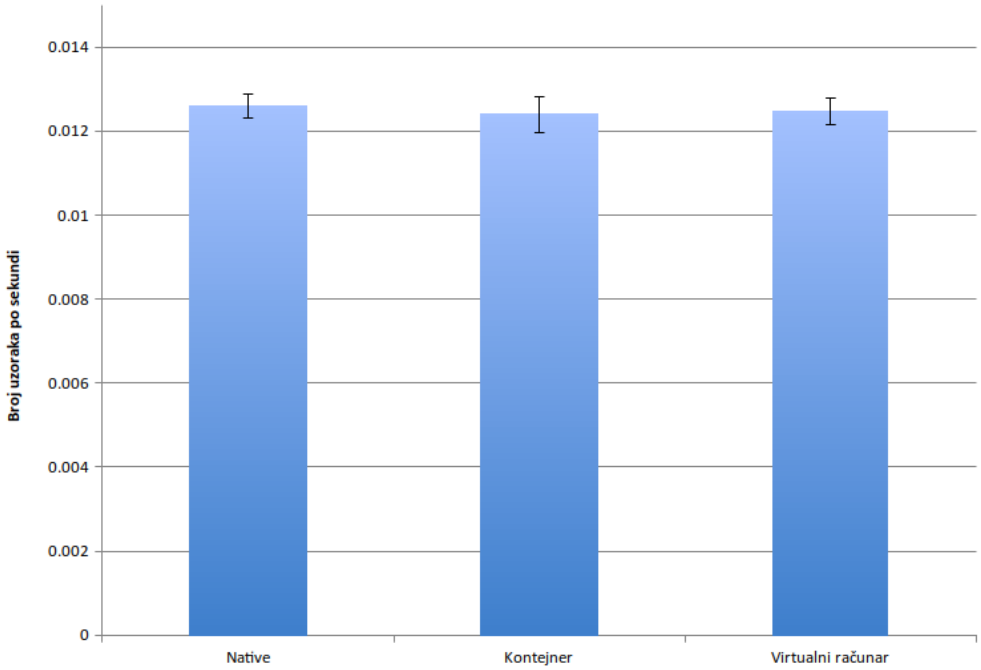


Slika 39. Pristup disku u microsekudama label i utrošak memorije u MB

Opterećenje procesora:



Slika 40. Broj obrađenih uzoraka u sekundi za 1 CPU



Slika 41. Broj obrađenih uzoraka u sekundi za 2 CPU

11.1 Diskusija i verifikacija rezultata

Na osnovu rezultata evaluacije može se zaključiti da najveći uticaj na performansu multibiometrijskog sistema ima dodeljeni broj procesora, te da maksimizacija broja procesora iznad 2 neće doneti značajniji porast performansi. To se može objasniti činjenicom da u identifakiconom modu rada (slika 5) odluka o tome da li je identitet pronađen zavisi od skorova pronađenih kandidata uzoraka, jer se kao rezultat vraća skup identiteta sa najvećim stepenom sličnosti. Da bi se došlo do tog skupa modul za pronalaženje mora da sačeka završetak svih poslova dodeljenih modulu za poređenje pre nego što nastavi sa izvršavanjem poslova modula za odlučivanje (slika 2).

Pristup disku (slika 39) dao je očekivane rezultate. Najbrži pristup disku je na fizičkom računaru. Pristup disku u okruženju u kome se izvršavaju kontejneri prolazi kroz imenski prostor i kontrolne grupe koje unose usporenje, iako su implementirane u kernelu operativnog sistema. Mapiranje pristupa skladišnom prostoru i ograničenja utroška resursima procesa u kome se izvršava kontejner su cena koja se plaća povećanim vremenima pristupa. Proces u virtualnom računaru u kome se izvršava multibiometrijski sistem prilikom pristupa disku prolazi kroz dva kernela operativnog sistema, kernel u virtualnom računaru i kernel fizičkog računara nad kojim je pokrenut virtualni računar. Mapiranja poziva pristupa disku koja se tom prilikom događaju između dodeljenog skladišta virtualnom računaru i

fizičkog disk prostora skladišta na fizičkom računaru unose povećanje vremena pristupa. Sa povećanjem broja zahteva za pristup disku raste i vreme pristupa.

Utrošak memorije se pokazuje da se nema primetne razlike između rada multibiometrijskog sistema na fizičkom i virtualnom računaru. Dok kod kontejnera utrošak memorije jednog kontejenera je skoro za jednu treći manji od fizičkog ili virtualnog računar. Kako predložena distribuirana obrada omogućuje istovremeno pokretanje više obradnih kontejnera maksimalni utrošak memorije svih kontejnera pokrenutih u *kubernetes* klasteru troši više od šest puta više memorije za obavljanje istog posla.

Kako distribuirana MMBio arhitektura omogućuje pokretanje više instanci multibiometrijskog sistema to daje mogućnost za paralelnom obradom više zahteva sa različitim biometrijskim uzorcima, dok se ne dostigne maksimalan utrošak svih računarskih resursa dodeljenih od strane IaaS-a.

12 Zaključak

Pravljenje i održavanje *Data* centara u subjektima poslovanja privatnim i državnim, danas zahteva angažovanje sve više resursa. Kako ljudskih za projektovanje, implementaciju i održavanje tako i finansijskih. U poslovanju se upotrebljava sve više i više različitih aplikacija koje prestaju da se nude kao proizvod. Proizvođači aplikativnog softvera sve češće svoje proizvode prodaju kao uslugu prebacujući svoje aplikacije u veb okruženje. Upotreba sistema za upravljanje identitima, posebno digitalnim je neminovnost. Korišćenje biometrije u kontroli pristupa je sve prisutnija na kontrolnim tačkama pristupa ka prostorijama, zgradama, objektima, postrojenjima, aerodromima, granicama između država itd. Mogućnost što brže verifikacije i identifikacije ljudi u tim tačkama sve manje će uticati na protočnost u kretanju ljudi. Brzina pretraživanja biometrijskih sistema zavisi od propusne moći i performansi računarskih sistema i telekomunikacione infrastrukture. Korišćenje *Cloud* tehnologija pri projektovanju i implementaciji je neizbežno jer omogućuje skalabilnost računarskih resursa.

Osnovna ideja ovog rada je da definiše okvir za razvoj biometrijskih distribuiranih sistema korišćenjem poznatih *cloud* tehnologija.

Pristup ostvarivanja ove ideje je baziran na uvođenju modela upravljanja performansama distribuiranog multibiometrijskog sistema. Generički model zasnovan na IaaS-u apstrahuje fizičku implementaciju *clod* tehnologija. Mogućnost automatske alokacije resursa iz IaaS-a na zahtev omogućuje skaliranje multibiometrijskog sistema bez prisustva čoveka/administratora. Istraživnja u oblasti biometrije, unapređenje postojećih i osmišljavanje novih algoritama u procesnoj obradi biometrijskih uzoraka zahtevaju upravljanje nad ogromnim količinama podataka i njhovom obradom. Obezbeđivanje biometrijskih uzoraka od zloupotrebe sprečavaju istraživače pristupu biometrijskim podacima radi evaluacije algoritama. Jedan od gradivnih elemenata svakog multibiometrijskog sistema predstavljaju upravo algoritmi pa je izgradnja repozitorijuma algoritama bitna prilikom projektovanja multibiometrijskih sistema.

Evidentiranje elemenata softverskih ekosistema i njihove međusobne veze su omoguće iznalaženje adektnih tehnologija za generisanje modela odgovarajućeg radnog okruženja i generičke arhitekture distribuiranog multibiometrijskog sistema.

Pristup ostvarivanja programskog okvira u ovoj doktorskoj disertaciji je baziran na uvođenju modela upravljanja performansama distribuiranog multibiometriskog ekosistema. Predloženi model uvođenjem heuristikaje umanjio kompleksnost modelea i omogućio optimizaciju performansi distribuiranog multibiometrijskog ekosistema u realnom vremenu.

Razvijena je metodologija koja uz datu generičku arhitekturu a zasnovana na predloženim tehnologijama omogućuje upravljanje nad performansama konkretne implementacije.

Registar šablona multibiometrijskog sistema i opisna struktura zapisa parametara šablona omogućuje da se pažnja projektanta usmeri ka multibiometrijskom sistemu. Interfejs putem koga se upravlja presformansama ekosistema ne zahteva poznavanje konkretne implementacije.

U samom radu je prikazna primena predloženog pristupa na izabranom multimodalnom sistemu. Pokazana je prednost predloženog pristupa u odnosu na dosadašnje postupke modelovanja multibiometrijskih ekosistema, sa osvrtom na korišćenje rešenja implementiranih u softveru otvorenog koda. Prvi ostvareni rezultati su verifikovali postavku fizičke infrastrukture neophodne za izgradnju multibioemtrijskog ekosistema.

12.1 Ostvareni doprinos

Rezultati istraživančkog procesa u predmetnoj oblasti kao i rad na razvoju metodologije za efektivno i efikasno upravljanje performansama implementiranog sistema omogućili su više naučnih i stručnih doprinosa. Tokom istraživačkog procesa urađen je celovit pregled istraživačke oblasti multibiometrijskih sistema uz njihovu kritičku analizu. Na osnovu urađene kritičke analize identifikovan je prostor za predlog nove naučne metode. Potom je uložen napor u razvoju i realizaciji metode za upravljanem performansama distribuiranog multibiometrijskog sistema.

U okviru predložene metode primenjene su *cloud* tehnologije u postupku projektovanja i implementacije infrastrukture nad kojom se izvršava multibiometrijski sistem. Za potrebe upravljanja performansom distribuiranog multibiometrijskog sistema razvijen je matematički model i predložene su heuristike za efektivnije i efikasnije upravljanje performansama sistema.

Za potrebe evaluacije i buduće primene predloženog modela prikazana je transformacija postojećeg multibiometrijskog sistema u njegov distribuirani oblik. Verifikacija ispravnosti postavljenog postupka ispitana je nad već primenjenim algoritmima i od njih napravljen repozitorjium algoritama i kreirana je posebna biometrijska baza podataka.

Inicijalno je ispitana mogućnost transformacije monolitne multibiometrijske aplikacije u distribuirani multibiometrijski sistem te da je od primenjenih unimodalnih algoritama moguće generisati repozitorijum algoritama koji se može održavati.

Narednom evaluacijom je ispitan uticaj promene raspoloživih resursa na ponašanje multibiometrijskog sitema. Tokom evaluacije su merene vrednosti iskorišćenosti računarskih resursa. Kao rezultat uočeno je da najviše uticaja na performansu sistema ima broj procesora koji su dodeljeni multibiometrijskom sistemu, a da povećanje broja procesora, veoma brzo, ne dovodi do daljeg rasta performansi. Promena količine memorije i disk prostora ne utiče u značajnoj meri na performansu sistema. Ova činjenica pokazuje da se maksimizacija utrošenih računarskih resursa može postići povećanjem ukupnog broja istovremenih zahteva za identifikacijom

Na osnovu evaluacije predloženog modela može se zaključiti da model ima potencijala, da postoji osnova za primenu u praksi.

12.2 Mogućnost primene

Uspešnost jednog biometrijskog sistema se, pre svega, ogleda u njegovoj primeni. Ocena performanse u velikoj meri zavisi od upotrebljivosti u realnim situacijama.

Predloženi model za upravljanje perofrmansama multibiometrijskog sistema pogoduje upotrebi u realnim situacijama. Činjenica da omogućuje sklabilnost po pitanju stepena iskorišćenosti resursa i modelu angažovanja na zahtev kvalifikuje ga za primenu u kontroli pristupa kroz kontrolne punktove. Tipični primeri su prolazak kroz kontrolne pinktove na međudržavnim granicama, aerodromima. U ovakvim situacijama, skalabilnost multibiometrijskog sitema omogućuje upotrebu tačno onoliko resursa koliko je potrebno, a posebno pri većim migracijama stanovništva tokom sezone godišnjih odmora ili praznika.

Kao dodatni vid upotrebe ovakav sistem bi mogao poslužiti istraživačkoj i naučnoj zajednici prilikom evaluacije primenjenih biometrijskih algoritama i rešenja. Kao

glavni problem realne evaluacije se često naglašava količina podataka nad kojim se vrši evaluacija, a da se pri tome ne naruši sigurnost i privatnost identiteta vlasnika biometrijskih uzoraka.

Činjenica da distribuirani multibiometrijski sistem u predloženom modelu ima mogućnost pokretanja više instanci, otvara mogućnost primene u situacijama kada postoji visoka frekvencija ljudi, kao što je raznim kulturnim i sportskim dešavanjima.

12.3 Mogući dalji pravci istraživanja

Predloženi model upravljanja performansama distribuiranog multibiometrijskog sistema pokazao se primenljivim u razvojnom okruženju. Uprkos inicijlanim dobrim rezultatima postoji puno mesta da se sistem unapredi, a pre svega u dodavanju novih funkcionalnosti.

Prvenstveno, potrebno je funkcionalnost transformacije ka distribuiranom *cloud* okruženju prilagoditi primeni u realnim uslovima. Prvi sledeći napor treba uložiti u automatizaciji i standardizaciji izgradnje repozitorijuma algoritama i izgradnje šablona kontejnera u registru šablona biometrijskih aplikacija. Rešavanje ovog problema treba tražiti u izradi modula za upravljanje nad navedenim delovima multibiometrijskog sistema, koji će omogućiti veb interfejs ka funkcijama upravljanja. Iz dobijenih rezultata utroška procesorskih resursa evidentirano je usko grlo primenjenog algoritma u modulu za fuziju. Rad na njegovom usavršavanju na iskorišćenju *cloud* okruženja, odnosno paralizaciji obrade modula za fuziju predstavlja još jedan pravac istraživanja kome se treba posvetiti.

Za potrebe daljeg ispitivanja i poboljšanja modela upravljanja performansama potrebno je omogućiti pristup *cloud* provejderima sa različitim implementacijama IaaS-a.

Konačno, potrebno je uložiti napor u dalji razvoj modela upravljanja performansi radi postizanja produkcionog stepena upotrebe.

13 Literatura

[1] Anil K Jain, Arun Ross, and Salil Prabhakar. An introduction to biometric recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(1):4–20, 2004.

[2] Elisa Bertino and Kenji Takahashi. *Identity Management: Concepts, Technologies, and Systems*. Artech House, 2011.

[3] Vasilis Boucharas, Slinger Jansen, and Sjaak Brinkkemper. Formalizing software ecosystem modeling. In *Proceedings of the 1st international workshop on Open component ecosystems*, pages 41–50. ACM, 2009.

[4] Uros Sosevic. Razvoj komunikacionih protokola za integraciju unimodalnih biometrijskih resenja. Master's thesis, Fakultet organizacionih nauka, 2013.

[5] Phillip J Windley. *Digital identity*. "O'Reilly Media, Inc.", 2005.

[6] Ivan Milenković, Uroš Šošević, and Dejan Simić. Architectures of comprehensive identity and access management. In *Electronic International Interdisciplinary Conference*, 2012.

[7] Ali M Al-Khouri. Pki in government digital identity management systems. *European Journal of ePractice*, 4:4–21, 2012.

[8] Marija Bogicevic, Ivan Milenkovic, and Dejan Simic. Identity management–a survey. *Innovative Management and Firm Performance: An Interdisciplinary Approach and Cases*, page 370, 2014.

[9] Ganesh Prasad and Umesh Rajbhandari. Identity management on a shoestring. 2012.

[10] Ruud M. Bolle, Jonathan H. Connell, and Nalini K. Ratha. Biometric perils and patches. *Pattern Recognition*, 35(12):2727–2738, 2002.

[11] Bruce Schneier. Two-factor authentication: too little, too late. *Commun. ACM*, 48(4):136, 2005.

[12] Uroš Šošević, Ivan Milenković, Miloš Milovanović, and Miroslav Minović. Support platform for learning about multimodal biometrics. *Journal of Universal Computer Science, Germany*, 19(11):1684–1700, 2013.

[13] S. Prabhakar, S. Pankanti, and A.K. Jain. Biometric recognition: Security and privacy concerns. *IEEE Security & Privacy Magazine*, 1(2):33–42, 2003.

[14] Arun Ross and Norman Poh. Multibiometric systems: Overview, case studies, and open issues. In *Handbook of Remote Biometrics*, pages 273–292. Springer, 2009.

[15] Frances Zelazny. The evolution of india's uid program. *Center for Global Development*, 2012.

[16] Amol Sharma. India launches project to id 1.2 billion people. *The Wall Street Journal*, 29, 2010.

[17] Xiaoyu Yang, Lizhe Wang, and Gregor von Laszewski. Recent research advances in e-science. *Cluster Computing*, 12(4):353–356, 2009.

[18] Thomas L Casavant and Jon G Kuhl. A taxonomy of scheduling in general-purpose distributed computing systems. *Software Engineering, IEEE Transactions on*, 14(2):141–154, 1988.

[19] Bill Godfrey. A primer on distributed computing. *DOI= http://www. bacchae. co. uk/docs/dist. html. Accessed March*, 8:2010, 2006.

[20] Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management, 2010.

[21] Matt Bishop. *Computer security: art and science*, volume 200. Addison-Wesley, 2012.

[22] ITU-T. Ngn identity management framework itu-t recomendation y.2720, 2009.

[23] Thierry Nabeth. Identity of identity. In *The Future of Identity in the Information Society*, pages 19–69. Springer, 2009.

[24] Andre Durand. Three tiers of identity. *Digit Identity World*, pages 19–69, 2002.

[25] Ted Humphreys. State-of-the-art information security management systems with iso/iec 27001: 2005. *ISO Management Systems*, 6:1, 2006.

[26] Eric Rescorla and Brian Korver. Guidelines for writing rfc text on security considerations. 2003.

[27] Tom Barton, Jim Basney, Tim Freeman, Tom Scavo, Frank Siebenlist, Von Welch, Rachana Ananthakrishnan, Bill Baker, Monte Goode, and Kate Keahey. Identity federation and attribute-based authorization through the globus toolkit, shibboleth, gridshib, and myproxy. In *5th Annual PKI R&D Workshop*, volume 4, 2006.

[28] Fernando J Corbató, Jerome H Saltzer, and Chris T Clingen. Multics: The first seven years. In *Proceedings of the May 16-18, 1972, spring joint computer conference*, pages 571–583. ACM, 1972.

[29] Davide Maltoni, Dario Maio, Anil Jain, and Salil Prabhakar. *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.

[30] Craig I Watson, Gregory P Fiumara, Elham Tabassi, Su L Cheng, Patricia A Flanagan, and Wayne J Salamon. Fingerprint vendor technology evaluation. Technical report, 2015.

[31] Anil Jain, Ruud Bolle, and Sharath Pankanti. *Biometrics: personal identification in networked society*, volume 479. Springer Science & Business Media, 2006.

[32] David Zhang, Wai-Kin Kong, Jane You, and Michael Wong. Online palmprint identification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1041–1050, 2003.

[33] Joseph P Campbell Jr. Speaker recognition: a tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, 1997.

[34] Stan Z Li and Anil Jain. *Handbook of Face Recognition*. Springer Science & Business Media, 2011.

[35] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[36] Roberto Brunelli and Tomaso Poggio. Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (10):1042–1052, 1993.

[37] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1063–1074, 2003.

[38] John Daugman. How iris recognition works. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(1):21–30, 2004.

[39] John Daugman. Recognizing people by their iris patterns. *Information Security Technical Report*, 3(1):33–39, 1998.

[40] Craig Fancourt, Luca Bogoni, Keith Hanna, Yanlin Guo, Richard Wildes, Naomi Takahashi, and Uday Jain. Iris recognition at a distance. In *Audio-and Video-Based Biometric Person Authentication*, pages 187–200. Springer, 2005.

[41] Michael Negin, M Salganicoff, and Grace G Zhang. An iris biometric system for public and personal use. *Computer*, 33(2):70–75, 2000.

[42] Mark S Nixon, John N Carter, D Cunado, Ping S Huang, and SV Stevenage. Automatic gait recognition. In *Biometrics*, pages 231–249. Springer, 1996.

[43] Fabian Monrose and Aviel Rubin. Authentication via keystroke dynamics. In *Proceedings of the 4th ACM conference on Computer and communications security*, pages 48–56. ACM, 1997.

[44] Vishvjit S Nalwa. Automatic on-line signature verification. *Proceedings of the IEEE*, 85(2):215–239, 1997.

[45] Luan L Lee, Toby Berger, and Erez Aviczer. Reliable online human signature verification systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(6):643–647, 1996.

[46] Wilson R Harrison. Suspect documents. their scientific examination. 1958.

[47] Peter Komarinski. *Automated fingerprint identification systems (AFIS)*. Academic Press, 2005.

[48] Department of Homeland Security. Privacy impact assessment for the automated biometric identification system (ident). https://www.dhs.gov/sites/default/files/publications/privacy-pia-nppd-ident-06252013_0.pdf, july 2006. (Accessed on 05/09/2016).

[49] Stan Z Li. *Encyclopedia of Biometrics: I-Z.*, volume 1. Springer Science & Business Media, 2009.

[50] Anders Eriksson and Pär Wretling. How flexible is the human voice?–a case study of mimicry. *Target*, 30(43.20):29–90, 1997.

[51] W.R. Harrison. *Suspect Documents: Their Scientific Examination*. Nelson-Hall Publ., 1981.

[52] Tsutomu Matsumoto, Hiroyuki Matsumoto, Koji Yamada, and Satoshi Hoshino. Impact of artificial gummy fingers on fingerprint systems. In *Electronic Imaging 2002*, pages 275–289. International Society for Optics and Photonics, 2002.

[53] Ton Van der Putte and Jeroen Keuning. Biometrical fingerprint recognition: don't get your fingers burned. In *Smart Card Research and Advanced Applications*, pages 289–303. Springer, 2000.

[54] Nalini K Ratha, Jonathan H Connell, and Ruud M Bolle. An analysis of minutiae matching strength. In *Audio-and Video-Based Biometric Person Authentication*, pages 223–228. Springer, 2001.

[55] Roberto Brunelli and Daniele Falavigna. Person identification using multiple cues. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(10):955–966, 1995.

[56] Elizabeth Saers Bigün, Josef Bigün, Benoît Duc, and Stefan Fischer. Expert conciliation for multi modal person authentication systems by bayesian statistics. In *Audio-and Video-based Biometric Person Authentication*, pages 291–300. Springer, 1997.

[57] Josef Kittler, Mohamad Hatef, Robert PW Duin, and Jiri Matas. On combining classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(3):226–239, 1998.

[58] Arun A Ross, Karthik Nandakumar, and Anil Jain. *Handbook of multibiometrics*, volume 6. Springer Science & Business Media, 2006.

[59] Anil K. Jain and Arun Ross. Multibiometric systems. *Communications of the ACM*, 47(1):34, 2004.

[60] Louisa Lam and Ching Y Suen. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 27(5):553–568, 1997.

[61] Diego A Socolinsky, Andrea Selinger, and Joshua D Neuheisel. Face recognition with visible and thermal infrared imagery. *Computer vision and image understanding*, 91(1):72–114, 2003.

[62] Seong G Kong, Jingu Heo, Besma R Abidi, Joonki Paik, and Mongi A Abidi. Recent advances in visual and infrared face recognition—a review. *Computer Vision and Image Understanding*, 97(1):103–135, 2005.

[63] Xin Chen, Patrick J Flynn, and Kevin W Bowyer. Ir and visible light face recognition. *Computer Vision and Image Understanding*, 99(3):332–358, 2005.

[64] Zhihong Pan, Glenn Healey, Manish Prasad, and Bruce Tromberg. Face recognition in hyperspectral images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(12):1552–1560, 2003.

[65] Robert K Rowe and Kristin A Nixon. Fingerprint enhancement using a multispectral sensor. In *Defense and Security*, pages 81–93. International Society for Optics and Photonics, 2005.

[66] Gian Luca Marcialis and Fabio Roli. Fingerprint verification by fusion of optical and capacitive sensors. *Pattern Recognition Letters*, 25(11):1315–1322, 2004.

[67] Arun Ross, Anil Jain, and James Reisman. A hybrid fingerprint matcher. *Pattern Recognition*, 36(7):1661–1673, 2003.

[68] Salil Prabhakar and Anil K Jain. Decision-level fusion in fingerprint verification. *Pattern Recognition*, 35(4):861–874, 2002.

[69] Jain Jang, Kang Ryoung Park, Jinho Son, and Yillbyung Lee. Multi-unit iris recognition system by image check algorithm. In *Biometric Authentication*, pages 450–457. Springer, 2004.

[70] Heinrich H Bülthoff, Nikolaus F Troje, Thomas Vetter, et al. Face recognition across large viewpoint changes. 2007.

[71] Harold Hill, Philippe G Schyns, and Shigeru Akamatsu. Information and viewpoint dependence in face recognition. *Cognition*, 62(2):201–222, 1997.

[72] Claude C Chibelushi, John SD Mason, and Farzin Deravi. Feature-level data fusion for bimodal person recognition. In *Image Processing and Its Applications, 1997., Sixth International Conference on*, volume 1, pages 399–403. IET, 1997.

[73] Kyong I Chang, Kevin W Bowyer, and Patrick J Flynn. An evaluation of multimodal 2d+ 3d face biometrics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(4):619–624, 2005.

[74] Conrad Sanderson and Kuldip K Paliwal. Information fusion and person verification using speech and face information. *Research Paper IDIAP-RR*, pages 02–33, 2002.

[75] Arun Ross, Samir Shah, and Jidnya Shah. Image versus feature mosaicing: A case study in fingerprints. In *Defense and Security Symposium*, pages 620208–620208. International Society for Optics and Photonics, 2006.

[76] Richa Singh, Mayank Vatsa, Arun Ross, and Afzel Noore. Performance enhancement of 2d face recognition via mosaicing. In *Automatic Identification Advanced Technologies, 2005. Fourth IEEE Workshop on*, pages 63–68. IEEE, 2005.

[77] Byungjun Son and Yillbyung Lee. Biometric authentication system using reduced joint feature vector of iris and face. In *Audio-and Video-Based Biometric Person Authentication*, pages 513–522. Springer, 2005.

[78] Arun A Ross and Rohin Govindarajan. Feature level fusion of hand and face biometrics. In *Defense and Security*, pages 196–204. International Society for Optics and Photonics, 2005.

[79] Tin Kam Ho, Jonathan J Hull, and Sargur N Srihari. Decision combination in multiple classifier systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(1):66–75, 1994.

[80] John Daugman. Combining multiple biometrics. *the Computer Laboratory at Cambridge University*, 2000.

[81] Ludmila I Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2004.

[82] Lei Xu, A Krzyzak, and CY Suen. ªmethods for combining multiple classifiers and their applications in handwritten character recognition. º *IEEE Trans. Systems, Man, and Cybernetics*, 22:418–435, 1992.

[83] Jan Bosch. From software product lines to software ecosystems. In *Proceedings of the 13th international software product line conference*, pages 111–119. Carnegie Mellon University, 2009.

[84] Slinger Jansen, Anthony Finkelstein, and Sjaak Brinkkemper. A sense of community: A research agenda for software ecosystems. In *Software Engineering-Companion Volume, 2009. ICSE-Companion 2009. 31st International Conference on*, pages 187–190. IEEE, 2009.

[85] David G Messerschmitt, Clemens Szyperski, et al. Software ecosystem: understanding an indispensable technology and industry. *MIT Press Books*, 1, 2005.

[86] Paul L Bannerman and Liming Zhu. Standardization as a business ecosystem enabler. In *International Conference on Service-Oriented Computing*, pages 298–303. Springer, 2008.

[87] Sjaak Brinkkemper, Ivo Van Soest, and Slinger Jansen. Modeling of product software businesses: Investigation into industry product and channel typologies. In *Information Systems Development*, pages 307–325. Springer, 2009.

[88] Eric S Raymond. *The Cathedral & the Bazaar: Musings on linux and open source by an accidental revolutionary*. "O'Reilly Media, Inc.", 2001.

[89] C Walton. The open source software ecosystem. *IIIA Communications. Institut d'Investigacion en Intel. ligencia Artificial, IIIA, Barcelona*, 2002.

[90] Björn Lundell, Bo Forssten, Jonas Gamalielsson, Henrik Gustavsson, Robert Karlsson, Christian Lennerholt, Brian Lings, Anders Mattsson, and Erik Olsson. Exploring health within oss ecosystems. In *First International Workshop on Building Sustainable Open Source Communities (OSCOMM 2009), Skövde, Sweden*, 2009.

[91] Madhu Chetty and Rajkumar Buyya. Weaving computational grids: how analogous are they with electrical grids? *Computing in Science & Engineering*, 4(4):61–71, 2002.

[92] Ian Foster, Carl Kesselman, et al. The grid 2: Blueprint for a future computing infrastructure. *Waltham: Morgan Kaufmann Publishers*, 2004.

[93] Dejan S Milojicic, Vana Kalogeraki, Rajan Lukose, Kiran Nagaraja, Jim Pruyne, Bruno Richard, Sami Rollins, and Zhichen Xu. Peer-to-peer computing. 2002.

[94] J Geelan. Twenty experts define cloud computing. *Cloud Comput. J. SYS-CON Media Inc*, 2008.

[95] Gregory F Pfister. *In search of clusters*, volume 2. Prentice Hall PTR Englewood Cliffs, 1998.

[96] Rajkumar Buyya. High performance cluster computing: Architectures and systems, volume i. *Prentice Hall, Upper SaddleRiver, NJ, USA*, 1:999, 1999.

[97] R Buyya. Economic paradigm for service-oriented grid computing. In *The INT Media's Grid Computing Planet Conference and Expro, San Jose, California, USA*, 2002.

[98] Peter Mell, Tim Grance, et al. The nist definition of cloud computing. 2011.

[99] André B Bondi. Characteristics of scalability and their impact on performance. In *Proceedings of the 2nd international workshop on Software and performance*, pages 195–203. ACM, 2000.

[100] Rajkumar Buyya, Chee Shin Yeo, and Srikumar Venugopal. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In *High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on*, pages 5–13. Ieee, 2008.

[101] Ivona Brandic, Dejan Music, Philipp Leitner, and Schahram Dustdar. Vieslaf framework: Enabling adaptive and versatile sla-management. In *International Workshop on Grid Economics and Business Models*, pages 60–73. Springer, 2009.

[102] Alexander Lenk, Markus Klems, Jens Nimis, Stefan Tai, and Thomas Sandholm. What's inside the cloud? an architectural map of the cloud landscape. In

Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, pages 23–31. IEEE Computer Society, 2009.

[103] Rob Pike, Dave Presotto, Ken Thompson, Howard Trickey, and Phil Winterbottom. The use of name spaces in plan 9. In *Proceedings of the 5th workshop on ACM SIGOPS European workshop: Models and paradigms for distributed systems structuring*, pages 1–5. ACM, 1992.

[104] Eric W Biederman and Linux Networx. Multiple instances of the global linux namespaces. In *Proceedings of the Linux Symposium*, volume 1, pages 101–112. Citeseer, 2006.

[105] Rami Rosen. Resource management: Linux kernel namespaces and cgroups.

[106] Jerome H Saltzer and Michael D Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, 1975.

[107] Gerald J Popek and Robert P Goldberg. Formal requirements for virtualizable third generation architectures. *Communications of the ACM*, 17(7):412–421, 1974.

[108] Peter Peer, Jernej Bule, Jerneja Zganec Gros, and Vitomir Struc. Building cloud-based biometric services. *Informatica*, 37(2):115, 2013.

[109] Kevin C Mangold. Biometrics in a networked world. Technical report, 2012.

[110] Raj Jain. *The art of computer systems performance analysis : techniques for experimental design, measurement, simulation, and modeling*. Wiley, 1991.

[111] Bojan Jovanović, Ivan Milenković, Marija Bogićević, and Dejan Simić. The archutecture of integrated identity management and multimodal biometric system, 2014.

[112] Bojan Jovanović, Ivan Milenković, Marija Bogićević, and Dejan Simić. Extending identity management system with multimodal biometric authentication. *Computer Science and Information Systems*, (00):3–3, 2016.

Biografija autora

Bojan Jovanović je rođen 08.11.1968. godine u Leskovcu, gde je završio osnovnu i srednju školu. Diplomirao je 1998. godine na Fakultetu organizacionih nauka u Beogradu. Od 1999. do 2001. godine radio je kao sistem inženjer na Fakultetu organizacionih nauka u Laboratoriji za AOP. Od 2001. godine radi kao asistent pri Katedri za informacione sisteme i tehnologije na predmetima „Ekspertni sistemi“, „Organizacija računara i operativni sistemi“, „Distribuirani informacioni sistemi“, „Multimedijalni informacioni sistemi“, „Arhitektura računara i operativni sistemi“, „Računarske mreže i telekomunikacije“, „Distribuirani računarski sistemi“, „Zaštita računarskih sistema“.

Pored saradničkih obaveza u održavanju nastave, Bojan Jovanović je nastavio sa praktičnim radom na održavanju i uvođenju računarskih sistema i internet servisa u Računskom centru Fakulteta organizacionih nauka, važnoj organizacionoj jedinici Fakulteta. Na poslediplomskim studijama je položio svih 9 ispita sa prosečnom ocenom deset i magistrirao sa temom pod naslovom “Model tankih klijenata pod Linuks operativnim sistemom u organizaciji računarskih učionica”. и стекао академски назив магистра наука.

Bojan Jovanović se aktivno bavi računarstvom od 1985. godine. Jedan je od pionira uvođenja Unix i Linux operativnih sistema u ovom regionu. Sa operativnim sistemom Unix upoznao se 1991., a sa Linux opprativnim sistemom 1993. godine i od tada mu Unix/Linux administracija i pisanje sistemskog softvera postaje predmet interesovanja. U februaru 2005. godine stiče sertifikate „Red Hat Certified Engeneer“, „Red Hat Certified Instructor“, „Red Hat Certified Examiner“. Odlično poznaje „SCO Unix OpenServer/Unixware“, „Digital Unix/True64“ i „Solaris“, za koje je pisao sistemski softver korišćen u administraciji tih operativnih sistema u Laboratoriji za AOP, odnosno Računskom centru.

Bojan Jovanović učestvovao je u realizaciji sedam istraživačkih projekata izvedenih na Fakultetu organizacionih nauka, odnosno Inovacionog centra Fakulteta organizacionih nauka. Glavni je organizator i ovlašćeni instruktor u Edukacionom centru za Red Hat Linux program. Bio je organizator projekta „Lokalizacija Fedora Linuks distribucije“ koji se izvodio na Fakultetu organizacionih nauka pod okriljem Ministarstva telekomunikacija i informatičkog društva. Redovno učestvuje u organizovanju i realizaciji naučno-stručnog skupa InfoTech.

Прилог 1.

Изјава о ауторству

Име и презиме аутора Бојан Јовановић

Број индекса _____

Изјављујем

да је докторска дисертација под насловом

Управљање перформансама дистрибуираног мултибиометријског екосистема

- резултат сопственог истраживачког рада;
- да дисертација у целини ни у деловима није била предложена за стицање друге дипломе према студијским програмима других високошколских установа;
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио/ла интелектуалну својину других лица.

Потпис аутора

У Београду, 30.05.2018.

_____

–

Прилог 2.

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора _Бојан Јовановић_____

Број индекса _____

Студијски програм _Информациони системи_____

Наслов рада _Управљање перформансама дистрибуираног мултибиометријског екосистема_

Ментор _Проф. Др Душан Старчевић, редовни професор ФОН-а _____

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла ради похрањена у **Дигиталном репозиторијуму Универзитета у Београду.**

Дозвољавам да се објаве моји лични подаци везани за добијање академског назива доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис аутора

У Београду, _30.05.2018._____

_____

Прилог 3.

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

Управљање перформансама дистрибуираног мултибиометријског екосистема

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигиталном репозиторијуму Универзитета у Београду и доступну у отвореном приступу могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство (CC BY)

2. Ауторство – некомерцијално (CC BY-NC)

3. Ауторство – некомерцијално – без прерада (CC BY-NC-ND)

4. Ауторство – некомерцијално – делити под истим условима (CC BY-NC-SA)

5. Ауторство – без прерада (CC BY-ND)

6. Ауторство – делити под истим условима (CC BY-SA)

(Молимо да заокружите само једну од шест понуђених лиценци.
Кратак опис лиценци је саставни део ове изјаве).

Потпис аутора

У Београду, 30.05.2018

_____

1. **Ауторство**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. **Ауторство – некомерцијално**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. **Ауторство – некомерцијално – без прерада**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. **Ауторство – некомерцијално – делити под истим условима**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. **Ауторство – без прерада**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. **Ауторство – делити под истим условима**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.