UNIVERZITET SINGIDUNUM
BEOGRAD
DEPARTMAN ZA POSLEDIPLOMSKE STUDIJE

DOKTORSKA DISERTACIJA

ENERGETSKI EFIKASNO REŠENJE PLATFORME ZA MOBILNO UČENJE U CLOUD COMPUTING OKRUŽENJU

Mentor: prof. dr Dragan S. Marković

Kandidat: Oliver Popović, master
Broj indeksa: 460066/2009

Beograd, 2016.

SADRŽAJ

Spisak slika

Spisak tabela

Spisak programskog koda

Apstrakt

Edukacija u 21.veku predstavlja ključni faktor ekonomskog i socijalnog razvoja društva. Većina tradicionalnih metoda namenjenih obrazovanju nisu više prikladni za tu svrhu jer nisu u mogućnosti da ispune potrebe studenata. Mobilno učenje predstavlja novi edukacioni kanal, kojim se omogućuje pristup sa bilo kog mesta u bilo koje vreme. Ubrzana integracija informaciono komunikacionih tehnologija uticala je na mnoge aspekte naših života. Oblast obrazovanja nije izuzetak takvog trenda. Razvoj računara, internet servisa, mobilnih uređaja, a posebno *cloud computinga,* pružilo je obrazovnim ustanovama mnogo novih mogućnosti.

Abstract

In the 21st century, education represents a key factor of economic and social development of a community. The majority of traditional methods intended for education are no longer suitable for it, since they are unable to meet the students' needs. Mobile learning represents a new educational channel, which allows access from any place at any time. The area of education is no exception to this trend. The development of computers, internet services, mobile devices, and in particular cloud computing has provided the educational institutions with many new features.

1. UVOD

Cloud computing predstavlja evoluciju na polju računarske tehnike i nauke. *Cloud* je paradigma distribuiranog računarstva koja se potpuno oslanja na internet. On predstavlja skup međusobno povezanih, skalabilnih i virtuelizovanih računara, koji su predstavljeni kao dinamički unificirani računarski resursi koji se mogu iznajmiti na korišćenje [1].

Princip *cloud* tehnologije zasniva se na tome da svaka usluga koja je neophodna klijentima, bude dostupna u svakom trenutku i na svakoj lokaciji, naravno, uz prisustvo interneta. Vrste usluga koje se isporučuju mogu biti infrastruktura, operativni sistemi, razvojna okruženja, aplikacije, skladišta podataka ili nešto drugo. Može se reći da je *cloud computing* koncept dostavljanja usluga umesto konkretnog proizvoda.

Povećanjem obima upotrebe *cloud computinga* i servisa koji se oslanjaju na *cloud* platformu pojavio se problem potrošnje energije koja se koristi za rad ovih sistema. Danas, minimizacija potrošnje energije bez gubitka na performansama predstavlja jedan od najvećih izazova u ovoj oblasti. Povećanje energetske efikasnosti svakog *cloud* sistema je problem koji zahteva konstantno razmatranje. Samo detaljnom evaluacijom svakog dela *cloud computinga* može se doći do željenog sistema koji uvažava principe energetske efikasnosti.

Hipervizori predstavljaju ključni deo svake *cloud* platforme i oni mogu znatno uticati na potrošnju energije i pravilnu alokaciju resursa. Polise za raspoređivanje takođe utiču na povećanje potrošnje resursa. Primena kompromisnog rešenja koje će omogućiti prihvatljivo raspoređivanje resursa u sistemu uz minimalnu potrošnju energije predstavlja značajan deo istraživanja efikasnosti. Sastavni deo svake *cloud* platforme predstavlja i njen klijentski deo. Klijentski deo platforme je značajan činilac koji utiče na efikasnost platforme u svakom smislu, a optimizacija potrošnje resursa i primena energetski efikasnih klijenata predstavlja jedan od najvažnijih faktora koje treba istražiti.

Primenom navedenih tehnologija, pored ostalih servisa, omogućen je i razvoj elektronskog učenja (*e-learning*). Elektronsko učenje se zasniva na efikasnoj isporuci servisa upotrebom sistema za upravljanje učenjem (*learning management system* – LMS), kako bi bili obezbeđeni materijali za učenje, uz jednostavan pristup i jednostavan korisnički interfejs [2].

Pojava mobilnih uređaja, a pre svega *smartphone* i tablet uređaja, kao i širokopojasnog mobilnog umrežavanja, dovela je do transformacije koncepta elektronskog učenja (*e-learning*) u koncept mobilnog učenja (*m-learning*). Mobilnim učenjem (*m-learning*) se smatra svaki vid učenja kada se student ne nalazi na fiksnoj, predefinisanoj lokaciji, koje se dešava u trenutku kada student koristi prednosti učenja uz pomoć mobilnih tehnologija [3]. Mobilno

učenje na taj način postaje prirodno proširenje tradicionalnih načina učenja, u vidu jedne forme elektronskog učenja. Za razliku od tradicionalnog načina nastave, kod mobilnog učenja proces nastave i učenja se može odvijati na neformalan način u neškolskom okruženju. Ovakav pristup dovodi do relaksirajućeg pristupa učenju, povećanja nivoa saradnje između nastavnika i studenata, dobijanja trenutnih povratnih informacija, kao i znatno veću prenosivost digitalnog sadržaja [4]. Studentu je omogućeno da uči u trenutku kada mu to najviše odgovara, uz upotrebu znatno kompaktnijih uređaja. Pomenuti proces učenja može obuhvatati gledanje video materijala sa predavanja, čitanje elektronskih knjiga, prezentacija kao i rešavanje testova. Uz pomoć mobilnog učenja, studenti sa invaliditetom takođe mogu mnogo lakše i kvalitetnije pristupiti procesu učenja. Danas, nezaobilazni trend razvoja obrazovanja usmeren je na personalizovani sadržaj za učenje [5], a mobilno okruženje i *m-learning* omogućuju takav pristup.

1.1. Ciljevi istraživanja

Cilj ove disertacije je proučavanje prednosti mobilnog učenja, pružanje analize principa projektovanja interfejsa, obezbeđenje taktike za rešavanje zajedničkih problema razvoja mobilnih aplikacija i razvoj mobilne aplikacije za Android platformu koja će obezbediti funkcije koje podržavaju učenje na daljinu.

Aplikacija će omogućiti svakom nastavniku da kreira kurs pod određenim nazivom, u okviru željenog studijskog programa. Takođe, nastavnik može da deli fajlove sa studentima prijavljenim na kurs, dodaje obaveštenja ili kreira testove.

Na taj način, studenti mogu učiti i rešavati testove. Ove funkcije će pomoći nastavniku da brzo prikupi odgovore i dobije rezultate testova, dok studenti mogu slobodno da vide rezultate testova.

Pored navedenog, cilj istraživanja je i analiza *cloud* sistema i evaluacija uticaja na potrošnju energije, kao i mogućnosti smanjenja potrošnje energije u *cloud* sistemima. U pojedinim segmentima evaluacija je obavljena u ograničenim uslovima, usled nemogućnosti merenja potrošnje kod provajdera.

Praktični cilj ovog istraživanja je dizajniranje i implementacija mobilne platforme namenjene obrazovanju u visokoškolskoj ustanovi.

Na osnovu dosadašnjih naučno-istraživačkih doprinosa, kao i istraživanja u ovoj doktorskoj disertaciji, biće predloženo najoptimalnije rešenje u cilju povećanja efikasnosti sistema upotrebom mobilne platforme, čiji će se efekti najviše odraziti u obrazovanju.

Cilj istraživanja jeste razvoj i usavršavanje sistema namenjenog učenju koji koristi *cloud* sistem kroz nalaženje novih metoda primene u obrazovanju.

1.2. Predmet istraživanja

Da bi se postigli navedeni ciljevi istraživanja, neophodno je definisati više predmeta istraživanja. To su:

1. Proučavanje oblasti *cloud computinga*,
2. Pregled *cloud computing* platformi,
3. Pregled energetski efikasnih polisa,
4. Pregled energetski efikasnih hipervizora,
5. Pregled energetski efikasnih klijentskih rešenja koja se mogu koristiti u učenju,
6. Pregled bezbednosnih problema i potencijalnih rešenja vezanih za *cloud*,
7. Pregled literature o teoriji mobilnog učenja,
8. Pregled karakteristika i faktora koji utiču na mobilno učenje,
9. Proučavanje perspektive, elemenata i argumenata za i protiv mobilnog učenja,
10. Proučavanje principa dizajniranja koje mobilna aplikacija treba da uvažava,
11. Proučavanje platformi koje dolaze u obzir za izvršavanje mobilne aplikacije,
12. Obezbeđivanje taktike za rešavanje zajedničkih problema u razvoju mobilnih aplikacija za učenje kao vodič novim programerima,
13. Dizajn i razvoj mobilne aplikacije koja treba da pomogne nastavnicima da uspostave direktnu komunikaciju između učesnika kursa bez obzira na ulogu i geografsku lokaciju, bez prekidanja predavanja,
14. Evaluacija korisnosti aplikacije. Evaluacija će se zasnivati na studentima i nastavnicima koji će koristiti mobilnu aplikaciju u realnom vremenu.

Predmet istraživanja ovog rada jesu komercijalne i nekomercijalne *cloud computing* platforme, čijom će se analizom odrediti najoptimalniji uslovi za stvaranje energetski efikasnog okruženja. Kada se govori o efikasnom okruženju, neophodno je razmotriti polise raspoređivanja, hipervizore, ali i klijentski deo platforme koji obuhvata desktop računare, tanke klijente i *smartphone/*tablet uređaje.

Pored energetskih aspekata, predmet istraživanja su i mobilni sistemi namenjeni učenju, kako bi se utvrdile poželjne karakteristike koje jedan savremeni sistem treba da poseduje. Analiza *cloud* i mobilnih sistema predstavljala bi osnovu za postavljanje mobilnog sistema namenjenog učenju, uz uvažavanje pedagoških metoda za razvoj takve vrste sistema. U ovom

trenutku većina konvencionalnih oblika nastave nije pogodna za potrebe socijalnog napretka i razvoj obrazovanja nije u stanju da uhvati korak sa promenama u učenju. U tradicionalnom režimu učenja zasnovanom na vebu, izgradnja i održavanje sistema se nalaze unutar obrazovne institucije, kao deo informacionog sistema. Ovo može stvoriti probleme u vidu velikih investicija, ali bez razvojnog potencijala. Aktuelno rešenje koje bi moglo rešiti navedene probleme jeste upotreba *cloud computing* tehnologije.

Predmet istraživanja su i bezbednosni aspekti *cloud* sistema i njihov uticaj na opšte karakteristike istog.

1.3. Hipoteza

Osnovna hipoteza koja je postavljena i dokazana u doktorskoj disertaciji je da upotreba mobilne platforme za obrazovanje može doprineti tome da ciljna grupa ima lakši pristup nastavnom sadržaju. Iako mobilna platforma ima određena ograničenja u pogledu komfora korišćenja, ona je opšteprihvaćena od strane mladih kada su u pitanju neobavezni sadržaji. Mobilna platforma ima manju potrošnju električne energije u odnosu na druge računarske platforme koje se tradicionalno koriste u obrazovanju. Postavljena je hipoteza da će mladi prihvatiti mobilnu platformu i u obrazovnom procesu s obzirom na mogućnost korišćenja nastavnih sadržaja uz jedini ograničavajući uslov – pristup internet vezi. Dakle, mobilna platforma olakšava učenje i doprinosi povećanju prolaznosti u ispitnim rokovima, dok korišćenje mobilne *cloud computing* tehnologije u obrazovnom procesu utiče na efikasnost, ekonomičnost i skalabilnost tog procesa. Imajući u vidu da je softver potpuno usmeren ka upotrebi u obrazovanju, ciljna grupa koja koristi ovaj softver bi trebalo da postigne bolje rezultate od grupe koja ga ne koristi u nastavi. Razlika u postignutim rezultatima se može izmeriti pomoću faktora kao što su: broj studenata koji prisustvuje nastavi, broj studenata izašlih u prvom ispitnom roku, kao i procenat studenata koji su položili ispit. Upotrebom savremene IT infrastrukture povećava se isplativost i efikasnost obrazovanja. Primenom *cloud computing* tehnologije i mobilnih platformi stvaraju se nove metode razvoja obrazovanja.

1.4. Metode istraživanja

Metode koje će biti korišćene u razmatranju i rešavanju problema jesu uporedne analize, studije slučajeva, UML tehnologija za razvoj aplikacija, proučavanje međunarodne literature iz oblasti *cloud* sistema, bezbednosti, energetske efikasnosti, kao i kritička analiza mobilnih platformi koje se koriste u obrazovanju.

Kako bi se prilikom sprovođenja statističkog istraživanja prikupili podaci na sistematski način, koristiće se statističke analize, istraživanja i obrada podataka uz pomoć računarskog softvera.

Rezultati istraživanja su prezentovani tekstualno, opisivanjem i prikazom više tabela, slika i dijagrama sa uporednim rezultatima. Istraživanje je interdisciplinarno, jer uključuje metodologiju, statistiku, pedagogiju, računarstvo i druge naučne discipline.

1.5. Struktura rada

Doktorska disertacija se sastoji od devet glavnih celina i literature. U prvom poglavlju su data uvodna razmatranja. U drugom delu rada definisan je pojam i funkcija *cloud computing* platforme. Treće poglavlje opisuje komercijalne i nekomercijalne platforme koje imaju najveći uticaj na oblast *cloud computinga*. U posebnom delu opisana je virtuelizacija kao suštinski deo *cloud computing* platforme. Četvrto poglavlje obuhvata sigurnost *cloud* sistema i opisuju se potencijalni bezbednosni rizici, uz evaluaciju sistema za zaštitu. U petom poglavlju opisani su načini povećanja energetske efikasnosti primenom predefisanih polisa i algoritama u *cloudu*. U šestom delu opisane su mogućnosti povećanja energetske efikasnosti upotrebom tankih klijenata. Sedmo poglavlje se bavi elektronskim učenjem i arhitekturom tradicionalnih i savremenih modela elektronskog učenja. U osmom poglavlju opisana su savremene mobilne aplikacije koje se koriste u edukaciji. Kao praktičan primer takve mogućnosti predstavljena je nova mobilna platforma za primenu u nastavnom procesu i procesu provere znanja studenata. U devetom poglavlju data su zaključna razmatranja.

2. DEFINISANJE *CLOUD COMPUTING* OKRUŽENJA

Cloud computing predstavlja način isporučivanja resursa, aplikacija i usluga na daljinu, uz upotrebu interneta. *Cloud computing* može obuhvatati različite vrste usluga namenjene mnogim segmentima društva i poslovanja. Iz tog razloga ne postoji jedinstvena definicija *cloud computinga*.

Gartner [6] definiše *cloud computing* kao oblast računarstva u kojoj se visoko skalabilni računarski kapaciteti obezbeđuju brojnim eksternim klijentima u vidu usluge isporučene putem interneta.

Forester [7] definiše *cloud computing* kao apstrahovanu, veoma skalabilnu i kontrolisanu kompjutersku infrastrukturu koja hostuje aplikacije namenjene krajnjim korisnicima i čije se usluge naplaćuju na bazi ostvarene potrošnje.

Badger i autori iz Nacionalnog Instituta za tehnologiju i standarde (*National Institute of Standards and Technology* - NIST) SAD-a dali su veoma koncizan opis *cloud computinga* i predstavlja jednu od najprihvaćenijih definicija u naučnoj i stručnoj javnosti. *Cloud computing* definišu kao „model za omogućavanje pouzdanog mrežnog pristupa na zahtev ka deljenom prostoru konfigurabilnih računarskih resursa (npr. mreže, serveri, skladišta, aplikacije i servisi) koji mogu biti brzo obezbeđeni i pruženi sa minimalnim upravljačkim aktivnostima ili interakcijom servis provajdera.“ [8]

Sam termin „*cloud computing*“ može obuhvatati raznovrsne računarske tehnologije, sisteme, usluge, ali i modele razvoja i poslovanja.

2.1. Koncept cloud computing okruženja

Cloud computing predstavlja sledeći korak u razvoju računarstva orijentisanog ka uslugama, gde se virtuelizovani resursi iznajmljuju korisnicima putem interneta, i to u vidu dinamički skalabilnog servisa. Na taj način, korisnik može promeniti vrstu i obim usluge, dok se istovremeno ne remeti upotreba iznajmljenih resursa. Upotreba *cloud computinga* omogućava smanjenje gotovo svih troškova vezanih za upotrebu informacionih tehnologija.

Cloud computing svakako predstavlja novi koncept računarstva i poslovanja. Međutim, u suštini je zasnovan na nekim postojećim modelima distribuiranog računarstva. Ti modeli su:

- mrežno računarstvo (*grid computing*) – distribuirani sistem koji predstavlja skup računarskih resursa. Iako se resursi nalaze na različitim fizičkim lokacijama, oni su međusobno umreženi i mogu zajedno raditi u rešavanju određenog zadatka.

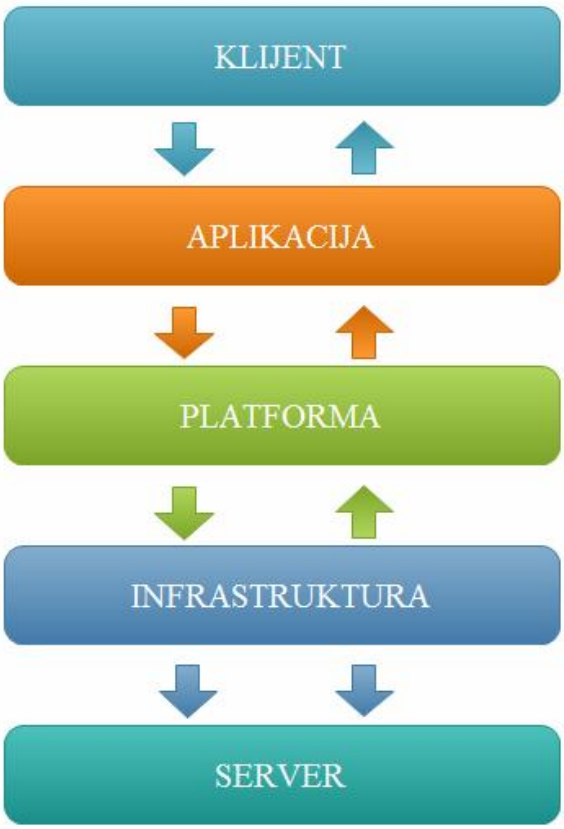
- računarstvo u vidu usluge (*utility computing*) - servisni model u kome servis provajder iznajmljuje računarske resurse i infrastrukturu po potrebi i vrši naplatu u odnosu na obim upotrebe. Kao i mrežno računarstvo, ovaj model teži ka efikasnoj upotrebi resursa uz minimizaciju pripadajućih troškova.
- usluge na zahtev (*on-demand services*) – koncept usluga gde korisnik dobija ono što želi u trenutku kada mu to najviše odgovara, bez ikakvih odlaganja. Prvobitno razvijen za isporuku video sadržaja na zahtev, kasnije se proširio na isporuku drugih elektronskih sadržaja i inkorporirao kao važan deo koncepta *cloud computinga*.
- softver u vidu usluge (*software-as-a-service*) – termin je ranije bio poznat kao aplikacioni servis provajderi (Application Service Providers – ASP), ali je osnovna ideja bila slična – korisnici su plaćali upotrebu aplikacije kojoj se pristupalo putem veb-sajta. Međutim, ovakav pristup je ranije imao problema zbog nedostatka elastičnosti infrastrukture i upotrebe aplikacija koje nisu bile prilagođene ovakvom načinu rada.

Koncept je nastao iz ideje mogućnosti uštede, na taj način što se računarski resursi (procesorski resursi, memorija, prostor za skladištenje podataka i sl.) ne bi kupovali nego iznajmljivali. Ovakva vrsta usluge se naplaćuje na osnovu obima i vremena upotrebe resursa.

Kao što je navedeno, *cloud computing* se razlikuje od klasičnih servisa po dinamičnoj i elastičnoj arhitekturi, što omogućuje trenutno prilagođavanje uvećanim ili umanjenim potrebama korisnika, uz plaćanje samo onoga što je potrošeno, tj. iznajmljeno..

Veliki deo današnje celokupne infrastrukture *cloud computinga* se koristi kao javna cloud platforma za servise kao što su *Google Search*, *Facebook*, *Microsoft Hotmail* ili *Flickr*.

Na slici 2.1 dat je šematski prikaz koncepta *cloud computing* okruženja.



Slika 2.1 - Koncept *cloud computing* okruženja

Raslojavanjem po funkcijama u *cloud computing* okruženju, konceptualno se kreira pet slojeva: Klijent, aplikacija, platforma, infrastruktura i server.

Klijent: *Cloud* klijent predstavlja krajnjeg korisnika iznajmljenih servisa koji pristupa *cloud* platformi pomoću svog računara, tankog (thin) klijenta ili nekog od prenosnih uređaja *(smartphone,* tablet)*;*

Aplikacija: Servis koji omogućuje isporuku softvera u vidu servisa (*Software as a Service* - SaaS). Isporuka softvera se vrši preko Interneta, čime prestaje potreba za instalacijom softvera na vlastitom računaru;

Platforma: *Cloud* servisne platforme (PaaS – *Platform as a Service*) pružaju operativnu platformu ili skup rešenja kao servis, koristeći *cloud* infrastrukturu i mnogobrojne aplikacije. Ova vrsta servisa olakšava razvoj aplikacija, nema troškova ni složenosti kupovanja i uređivanja određenog hardvera/softvera.

Infrastruktura: *Cloud* servisna infrastruktura (IaaS – *Infrastructure as a Service*) dostavlja računarsku infrastrukturu, platformsku virtuelizaciju sa određenim skladištem i mrežom.

Umesto kupovine servera, softvera, prostora za skladištenje podataka, klijenti su u mogućnosti da sve navedeno iznajme kao "*all in 1*" (sve u jednom) servis;

Server: Server se zasniva na skupu hardvera i softvera koji je namenjen upravljanju celokupnom *cloud* infrastrukturom.

Da bi provajder posedovao *cloud computing* rešenje potrebno je da poseduje:

- Servere za izvršavanje aplikacija,
- Diskove za skladištenje podataka,
- Sisteme za zaštitu i pravljenje rezervnih kopija podataka,
- Veze sa klijentima.

Kako bi korisnik pristupio *cloud* sistemu potrebno je da poseduje:

- Računar sa *veb čitačem*,
- Vezu sa internetom,
- Pretplatu kod provajdera koji pruža *cloud* usluge.

2.2. Arhitektura cloud computinga

Arhitekturu jednog *cloud* sistema mogu činiti aplikacije, podaci, razvojna okruženja, *middleware*, operativni sistem, virtuelizacija, serveri, skladišta za podatke i mreže [9].

Aplikacije se nalaze na vrhu *cloud* platforme. Koriste se od strane klijenata radi obavljanja određenih poslova, za obradu podataka, dokumenata i sl. To mogu biti e-mail klijenti, baze podataka, *office* paket proizvoda za manipulaciju dokumentima i drugim podacima.

Pod podacima se podrazumevaju kolekcije informacije koje koriste aplikacije. To mogu biti dokumenti, multimedijalni fajlovi, log fajlovi i sl.

Razvojna okruženja predstavljaju drugi nivo softverske platforme koja omogućuje kreiranje i izvršavanje standardizovanih aplikacija (Sun Java, Microsoft.NET, itd.).

Kako bi ceo sistem funkcionisao neophodno je da postoji utvrđeni set pravila, odnosno protokola, gde se koristi specijalna vrsta softvera koji se zove *middleware*. *Middleware* omogućuje mrežnim računarima da međusobno komuniciraju, aplikacijama da se povezuju sa bazama podataka i sl.

Operativni sistem predstavlja softversku platformu koja se koristi kao podrška prethodno navedenim slojevima (MS Windows, razne distribucije Linuxa, Unix, MacOS).

Virtuelizacija predstavlja hardver i/ili softver koji omogućuje dinamičku alokaciju servera, skladišnog prostora i računarskih mreža.

Serveri predstavljaju realni računarski hardver.

Skladišta za podatke – hard disk uređaji na kome je smešten sav softver i drugi podaci.

Mreže – uređaji za prenos podataka (kablovi ili uređaji za bežično povezivanje), ruteri i ostala oprema koja omogućuje međusobno povezivanje računara.

U situaciji kada je provajder ujedno i korisnik celokupnog sistema, za takav *cloud* sistem se kaže da je napravljen lokalno (*on-premises*). U ostalim situacijama kada se deo infrastrukture iznajmljuje klijentima, takvi iznajmljeni sistemi predstavljaju servisne modele usluga.

2.2.1. Servisni modeli usluga

Servisni modeli usluga predstavljaju osnovno viđenje arhitekture jednog *cloud computing* okruženja. Postoje tri servisna nivoa usluga u *cloudu* (slika 2.2):

- Infrastuktura u vidu servisa (IaaS),
- Platforma u vidu servisa (PaaS) i
- Softver u vidu servisa (SaaS).

Svi ostali nivoi usluga predstavljaju neku od kombinacija pomenutih servisnih nivoa, i oni neće biti razmatrani.

Infrastuktura u vidu servisa (IaaS) – računarska infrastruktura, kao što su procesorski resursi, memorija i skladišta podataka, ostvarena u vidu *cloud computinga*, obično upotrebom virtuelizacije. Korisnik ne kontroliše tu iznajmljenu infrastrukturu, ali ima kontrolu nad operativnim sistemima koji se nalaze na toj infrastrukturi, nad skladištem i drugim aplikacijama. Takođe, korisnik može imati ograničenu kontrolu nad određenim mrežnim uređajima (npr. *firewall* i sl.). Neke od najpoznatijih IaaS platformi su *Amazon EC2* i *vCloud*.

Platforma u vidu servisa (PaaS) – To su platforme koje mogu biti korišćene za razvoj aplikacija, ali i za realizaciju aplikacija napravljenih od strane klijenata ili partnera provajdera platforme. Te aplikacije su kreirane upotrebom programskih razvojnih okruženja, biblioteka, servisa i alata koje ima provajder (*Windows Azure, Google app engine*).

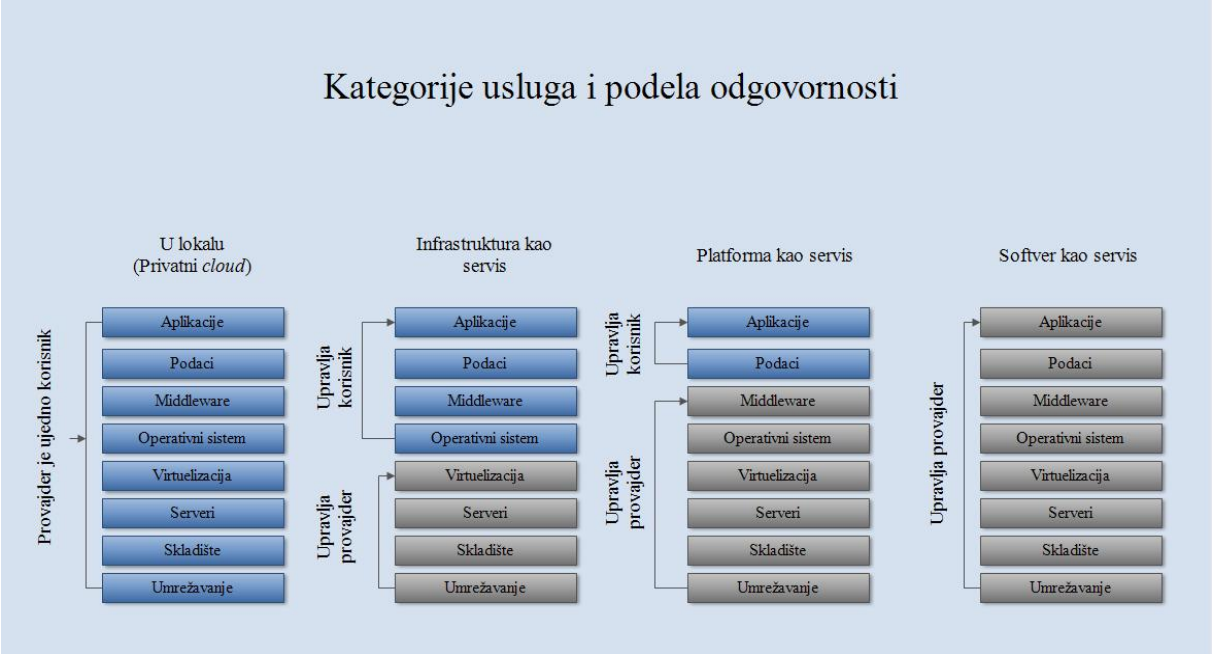
Softver u vidu servisa (SaaS) – SaaS predstavlja servis isporuke aplikacija krajnjim korisnicima putem interneta. Na ovaj način, korisnik ne mora da samostalno instalira softver i plaća licencu, već pristupa udaljenom servisu i plaća za vremenski period u kome je određeni softver koristio. Korisnici obično pristupaju softveru putem veb pregledača ili tankih klijenata. *Cloud* infrastruktura predstavlja kolekciju hardvera i softvera koja obezbeđuje navedene servise. *Cloud* infrastruktura se sastoji od fizičkog sloja i sloja apstrakcije. Fizički sloj se sastoji od hardverskih resursa koji su neophodni za obezbeđenje servisa i obično se sastoji od servera, skladišta i mrežnih komponenti. Sloj apstrakcije se sastoji od softvera koji se nalazi iznad fizičkog sloja, kojim se manifestuju najbitnije karakteristike *clouda*. Konceptualno, sloj apstrakcije se nalazi iznad fizičkog sloja. To obično predstavlja softver

koji je implementiran u obliku servisa kome se pristupa putem interneta, tj. veb pregledača ili nekog programskog interfejsa. Korisnik ne kontroliše nijedan deo infrastrukture, osim nekih ograničenih konfiguracija za pojedine aplikacije (*office* alati, socijalne mreže i dr.).

Kod IaaS-a, *cloud* provajder snabdeva korisnike skupom virtuelizovanih infrastrukturnih komponenti kao što su virtuelne mašine i skladišta pomoću kojih korisnici mogu praviti i pokretati aplikacije. Aplikacija će se nalaziti na virtuelnoj mašini i virtuelnom operativnom sistemu. Postoje i određeni problemi koji mogu uticati na narušavanje bezbednosti sistema. Takvi problemi mogu biti adekvatno obezbeđenje fizičkih servera na kojima se nalaze virtuelne mašine, zatim obezbeđenje konzistentnosti *image* fajlova virtuelnih mašina, kao i obezbeđenje komunikacije između sistema i njihovih korisnika.

PaaS omogućuje programskim okruženjima da upotrebe dodatne aplikacione gradivne blokove. Takva programska okruženja imaju vidljiv uticaj na arhitekturu samih aplikacija, kao što su ograničenja po kojima aplikacije mogu tražiti servise od strane operativnog sistema. Na primer, PaaS okruženje može ograničiti pristup ka dobro definisanim delovima fajl sistema, čime se zahteva kvalitetan autorizacioni servis.

U SaaS-u, *cloud* provajderi omogućuju i obezbeđuju aplikacije kao servis na zahtev. Iz razloga što klijenti koriste softverske komponente različitih provajdera [9], ključni problemi obuhvataju bezbedan rad i obezbeđenje informacija koje se koriste u radu ovih servisa.



Slika 2.2 - Kategorije usluga *cloud computing* servisa i podela odgovornosti u odnosu na odabrani model

Osnovni problem savremenog poslovanja kompanija u IT sektoru jesu fiksni troškovi koje stvaraju nabavka opreme, softvera i podrška za njihovo pravilno funkcionisanje. Ipak, mnogo značajniji gubitak novca ogleda se u neiskorišćenosti te opreme.

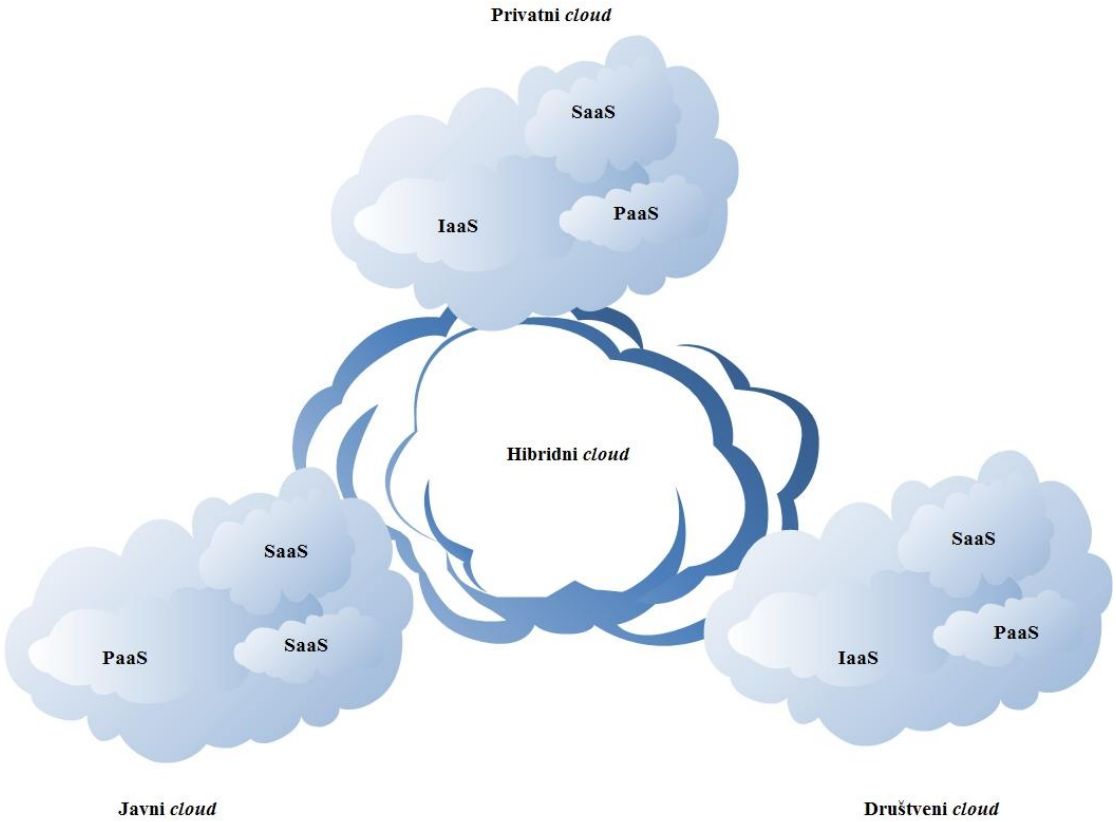
Iznajmljivanjem usluga klijent praktično dobija tri proizvoda u jednom:

- hardver kroz virtuelizaciju,
- softversku platformu, koja omogućava pokretanje aplikacija i
- samu aplikaciju.

2.2.2. Razvojni modeli usluga

Postoje četiri razvojna modela (slika 2.3) po kojima se upotrebljavaju *cloud computing* servisi. To su:

- Privatni *cloud*,
- Javni *cloud*,
- Društveni *cloud* ili *cloud* zajednice (eng. *Community cloud*) i
- Hibridni *cloud*.



Slika 2.3 - Razvojni modeli *cloud computing* platforme

Kod privatnog *cloud* sistema, infrastruktura se pravi radi isključive upotrebe unutar kompanije ili bilo koje organizacije koja ima više potrošača (npr. poslovne jedinice). Takav sistem je obično u vlasništvu organizacije koja ga i koristi.

U javnom *cloud* sistemu infrastruktura se daje na korišćenje svakom ko to želi. Vlasnik ove infrastrukture može biti neka kompanija, akademska ili državna institucija.

Infrastruktura hibridnog *cloud* sistema predstavlja skup dve ili više različitih infrastruktura (privatna, javna, pripada nekoj zajednici) koji ostaju jedinstveni entiteti, ali su povezani standardizovanim ili licenciranim tehnologijama koje omogućuju portabilnost podataka i aplikacija.

Kod *cloud* sistema namenjenih zajednici, infrastruktura se koristi isključivo za neku određenu zajednicu korisnika iz kompanija koje imaju iste ciljeve (npr. misiju, bezbednosne zahteve, politiku privatnosti i sl.).

2.3. Karakteristike

Prepoznavanje tačnih karakteristika predstavlja važan korak u uspostavljanju sistemskih zahteva i daljeg razvoja. Osnovne karakteristike koje *cloud* sistemi poseduju su:

1. Samoposluživanje na zahtev,
2. Mrežni pristup,
3. Udruživanje resursa,
4. Elastičnost,
5. Kvalitet usluga (QoS),
6. Dostupnost servisa i
7. Mereni servis.

Samoposluživanje na zahtev. Korisnik može jednostrano iskoristiti računarske mogućnosti, kao što je upotreba udaljenog servera i mrežnog skladišta, bez potrebe za direktnom interakcijom sa svakim servis provajderom.

Mrežni pristup. Sve mogućnosti su dostupne preko mreže i pristupa im se preko standardnih mehanizama koji potenciraju korišćenje putem *thin* ili *thick* klijentskih platformi (npr. mobilni telefoni, tablet računari, laptop računari i radne stanice).

Udruživanje resursa. Računarski resursi provajdera su udruženi kako bi služili višestrukim korisnicima, sa različitim fizičkim i virtuelnim resursima koji se dodeljuju dinamički prema potrebama korisnika. U resurse mogu spadati skladišta podataka, procesorska snaga, memorija i mrežni protok.

Elastičnost. Kapaciteti mogu biti elastično alocirani i oslobođeni, u većini slučaja potpuno automatski. Za korisnika, kapaciteti koji mu se dodeljuju na skalabilan način često izgledaju kao neograničeni i mogu biti prisvojeni u bilo kom vremenskom roku.

Kvalitet usluge (QoS) je od vitalnog značaja zbog zadovoljavanja korisnika i zadovoljenja pouzdanosti sistema, a postiže se nesmetanim pružanjem resursa i usluga. Kvalitet usluge se definiše ugovorima, i obično se time garantuje definisani minimum bezbednosti, brzine odaziva i propusnog opsega. Pouzdanost predstavlja jedan od aspekata kvaliteta usluga.

Dostupnost servisa predstavlja sposobnost obezbeđenja redundantnih servisa i podataka kako bi se eventualne greške sakrile na transparentan način. Mogućnost tolerancije grešaka zahteva ovakav pristup kako bi se zamenila čvorišta koja su u međuvremenu postala neaktivna ili su se pokvarila, a sve to bez gubitka na performansama. Dostupnost se dakle ispunjava replikacijom servisa i podataka obezbeđenjem iz različitih izvora.

Mereni servis. *Cloud* sistemi automatski kontrolišu i optimizuju korišćenje resursa usklađivanjem sposobnosti merenja na nekom nivou apstrakcije koji je prilagođen vrsti pružene usluge (skladište podataka, procesorska snaga, protok i aktivni korisnički nalozi). Korišćenje resursa može biti praćeno, kontrolisano i prikazano u vidu izveštaja, obezbeđenjem transparentnosti i za provajdera i za korisnika usluge.

3. *CLOUD COMPUTING* PLATFORME

Danas postoji više vrsti *cloud computing* platformi. Sve one zadovoljavaju potrebe određenog kruga kompanija i/ili korisnika. U osnovi postoje dve vrste:

1. Komercijalne *cloud computing* platforme i
2. *Cloud computing* platforme otvorenog koda (*open source*).

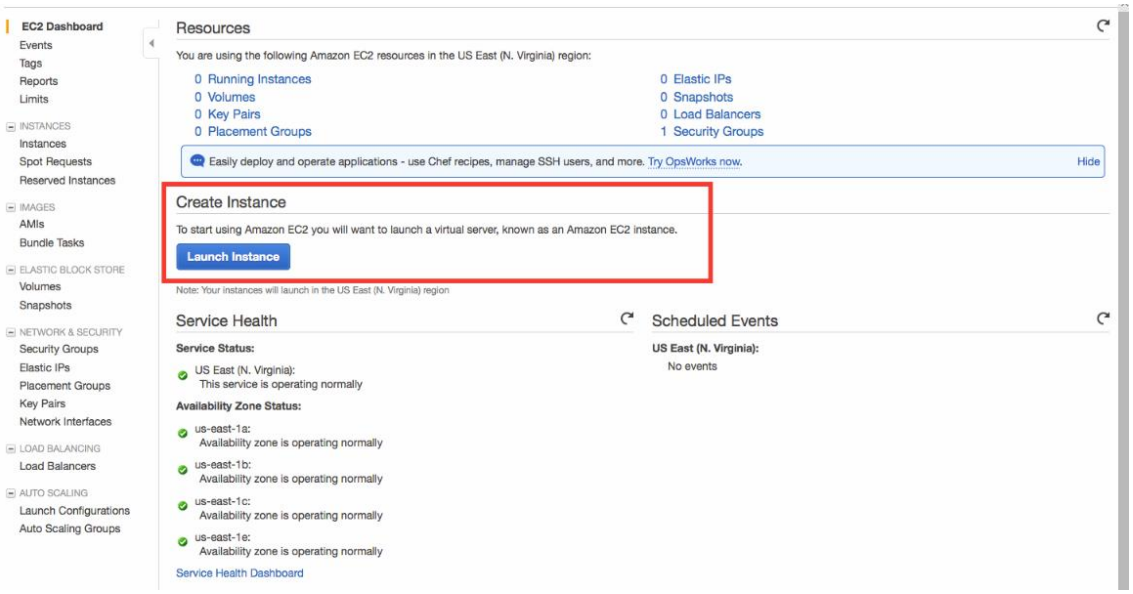
3.1. Komercijalne cloud computing platforme

Komercijalne platforme pripadaju određenim kompanijama koje samostalno razvijaju svoju *cloud* platformu za svoje potrebe i za potrebe svojih korisnika. Te platforme se mogu, ali ne moraju, oslanjati na druge platforme ili delove platformi otvorenog koda.
Primeri takvih platformi su:

1. Amazon Web Services,
2. Google App Engine,
3. Microsoft Azure i
4. VMware vCloud.

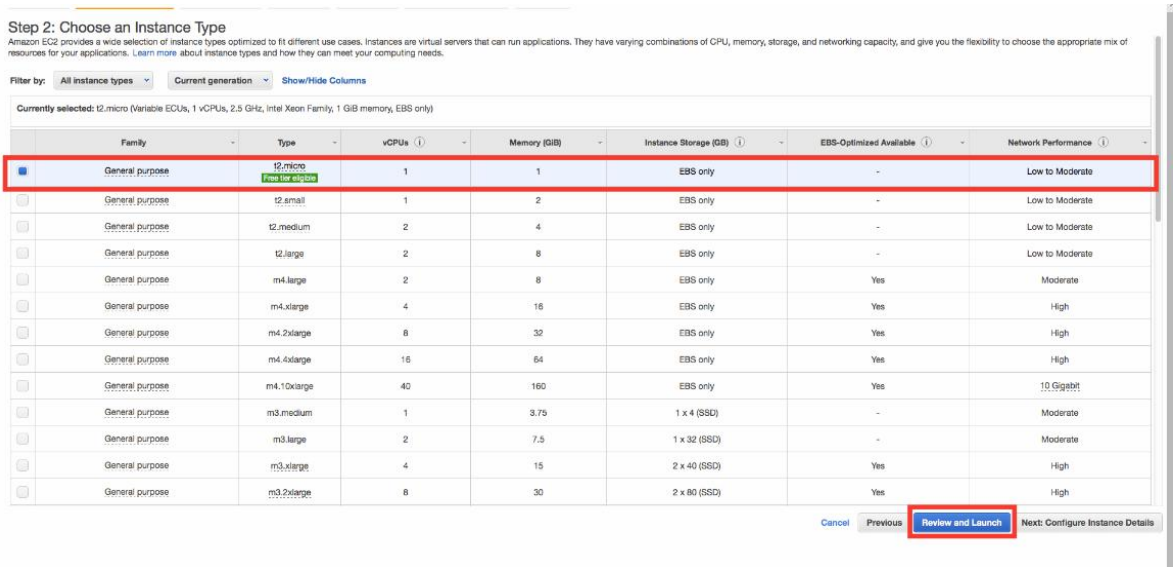
3.1.1. Amazon Web Services

Amazon Web Services (AWS) predstavlja sastavni deo kompanije Amazon [10]. To je servis zasnovan na *cloudu* sa ciljem obezbeđenja visoko automatizovanog i cenovno efikasnog servisa. Osnovna prednost AWS-a je sposobnost fleksibilne isporuke svih merenih servisa na zahtev. AWS nudi IaaS servis uz virtuelizaciju zasnovanu na Xen platformi, sa višestrukim načinom skladištenja podataka (Amazon S3 na različitim lokacijama) i Docker kontejnerskim servisom (Amazon EC2). AWS ima veoma široku paletu softvera i usluga drugih kompanija [11].
Amazon Elastic Compute Cloud (*Amazon EC2*) je centralni deo AWS-a i predstavlja virtuelno računarsko okruženje koje omogućuje korisnicima da koriste veb interfejs kako bi pokretali svoje instance (virtuelne mašine) sa različitim operativnim sistemima, aplikacijama po izboru, itd. Pored toga, Amazon EC2 omogućuje upravljanje dozvolama za pristup, uz potpunu kontrolu sistema i softvera koji će se pokretati na svakoj instanci. Izgled interfejsa aplikacije dat je na slici 3.1.



Slika 3.1 - Izgled interfejsa Amazon EC2

Amazon Machine Image (AMI) je primarna konfiguraciona datoteka koja sadrži konfiguraciju softvera (na primer operativnih sistema, serverskih aplikacija i drugih aplikacija). Od AMI-ja se pokreću instance koje prave kopije. Moguće je pokrenuti više instanci različitih karakteristika, kao što je prikazano na slici 3.2. Ove instance će raditi sve dok ih korisnik ne prekine ili dok se ne pojavi greška, pa neka od njih prestane da radi. Ukoliko se ovako nešto desi, korisnik uvek može da pokrene novu instancu. Moguće je korišćene jednog ili više AMI-ja, u zavisnosti od potreba korisnika usluga, odnosno od zakupljenog prostora, snage za obradu podataka koja je potrebna i sl.

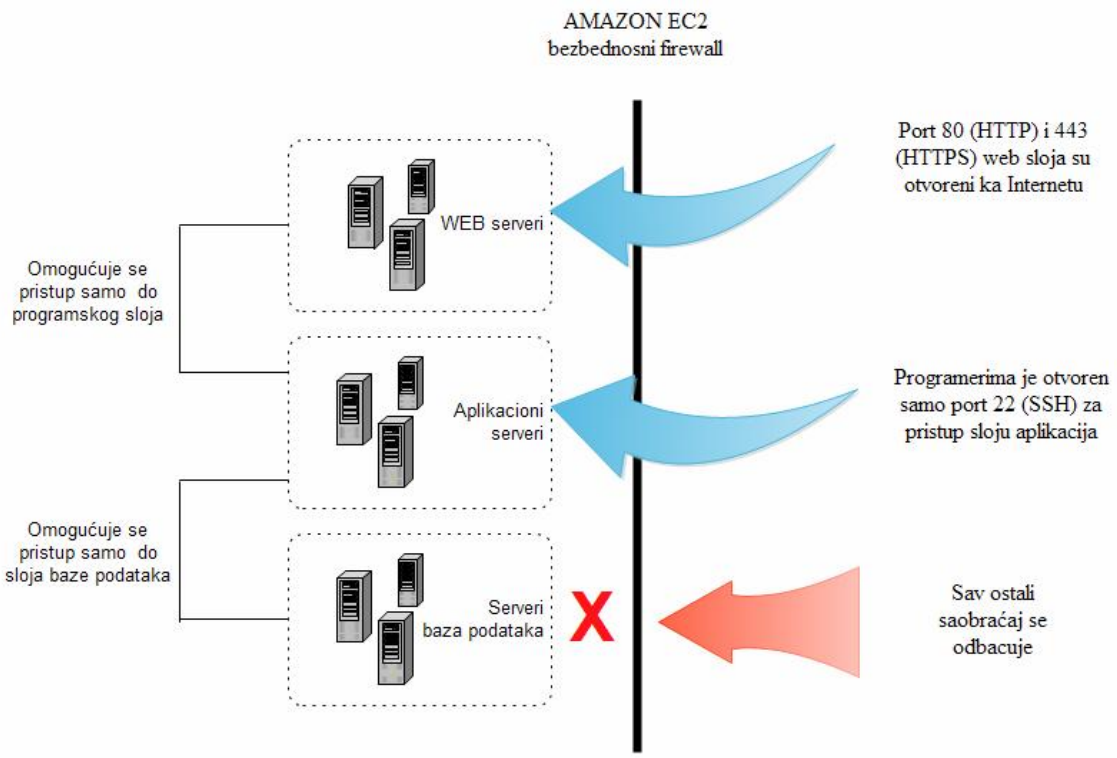


Slika 3.2 - Biranje predefinisanih instanci pomoću *Amazon Machine Image* (AMI)

Amazonov EC2 se vodi principima elastičnosti, tako da omogućuje promenu obima i broja instanci u svakom trenutku, na zahtev korisnika. Svaka instanca se pokreće u Amazon EC2 mrežnom prostoru i dodeljuje joj se javna IP adresa. Instance mogu da se prekinu iz različitih razloga koji su van kontrole korisnika. Ukoliko dođe do prekida, klijent pokreće novu instancu. Ovaj postupak dovodi do dodele nove javne IP adrese. Amazon EC2 servis omogućava i dodelu statičkih IP adresa u slučaju da je korisniku neophodna takva usluga.

Za kontrolu pristupa instancama koriste se ključevi i bezbednosne grupe. Prilikom kreiranja instance, korisnik preuzima svoj ključ na svoj računar. Nakon toga, svaki put kada korisnik želi da se prijavi na udaljenu instancu, on mora posedovati dodeljeni ključ i lozinku za pristup instanci. Bezbednosne grupe su analogne ulazima kod mrežnih zaštitnih zidova koji omogućavaju da klijent odredi protokole, portove i opsege IP adresa kojima je dozvoljeno da pristupe instanci. Moguće je kreirati više grupa bezbednosti i dodeliti različita pravila svakoj od njih. Zatim, moguće je da svaka instanca ima jednu ili više bezbednosnih grupa, a Amazon EC2 koristi sva ta pravila za određivanje saobraćaja i kome je dozvoljeno da pristupi kom delu podataka. Takođe, moguće je konfigurisati sistem i sigurnosne grupe tako da samo određena IP adresa ili određena sigurnosna grupa ima pristup nekoj instanci.

Na slici 3.3 prikazana je arhitektura osnovna tri nivoa veb hostinga po kome radi Amazon EC2. Svaki sloj ima različite bezbednosne grupe (označeno isprekidanom linijom oko svakog skupa instanci). Bezbednosna grupa za veb servere dozvoljava pristup samo sa hosta preko TCP-ja na portovima 80 i 443 (HTTP i HTTPS) i od instanci u bezbednosnoj grupi aplikacionih servera na portu 22 (SSH) za direktno upravljanje hostom.



Slika 3.3 - Arhitektura osnovna tri nivoa veb hostinga Amazon EC2 [12]

Grupa za bezbednost aplikacija odobrava zahteve za pristupanje veb sigurnosnoj grupi preko TCP-ja na portu 22 (SSH) za direktno upravljanje hostom. Inženjeri za podršku u samoj klijentskoj organizaciji mogu da se loguju direktno na aplikacione servere sa korporativne mreže, a zatim i da pristupe drugim instancama u delu aplikacionog servera.

Sigurnosna grupa za servere baza podataka dozvoljava pristup bazama podataka samo sigurnosnoj grupi aplikacionih servera.

Sav ostali saobraćaj je odbijen i ne može da pristupi serveru.

3.1.2. Google Cloud

Google *cloud* platforma kombinuje IaaS servise (*Compute Engine*) sa PaaS servisima (*App Engine*). Google *cloud* omogućuje čuvanje objekata i *Docker* kontejnerski servis (Co*ntainer Engine*). Virtuelne mašine kod IaaS servisa se pokreću pomoću KVM platforme i potrošnja resursa se meri u minutima. Takođe poseduje višedomenski način tolerancije grešaka, što je predviđeno SLA ugovorom.

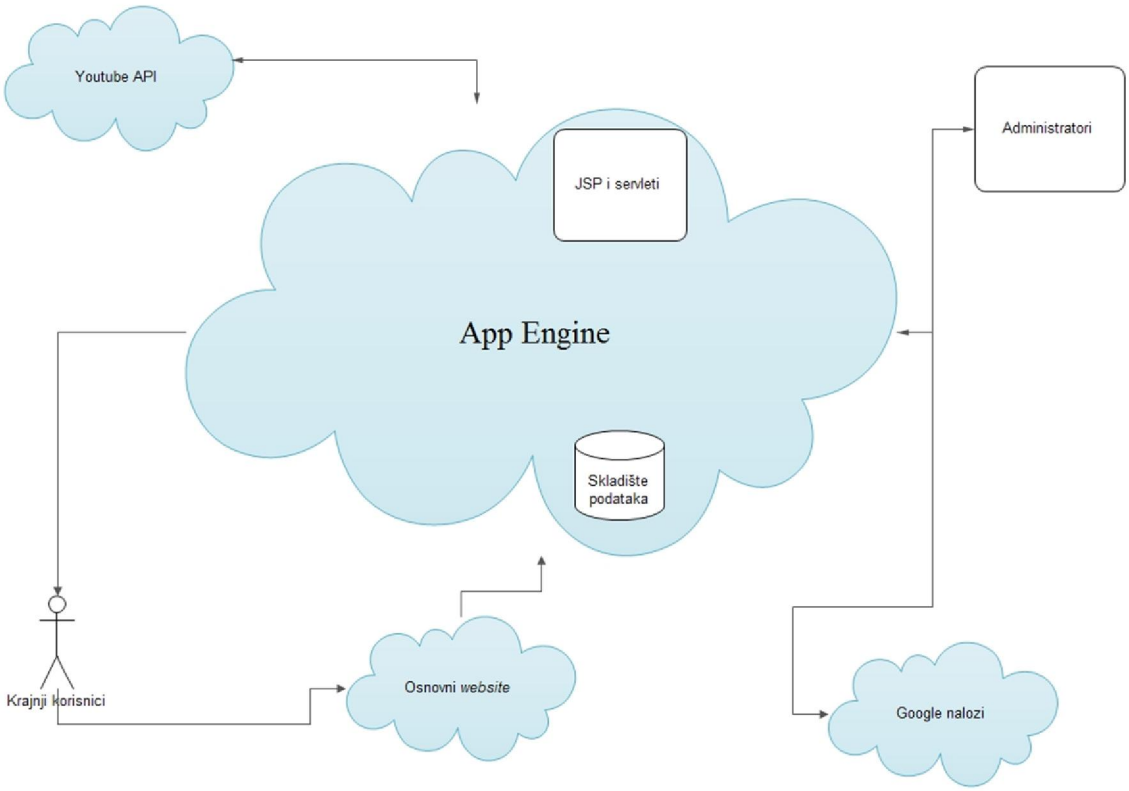
Google grupiše svoje data centre po regionima, uz dodatnu podelu po zonama dostupnosti, tako da korisnik može imati određene podatke koji se nalaze u data centru na jednom kontinentu, a rezervnu kopiju u data centru na drugom kontinentu. Na taj način se značajno

unapređuje zaštita od gubitka podataka u slučaju nepredviđenih okolnosti i prirodnih katastrofa.

Google App Engine [13] je platforma za pravljenje skalabilnih veb i mobilnih aplikacija. U okviru servisa postoje već predefinisane opcije za pravljenje NOSQL baza podataka, programski interfejsi za autentifikaciju korisnika, keširanje aplikacija u memoriji (*memcache*) radi povećanja performansi i dr. App Engine je vrlo skalabilna platforma i omogućuje automatsko skaliranje u odnosu na trenutni saobraćaj i opterećenje servera. Na taj način korisnici plaćaju samo ono što su i potrošili.

App Engine podržava programske jezike kao što su Java, Python, PHP ili Go i alate za razvoj aplikacija kao što su Eclipse, IntelliJ, Maven, Git i dr.

Google je u okviru Google App Engine uklonio mogućnost upisa u fajl sistem, jer se na taj način izbegava niz bezbednosnih pretnji. Aplikacijama je omogućeno da koriste samo virtuelne fajl sisteme. Za skladištenje podataka se mora koristiti Google baza podataka. Na taj način, Google je napravio okvir za pravljenje aplikacija.



Slika 3.4 - Dijagram entiteta Google App Engine platforme

Jedna od ključnih prednosti je skoro neograničena veličina aplikacije koja se može postaviti u *cloud*. Sledeća prednost je Python. Ovaj programski jezik je postao vrlo popularan za sve vrste aplikacija, a naročito za veb aplikacije. App Engine podržava razne alate za razvoj veb i mobilnih aplikacija. Postoji veliki broj podržanih veb šablona: Django, CherryPy, Pyramid, Flask, webapp2. Prednosti su mu velika pouzdanost i vrlo dobra dokumentacija.

U poređenju sa drugim skalabilnim servisima kao što je Amazon EC2, App Engine obezbeđuje bolju infrastrukturu za pisanje skalabilnih aplikacija, ali može raditi sa ograničenim brojem aplikacija koje su predviđene za rad sa takvom infrastrukturom.

Na slici 3.4 je prikazana pojednostavljena arhitektura Google App Engine platforme.

3.1.3. Microsoft Azure

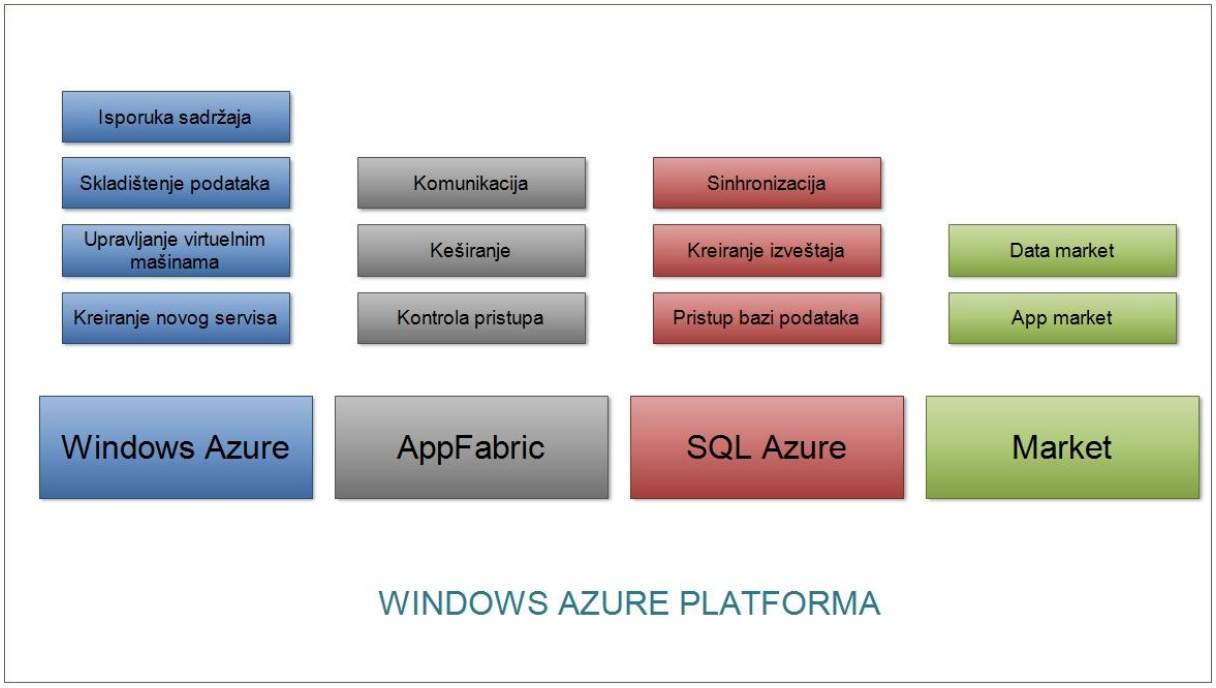
Microsoft je jedna od najvećih IT kompanija na svetu koja se sve više fokusira na isporuku svojih softverskih paketa putem *cloud* servisa. Microsoft Azure [14] *cloud* platforma prvobitno je bila isključivo PaaS platforma, ali je kasnije proširena Azure Infrastructure servisom.

Microsoft Azure platforma (slika 3.5) je *cloud* platforma koja obezbeđuje pravljenje, postavljanje i upravljanje internet servisima na privatnoj mreži decentralizovanih data centara. Platforma obezbeđuje PaaS i IaaS vrste usluga, uz podršku za mnoge programske jezike i alate.

Microsoft Azure je specijalizovani operativni sistem koji upravlja resursima u klasteru računara. On predstavlja tzv. „*cloud*“ sloj koji se nalazi iznad Windows Server 2008 sistema sa izmenjenom verzijom Hyper-V hipervizora, kako bi obezbedio virtuelizaciju servisa. Servisu se pristupa preko posebnog API interfejsa, kao i pomoću veb-interfejsa – Azure portala [15]. Ovaj portal omogućava korisnicima da pregledaju aktivne resurse, izvrše izmene u podešavanjima, pokrenu nove virtuelne mašine i druge resurse i imaju uvid u osnovne podatke sa aktivnih virtuelnih mašina i servisa.

Microsoft obezbeđuje razne vrste servisa. *Cloud* servisi predstavlja PaaS okruženje koje se može koristiti za kreiranje skalabilnih aplikacija i servisa. Ova platforma obezbeđuje kontejnere sa hostovanim aplikacijama. Aplikacije mogu biti razne vrste veb aplikacija (npr. aplikacije za elektronsku trgovinu, veb-sajtove i sl.) ili mogu obezbediti okruženje za privatnu obradu i analizu podataka. Programeri mogu samostalno pisati aplikacije za *cloud* servise u različitim programskim jezicima. Postoje određeni SDK alati za Python, Javu, node.js i .NET.

Uz pomoć IaaS usluge, Azure omogućuje migriranje aplikacija i infrastrukture bez obzira na vrstu virtuelnih mašina.



Slika 3.5 – Windows Azure platforma [16]

3.1.4. VMware vCloud

VMware je prvobitno bila softverska kompanija koja se bavila tehnologijom virtuelizacije i razvijala aplikacije za virtuelizaciju. Tek kasnije dolazi do razvoja sopstvene IaaS platforme – VMware vCloud Hybrid Service (vCHS), tj. vCloud Air.

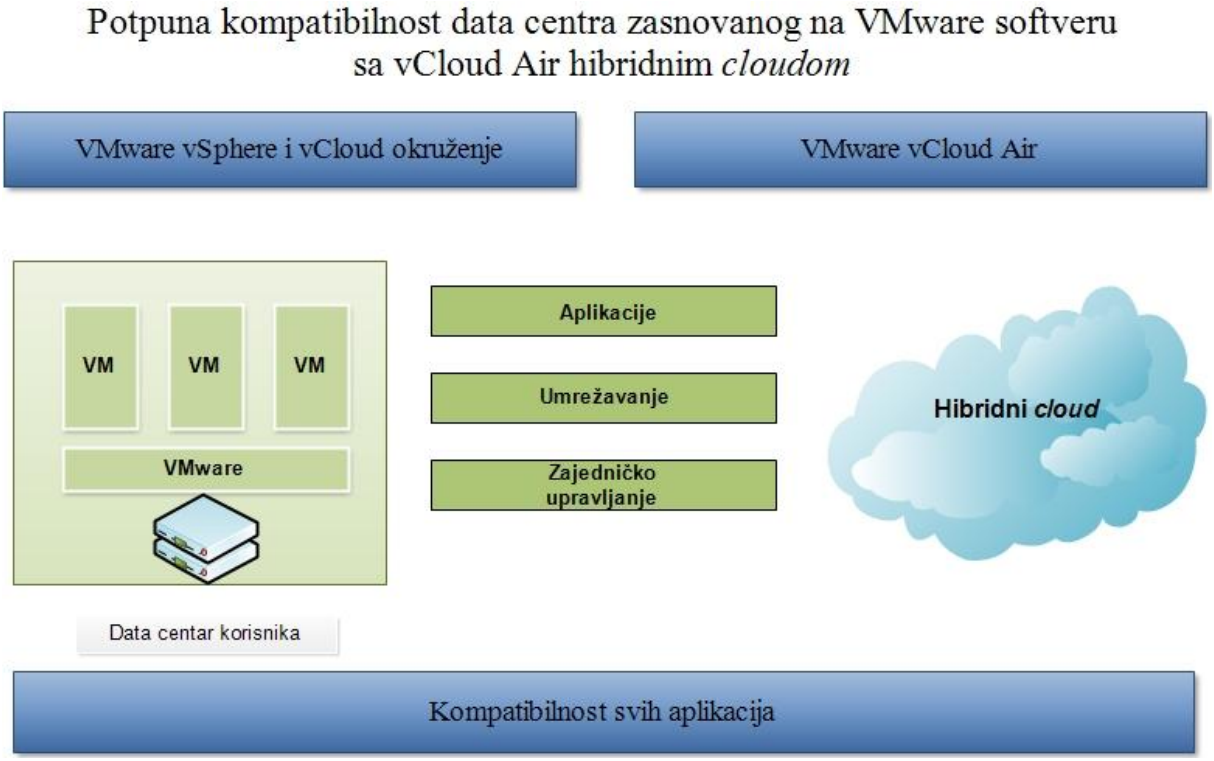
vCloud Air [17] vrši virtuelizaciju na sopstvenoj VMware platformi i podržava različite tipove virtuelizacije koje iznajmljuje korisnicima. Svi vCloud servisi se nalaze na istom portalu i isporučuju se kao skup resursa sa istog deljenog hardvera. Glavne prednosti vCloud okruženja su razvojna okruženja, hibridna VMware *cloud* okruženja, poslovne aplikacije i poseban sistem oporavka u slučaju nastalih nezgoda.

vCloud ima sledeće karakteristike:

- Servisne ponude po principu "Plati-po-usluzi", sa svakom virtuelnom mašinom naplaćenom posebno i različitim računima za plaćanje za svaku virtuelnu mašinu.
- Sigurnosne pristupe radnim okruženjima kao i mrežnoj izolaciji, dozvoljavajući da više organizacija dele resurse jednog vClouda.
- Portal gde IaaS platforma može biti iznajmljena iz kataloga predefinisanih servisa. Iznajmljene platforme mogu biti namenjene oporavku od katastrofa, za virtuelni privatni *cloud* ili posvećeni (dedicated) *cloud*.
- Korišćenje javnog vClouda zbog ubrzanog korišćenja aplikacija.

Ključne stvari vCloud arhitekture su združivanje resursa, apstrakcija i izolacija. VMware vCloud Director dalje apstrahuje virtuelne resurse prikazane od strane vSphere pružajući sledeće logičke konstrukcije vSphere logičkih resursa:

- Organizacija – Logički objekat kojim se postavljaju pravila u vidu polisa i sigurnosnih ograničenja. Organizacije su glavni metod stvaranja rezultata i obično prikazuju poslovnu jedinicu, projekat ili korisnika privatnog vCloud okruženja.
- Virtuelni datacentar – Okruženje u kome virtuelne mašine rade.
- Organizacioni virtuelni data centar – Organizacioni deo virtuelnog data centra koji pripada provajderu a uključuje CPU, memoriju i skladište podataka.
- Virtuelni data centar provajdera – vSphere resursno grupisanje računara, skladišta i mreža koji pokreću organizacione virtuelne data centre.



Slika 3.6 – VMware hibridni vCloud

3.2. Cloud platforme otvorenog koda

Cloud platforme otvorenog koda predstavljaju sve vrste *cloud* servisa koje su dostupne pod određenim licencama koje dozvoljavaju besplatnu upotrebu, uz mogućnost sopstvenog doprinosa u daljem razvoju. To može biti bilo koji model koji obezbeđuje IaaS, PaaS ili SaaS usluge.

Neke od platformi otvorenog koda su:

1. Eucalyptus,
2. OpenNebula,
3. Nimbus i
4. OpenStack.

3.2.1. Eucalyptus

Eucalyptus [18] je softverska platforma za realizaciju privatnih ili hibridnih *cloud* sistema na klasteru računara. Postoji otvoreno jezgro *Enterprise Edition* i *open source* izdanje. Trenutno, platforma funkcioniše tako što izvozi korisnički interfejs koji je kompatibilan sa Amazon EC2 i S3 servisima, ali platforma je modularizovana tako da može istovremeno da podržava niz različitih interfejsa. Razvoj softvera Eucalyptus je sponzorisan od strane Eucalyptus Systems kompanije. Eucalyptus radi sa većinom trenutno dostupnix Linux distribucija uključujući Ubuntu, Red Hat Enterprise Linux (RHEL), CentOS, SUSE Linux Entreprise Server (SLES) , OpenSUSE, Debian i Fedora. Sistem podržava i pravljenje Microsoft Windows virtuelnih mašina. Na sličan način Eucalyptus može da koristi različite tehnologije virtuelizacije uključujući WMware, Xen i KVM hipervizore za implementaciju instanci virtuelnih mašina. Eucalyptus je skraćenica od "***E**lastic **U**tility **C**omputing **A**rchitecture for **L**inking **Y**our **P**rograms **T**o **U**seful **S**ystems*".

Platforma obezbeđuje jedan interfejs koji omogućuje korisnicima da pristupe računarskim resursima koji su na raspolaganju u privatnom *cloudu* (implementirani od strane Eucalyptusa unutar postojećeg centra podataka organizacije) i resurse koji su dostupni eksterno u javnim *cloud* servisima. Softver je dizajniran sa modularnom i proširivom arhitekturom zasnovanom na veb uslugama koja omogućuje izvoz raznih programskih interfejsa prema korisnicima preko klijentskih alata. Eucalyptus je implementirao industrijski standard *Amazon Web Services* (AWS) API, koji omogućuje interoperabilnost postojećeg Eucalyptusa sa postojećim AWS servisima i alatima. Eucalyptus obezbeđuje sopstveni skup alata na komandnoj liniji pod nazivom Euca2ools, koji mogu biti interno iskorišćeni za interakciju sa Eucalyptus instalacijama na privatnom *cloudu* ili eksterno za komunikaciju sa javnim *cloud* servisama, uključujući Amazon EC2.

Eucalyptus ima sledeće karakteristike:

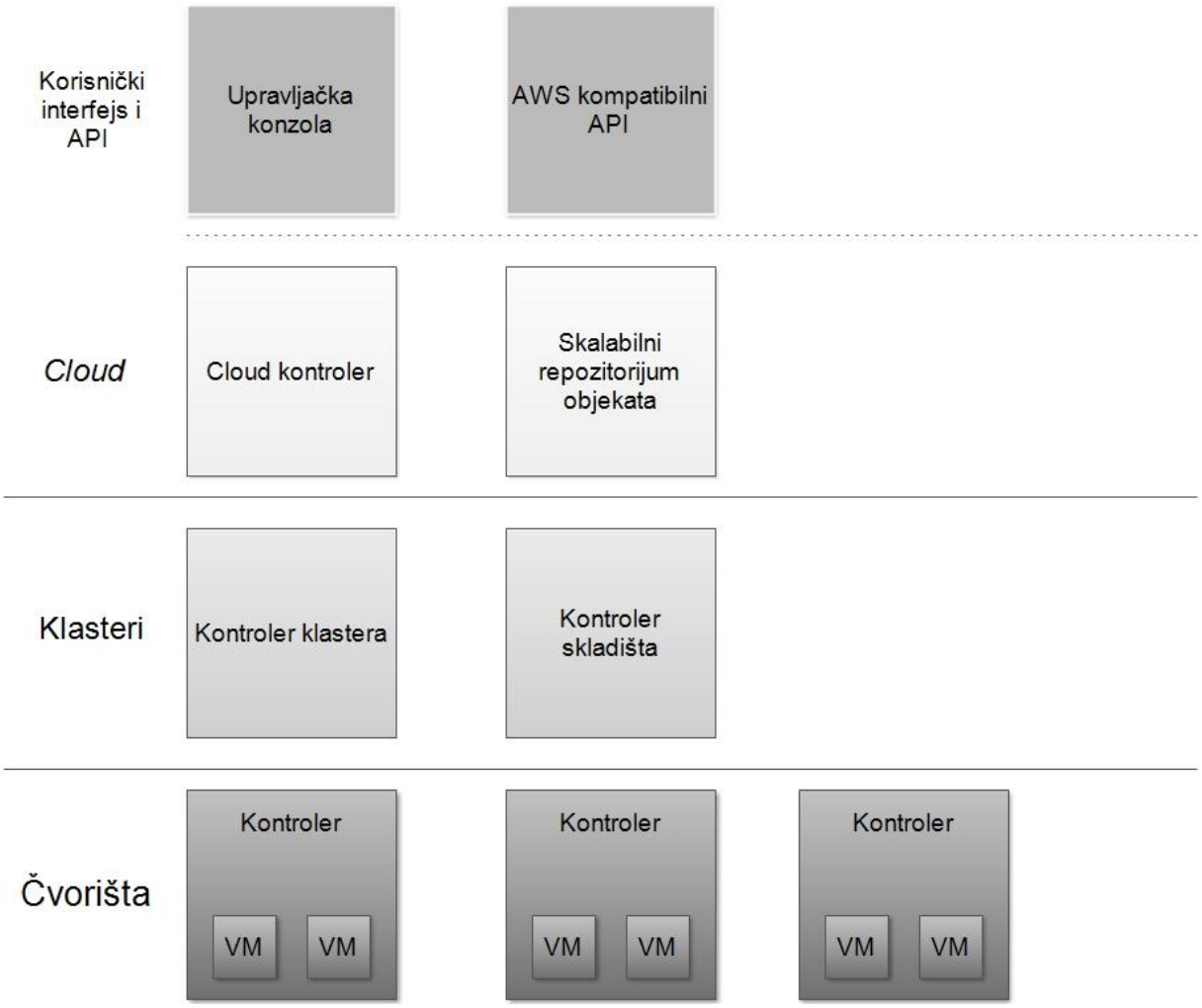
- Kompatibilnost sa Amazon Web Services API,
- Instalira se i briše od izvora ili DEB i RPM paketa,
- Sigurna komunikacija između internih procesa putem SOAP i WS-Security servisa,

- Podrška za Linux i Windows virtuelne mašine,
- Podrška za višestruke klastere kao jedan *cloud*,
- Elastične IP adrese i sigurnosne grupe,
- Upravljanje korisnicima i grupama,
- Service Level Agreement (SLA) i prilagodljiva politika raspoređivanja.

Eucalyptus *cloud computing* platforma ima šest komponenti:

1. *Cloud* kontroler (*Cloud Controller* - CLC),
2. Kontroler klastera (*Cluster Controller* - CC),
3. *Walrus*,
4. Kontroler skladišta (*Storage Controller* - SC),
5. Kontroler čvorišta (*Node Controller* - NC) i
6. *Vmware Broker (opciona komponenta).*

Svaka od ovih komponenti sistema ima svoj veb interfejs i implementirana je kao samostalni veb servis. Ovo ima dve prednosti: prvo, svaki veb servis otkriva dobro definisan jezičko-agnostički API u obliku WSDL dokumenta koji sadrži i operacije koje servis podržava i ulazno/izlazne strukture podataka. Drugo, Eucalyptus ispituje postojeće karakteristike veb servisa kao što su bezbednosne polise (WSS) u cilju bezbedne komunikacije između komponenti i oslanja se na softverske pakete za veb usluge zasnovane na industrijskim standardima. Na slici 3.7 dat je šematski prikaz komponenti Eucalyptus platforme.



Slika 3.7 - Komponente Eucalyptus platforme [19]

Cloud kontroler je odgovoran za izlaganje i upravljanje virtuelizovanim resursima preko korisničkih API-ja. Trenutno, CLC izvozi dobro definisani industrijski standard API (Amazon EC2) preko veb-baziranog korisničkog interfejsa.

Trenutna implementacija *Walrusa* je interfejs kompatibilan sa Amazonovim S3, pružajući mehanizam za stalno skladištenje i kontrolu pristupa virtuelnim mašinama i korisničkim podacima.

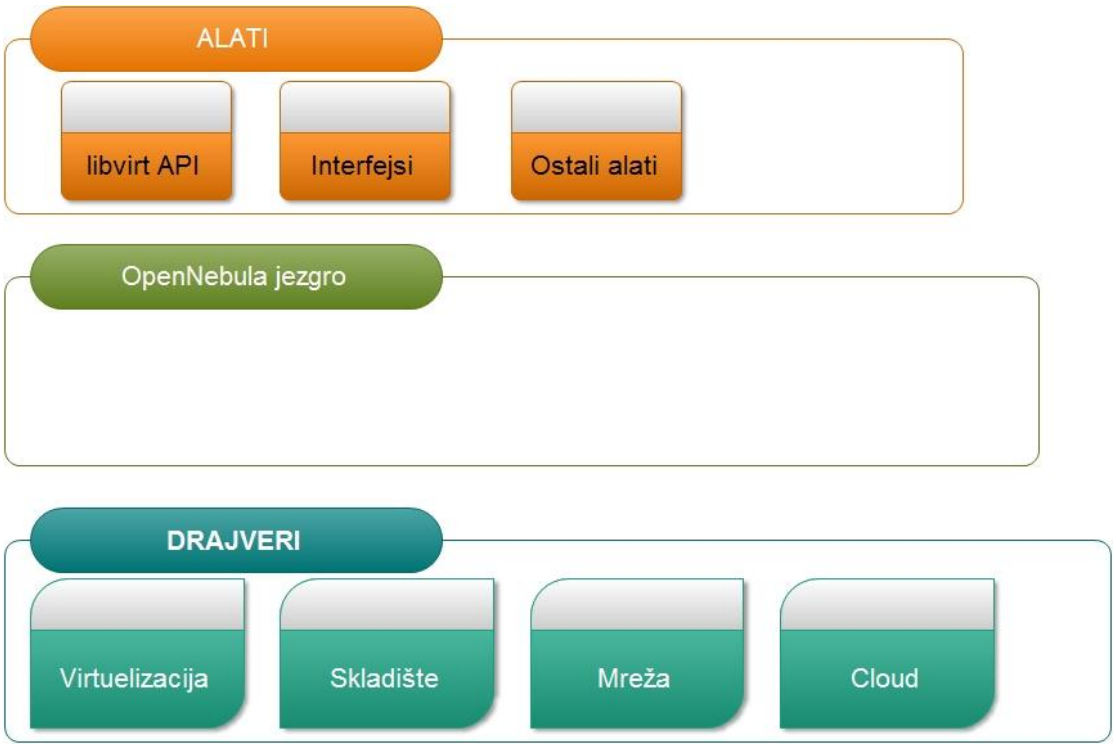
Kontroler klastera kontroliše izvršavanje virtuelnih mašina koje rade na čvorovima i rukovodi virtuelnim umrežavanjem između virtuelnih mašina i virtuelnih mašina eksternih korisnika.

Kontroler skladišta obezbeđuje zaštitu na nivou mrežnog skladištenja koje može dinamički da bude povezano sa virtuelnim mašinama. Trenutna implementacija podržava *Amazon Elastic Block Storage* semantiku.

Kontroler čvorišta, kroz funkcionalnost hipervizora, kontroliše aktivnosti virtuelnih mašina, uključujući izvršenje, inspekciju i prestanak rada instanci virtuelnih mašina.

3.2.2. OpenNebula

OpenNebula [20] je projekat pokrenut 2005. godine kao konfiguracioni alat za upravljanje virtuelnim mašinama u data centru [21]. OpenNebula je sada projekat otvorenog koda i može se koristiti kao platforma za izgradnju privatnog, javnog ili hibridnog *cloud* sistema. Ključni aspekt ove platforme jeste njegova arhitektura koja obezbeđuje visok stepen centralizacije. Arhitektura takođe podržava više tipova skladištenja, kao i različite hipervizore, kao što su Xen, Hyper-V, VMWare, OpenVZ i KVM.



Slika 3.8 - Arhitektura OpenNebula platforme

Arhitektura OpenNebula platforme je podeljena na tri sloja:

1. Drajveri,
2. Jezgro,
3. Alati.

Prvi sloj sadrži drajvere. Oni predstavljaju module koji obezbeđuju sloj apstrakcije nad operacijama nižeg nivoa, kao što je mehanizam transfera fajlova, hipervizor i različiti servisi.

Jezgro OpenNebula platforme predstavlja drugi sloj i izvršava upravljanje i konfiguraciju drugih komponenti. Njega čini i skup komponenata koje se koriste za kontrolisanje i praćenje virtuelnih mašina, virtuelnih mreža, hostova i skladišta.

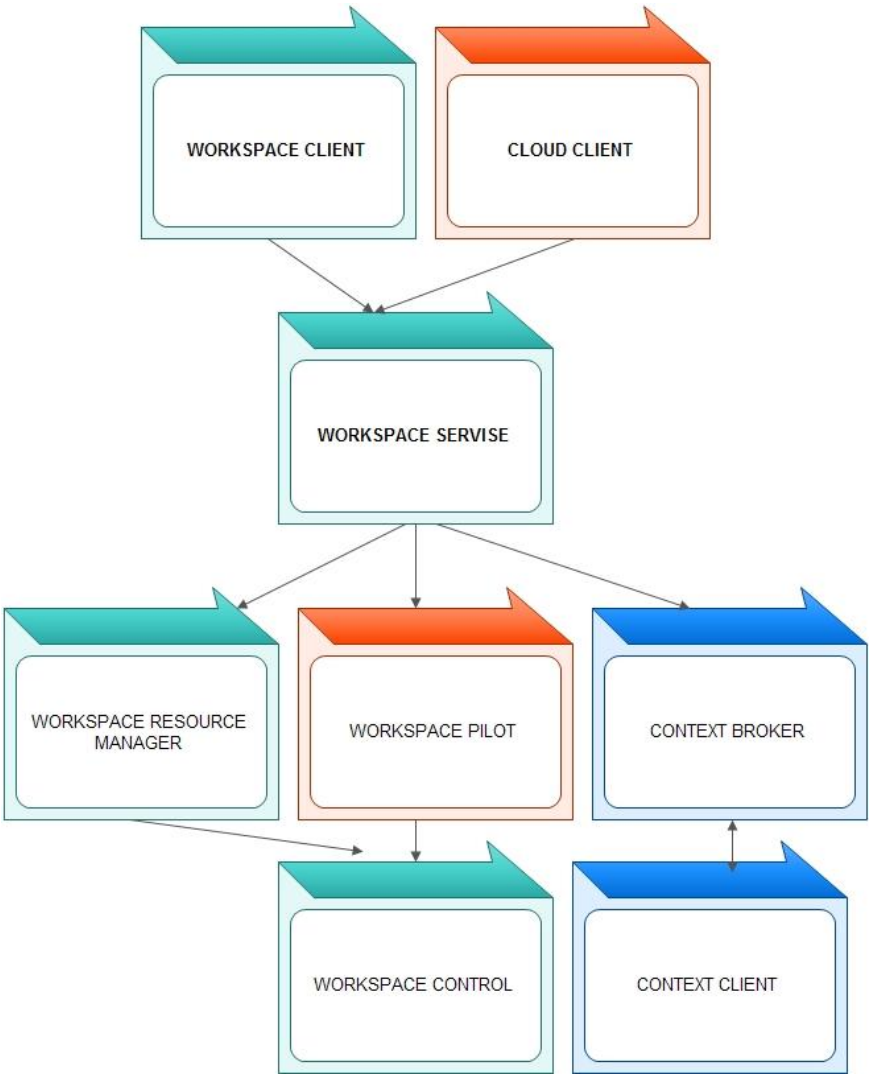
Treći sloj se sastoji iz alata za upravljanje u vidu različitih interfejsa (CLI i API interfejsi) u okviru OpenNebula platforme ili druge alate koji mogu biti kreirani upotrebom XML-RPC interfejsa.

OpenNebula je podesiv i prilagodljiv, pa je popularan kod istraživača koji žele da ispituju mogućnosti i nedostatke *cloud* platformi ili kombinuju *cloud* sisteme sa drugim tehnologijama.

3.2.3. Nimbus

Nimbus [22] je projekat otvorenog koda koji omogućava klijentima da iznajmljuju resurse postavljanjem virtuelnih mašina i obezbeđivanjem okruženja koje je korisniku potrebno. Nimbus se može prilagođavati, ali je ipak manje prilagodljiv nego OpenNebula. Npr. neke komponente kao što je skladište kopija virtuelnih mašina i akreditivi za autentifikaciju korisnika se ne mogu uklanjati. Nimbus podržava implementacionu šemu za *Amazon EC2* i druge veb servise koji mogu olakšati korišćenje ove platforme. Nimbus takođe podržava i AWS S3 za skladištenje podataka.

Uz pomoć alata olakšano je upravljanje servisima i skladištenje kopija virtuelnih mašina.



Slika 3.9 - Prikaz Nimbus platforme

Workspace service služi za upravljanje virtuelnim mašinama. Servis za skladištenje podataka je deo servisa i može biti posebno instaliran. *Workspace resource manager* upravlja čvorištima. *Workspace pilot* omogućuje integraciju predodređenih resursa za virtuelne mašine. *Workspace control* je zadužen za upravljanje i kontrolu instanci, diskova, integraciju instanci u mrežu i dodeljivanje MAC i IP adresa. *Workspace client* omogućuje pristup svim komandama u komandnoj liniji. Cilj *cloud* klijenta je da omogući proces pokretanja virtuelnih mašina upotrebom pokretača instanci. Klijenti mogu automatski pokrenuti virtuelni klaster upotrebom brokera.

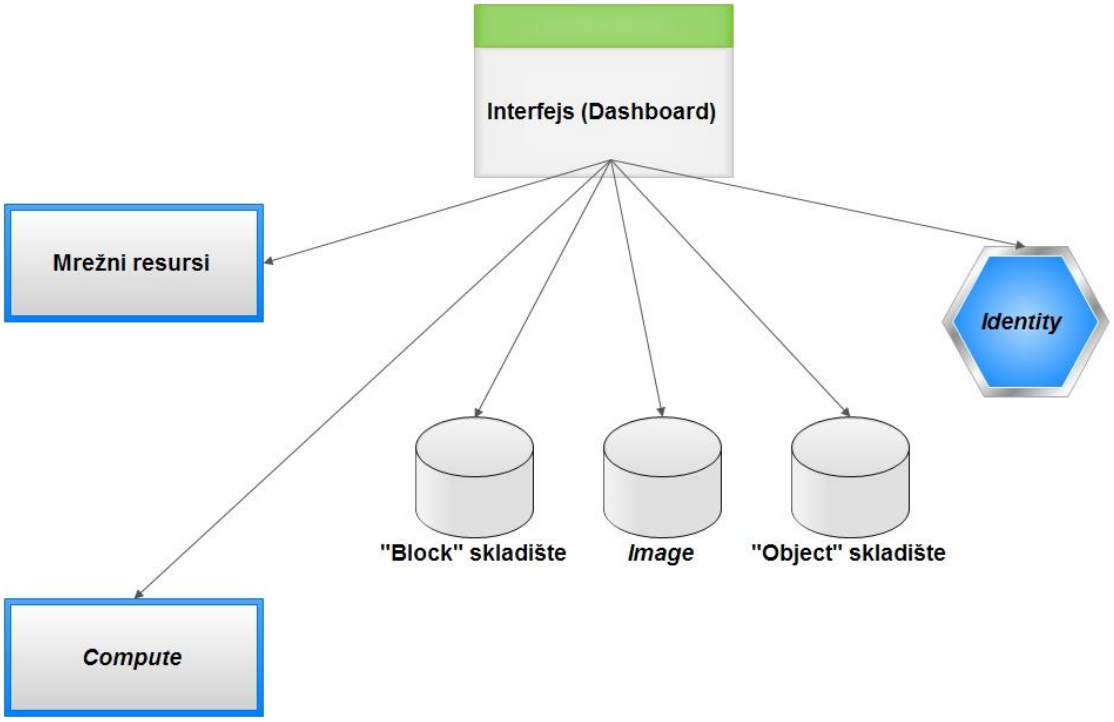
Nimbus podržava rad sa hipervizorima kao što su Xen i KVM.

3.2.4. OpenStack

OpenStack [23] je jedan od *cloud computing* projekata otvorenog koda, pre svega namenjen IaaS platformama. Razvijen je od strane američke agencije NASA i predviđen je za rad u

velikim data centrima. OpenStack spada u visoko skalabilna rešenja. Podržava implementaciju sa preko milion fizičkih hostova i preko 60 miliona virtuelnih mašina [24]. Zbog svoje otvorenosti, celokupni izvorni kod može biti pregledan i modifikovan po potrebi. OpenStack je skup više projekata (slika 3.10):

- OpenStack Compute (Nova), za obezbeđenje i upravljanje mrežom virtuelnih mašina,
- OpenStack Object Storage (code-name Swift),
- OpenStack Image Service (code-name Glance),
- OpenStack Identity (code-name Keystone),
- OpenStack Dashboard (code-name Horizon),
- OpenStack Networking (code-name Quantum),
- OpenStack Block Storage (code-name Cinder).



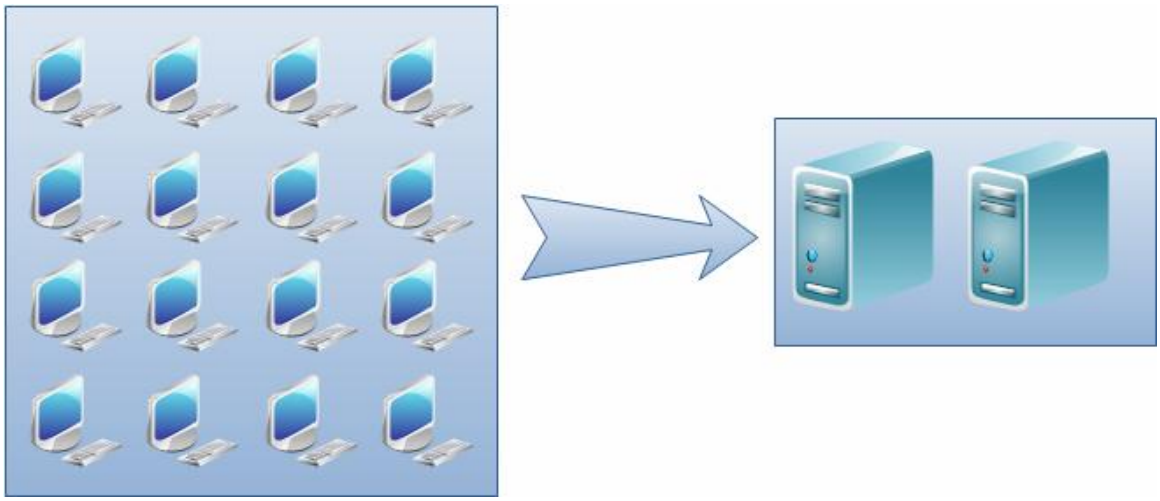
Slika 3.10 - Konceptualni prikaz OpenStack platforme

Kao i prethodno opisani sistem, OpenStack podržava EC2, S3 i ostale interfejse. Postoje mnoge komponente koje su vrlo slične kao i kod Eucalyptus platforme, kao što je dodela i prosleđivanje IP adresa, autentifikacija i hipervizori. OpenStack je prihvaćen od strane mnogih poznatih organizacija i kompanija i nalazi se u konstantnom razvoju [25].

3.3. Virtuelizacija

Virtuelizaciju [26] možemo definisati kao način da se na jednom računarskom sistemu izvršava konkurentno više operativnih sistema, što znači da se logički razdvaja računarski

hardver i operativni sistem koji se na njemu izvršava. Instancu svakog od ovih operativnih sistema nazivamo gostujući operativni sistem (guest OS). Upravljanje svakom instancom gosta obavlja *Virtual Machine Monitor* (VMM) koji se drugačije naziva i hipervizor (*hypervisor*) [27]. Hipervizor je zadužen za upravljanje računarskim resursima (procesor, memorija, skladište podataka) i dodeljivanje istih svakoj instanci gostujućeg operativnog sistema. Takođe, VMM je zadužen za eventualno nesmetano migriranje gostujućih operativnih sistema sa jednog fizičkog računarskog sistema na drugi. Na taj način možemo redukovati IT infrastrukturu modelom prikazanim na slici 3.11.



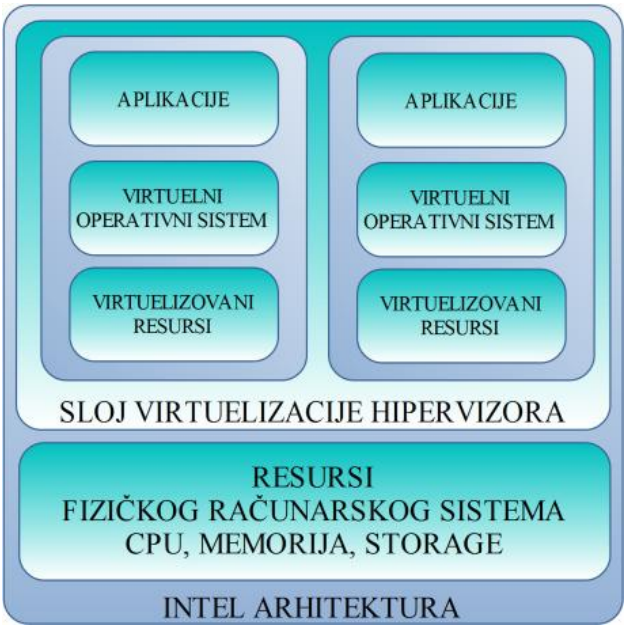
Slika 3.11 - Mogućnost redukcije IT infrastrukture na serverima na kojima su instalirani hipervizori

3.4. Tipovi virtuelizacije

Postoje tri tipa virtuelizacije:

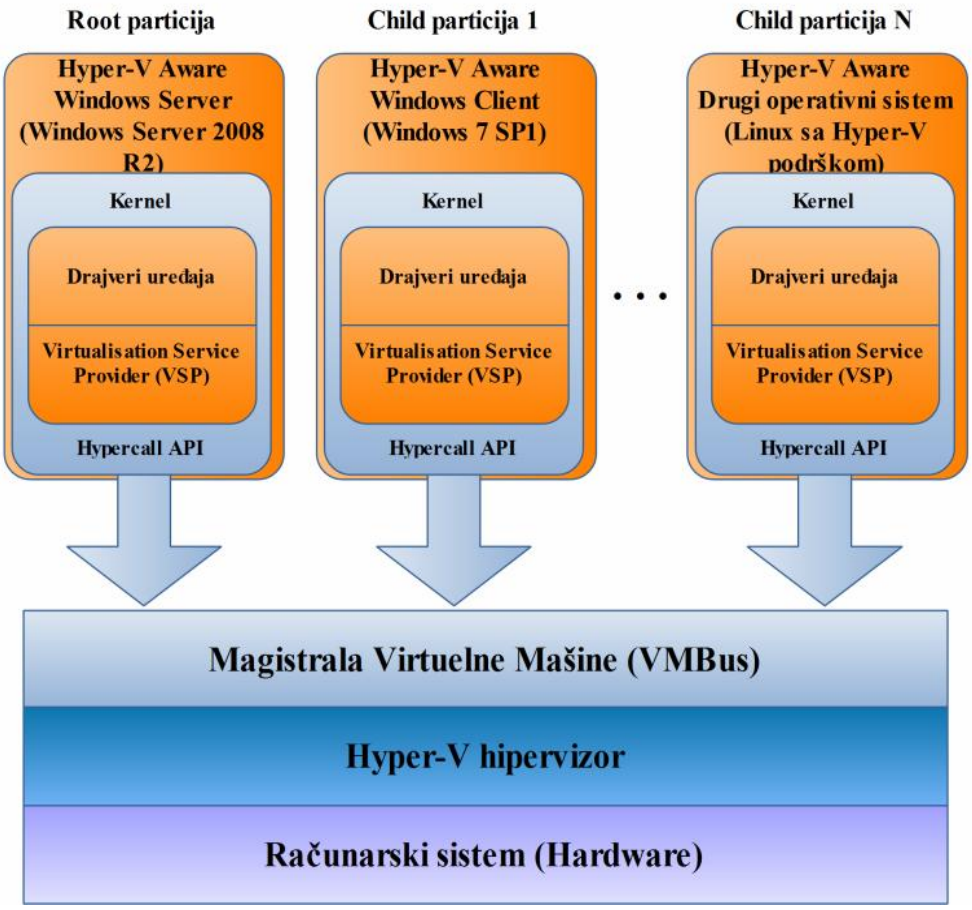
- Potpuna virtuelizacija,
- Paravirtuelizacija,
- Virtuelizacija na nivou operativnog sistema.

Pojam potpune virtuelizacije (slika 3.12) koristi prethodno pomenuti namenski softver koji se naziva hipervizor. Prilikom izvršavanja instrukcija, hipervizor je u direktnoj interakciji sa računarskim sistemom i njegovim hardverskim resursima. Jedan od glavnih zadataka hipervizora kod potpune virtuelizacije je da očuva integritet svakog gostujućeg operativnog sistema. Na taj način je moguće pokretanje više različitih platformi nezavisno.



Slika 3.12 - Dijagram primera potpune virtuelizacije

Paravirtuelizacija (slika 3.13) se razlikuje od potpune virtuelizacije. Razlika se odnosi na to da su kod paravirtuelizacije gostujući operativni sistemi u međusobnoj korelaciji. Osnovna prednost hipervizora kod paravirtuelizacije je da ne troši previše procesorske snage u upravljanju gostujućim operativnim sistemima, zato što su operativni sistemi na nivou sloja virtuelizacije međusobno svesni različitih zahteva, te se tako i resursi na nivou računarskog hardvera dinamički raspoređuju. U tom pogledu ceo računarski sistem funkcioniše kao jedinstvena koheziona jedinica. Tipičan primer paravirtuelizacije predstavlja Microsoft Hyper-V tehnologija.



Slika 3.13 - Primer paravirtuelizacije korišćenjem Hyper-V tehnologije

Kod virtuelizacije na nivou operativnog sistema (slika 3.14) ne koristi se hipervizor za kontrolu gostujućih virtuelnih mašina. Umesto njega za upravljanje resursima koristi se sama sposobnost virtuelizacije Host OS-a, koji pruža potpunu funkcionalnost kao hipervizor u slučaju potpune virtuelizacije. Najveći nedostatak ovakve virtuelizacije je ograničenje da sve virtuelne mašine koje se pokreću rade na nivou istog operativnog sistema kakav je kod Host OS-a. Svaki od virtuelizovanih operativnih sistema su nezavisni jedan od drugog ali platformski moraju biti istog tipa. Iz razloga potpune unifikacije operativnih sistema ovakav vid virtuelizacije se naziva i homogenim virtuelnim okruženjem.

Slika 3.14 - Virtuelizacija sa homogenim okruženjem

4. BEZBEDNOSNI ASPEKTI *CLOUD COMPUTING* PLATFORME

Cloud platforme imaju velike pogodnosti, kako za pojedince tako i za kompanije. Omogućuje se smanjenje troškova, deljenje resursa i mehanizama, skalabilnost, fleksibilnost servisa, mogućnost pristupa sa bilo kog mesta sa pristupom internetu, itd. Na ovaj način se smanjuje potreba za ažuriranjem softvera, kupovina i obnavljanje licenci, kao i održavanje sistema. Kako *cloud* vrši agregaciju resursa, provajderi obično angažuju eksperte za bezbednost sistema, dok obične kompanije angažuju mrežne administratore koji možda nemaju toliko iskustva u rešavanju bezbednosnih problema.

Novi koncepti koji postoje u *cloud* platformama, kao što je *outsourcing* proračuna, deljenje resursa i eksterno čuvanje podataka, povećava sigurnosne rizike i stvara nove izazove. Širok opseg *cloud* platformi, mogućnost pristupa mobilnim uređajima (*smartphone* i tablet uređaji), kao i direktan pristup *cloud* infrastrukturi povećava mogućnost zloupotrebe sistema. Kako su *cloud* platforme sve popularnije, tako rastu i bezbednosni rizici zbog koncentracije digitalnih dobara na jednom mestu.

Mnogi istraživači rade na prepoznavanju pretnji, napada i drugih sigurnosnih aspekata, kao i aspekta privatnosti u *cloud* sistemima. Njihova rešenja se sastoje u obezbeđenju kontramera u vidu bezbednosnih okvira, preporuka i servisno orijentisanih infrastruktura [28] [29]. Pored toga, rešavanje problema u drugim oblastima, kao što su *ad hoc* mreže, je povezano sa rešavanjem bezbednosnih problema u *cloud* infrastrukturi [30]. Mnogi istraživači su prepoznali pojedinačne bezbednosne probleme, kao što je integritet podataka, slabosti autentifikacije ili opšte slabe bezbednosti sistema [31] [32].

Drugi istraživači prikazuju probleme u specifičnim oblastima i pružaju određena rešenja za to [33].

Da bi uspešno odredili bezbednosne probleme u *cloud* sistemima, moraju se razumeti zajednički bezbednosni izazovi koji postoje u tim sistemima. Neophodno je:

- Istražiti različita svojstva bezbednosti u *cloud* sistemima, i to: pretnje, slabosti, rizici i modeli napada na sistem;
- Odrediti bezbednosne zahteve kao što je poverljivost, integritet, dostupnost, transparentnost i sl.;
- Prepoznati učesnike (klijenti, dobavljači usluga, spoljni i unutrašnji korisnici) i ulogu svakog od učesnika u modelu napada i odbrane;

- Razumeti uticaj bezbednosti na različite raspoložive *cloud* modele (javni, društveni, privatni, hibridni).

Osnovna namena ovog poglavlja jeste proučavanje bezbednosnih problema u *cloud* sistemima kojim bi bile pokrivene gotovo sve komponente (data centri, infrastruktura, umrežavanje), mrežni slojevi (aplikacioni, transportni, IP), kao i učesnici u *cloud* sistemima (provajderi, klijenti, i sl.).

4.1. Kategorizacija problema

Bezbednosni problemi u *cloud* sistemima su podeljeni u pet kategorija.

Sigurnosni standardi – U ovoj kategoriji su obuhvaćena regulatorna tela koja definišu polise bezbednosti u *cloud* sistemima kako bi osigurali bezbedno okruženje u svakoj arhitekturi. Ovim se obuhvataju SLA ugovori između korisnika, provajdera i drugih učesnika.

Mrežni problemi – Predstavlja medijume preko kojih se korisnici povezuju sa *cloud* infrastrukturom. Obuhvata veb pregledače, mrežne konekcije i razmenu informacija kroz registraciju.

Kontrola pristupa – Obuhvataju se problemi sa identifikacijom, autentifikacijom i autorizacijom korisnika.

***Cloud* infrastruktura** – Bezbednosni problemi unutar SaaS, PaaS i IaaS platforme. U suštini, problemi koji se javljaju u ovom delu javljaju se u virtuelizovanim okruženjima.

Podaci – Problemi sa integritetom i poverljivošću podataka.

Na osnovu prethodno navedenih kategorija, dalje se mogu raščlaniti vrste napada po kategorijama (tabela 4.1). *Cloud computing* platformama još uvek nedostaju dobro definisani bezbednosni standardi. Korisnici *cloud* platformi ne poznaju dovoljno procese i procedure provajdera. Ako servis provajder delegira servis trećoj strani gde nema transparentne funkcionalnosti, korisnici moraju biti u mogućnosti da ispitaju celokupan process. Sigurnosni standardi i regulatorna tela su deo SLA ugovora i pravnih normi, koji još uvek nisu u potpunosti uvedeni u praksu upotrebe *cloud computing* platformi. SLA definiše odnos između strana (provajder - korisnik) i bitan je za obe strane. On obuhvata definisanje potreba klijenata, uprošćavanje problema, smanjuje moguće oblasti nerazumevanja i uklanja nerealna očekivanja. Ako se ne koriste, korisnik može biti na gubitku u slučaju nestanka podataka. Ovim prethodno definisanim interakcijama se stvara međusobno poverenje, čime se stvaraju uslovi da korisnici prebace sve svoje podatke i poslove na *cloud* infrastrukturu.

Tabela 4.1 - Klasifikacija problema u *cloud* sistemima [34]

Sigurnosni standardi	**Mrežni problemi**	**Kontrola pristupa**	***Cloud* infrastruktura**	**Podaci**
Nedostatak sigurnosnih standarda	Pravilno podešen firewall	Zloupotreba usluga i naloga	Nebezbedan API interfejs	Redudandnost podataka
Problemi sa kompatibilnošću	Konfiguracija mrežne sigurnosti	Maliciozni insajderi	Kvalitet usluga (QoS)	Gubitak podataka
Nedostatak sertifikacije	Nedostaci mrežnih protokola	Mehanizam autentifikacije	Pouzdanost dobavljača	Oporavak podataka
Manjkavost pravne regulative	Internet međuzavisnost	Privilegovani korisnički pristup	Bezbednosni propusti	Privatnost podataka
Poverenje		Sigurnost veb klijenata	Lokacija servera i rezervne kopije (backup)	Zaštita podataka
				Dostupnost podataka

Problemi koji su povezani sa konfiguracijom računarskih mreža u *cloud* sistemima, kao i u internet vezama do *cloud* servisa predstavljaju najvažnije probleme. Sve *cloud* operacije su usko povezane i zavisne od umrežavanja. Takođe, kako se broj korisnika *cloud* platformi povećava i kako je količina podataka koja migrira na *cloud* sve veća, tako raste i broj napada na *cloud* sisteme. Mogući nedostatak pravilne instalacije mrežnih *firewall* uređaja, kao i loše podešeni sigurnosni protokoli olakšavaju pristup napadačima. Napadači mogu onemogućiti pristup resursima (hardveru i/ili aplikacijama) ili mogu pokretati svoje štetne programe na *cloud* sistemu i na taj način naneti štetu kako tom sistemu tako i svim korisnicima sistema.

Odbijanje usluga (DOS napadi) može biti pokrenuto iskorišćavanjem nedostataka u mrežnim protokolima, što kasnije dovodi do nesigurnih Internet konekcija.

Migriranje na *cloud* povećava zavisnost od interneta kao glavnog medijuma za pristup *cloud* sistemima. Ako je zbog napada pristup internetu onemogućen, samim tim je onemogućen i pristup *cloud* servisima, čime se sav posao koji je zavisan od tih servisa zaustavlja.

Zloupotreba naloga i usluga obuhvata *phishing*, prevaru i zloupotrebu softvera, gde napadač krade privilegije i dobija neautorizovani pristup serverima. Ovaj neautorizovani pristup predstavlja pretnju po integritet, poverljivost i pristup podacima i servisima. Neautorizovani pristup se može obaviti unutar ili van organizacije. Maliciozni insajderi mogu biti administratori sistema koji zloupotrebe svoj položaj. Time se umanjuje sigurnost organizacije, uništava brend i stvaraju se ogromni finansijski gubici, od čega se kompanije teško

oporavljaju. Za klijente *cloud* platformi veoma je važno da provajderi pruže garancije da će ih zaštititi od unutrašnjih pretnji.

Klijent može pristupiti sistemu i servisima upotrebom mobilne aplikacije ili veb klijenta. Na ovaj način sistemu može pristupiti neko ko nema te privilegije, i to iskorišćavanjem slabosti samog veb klijenta. Dakle, klijent predstavlja prvi korak gde treba preduzeti mere zaštite, jer se na taj način sprečavaju svi dalji potencijalni napadi na sistem.

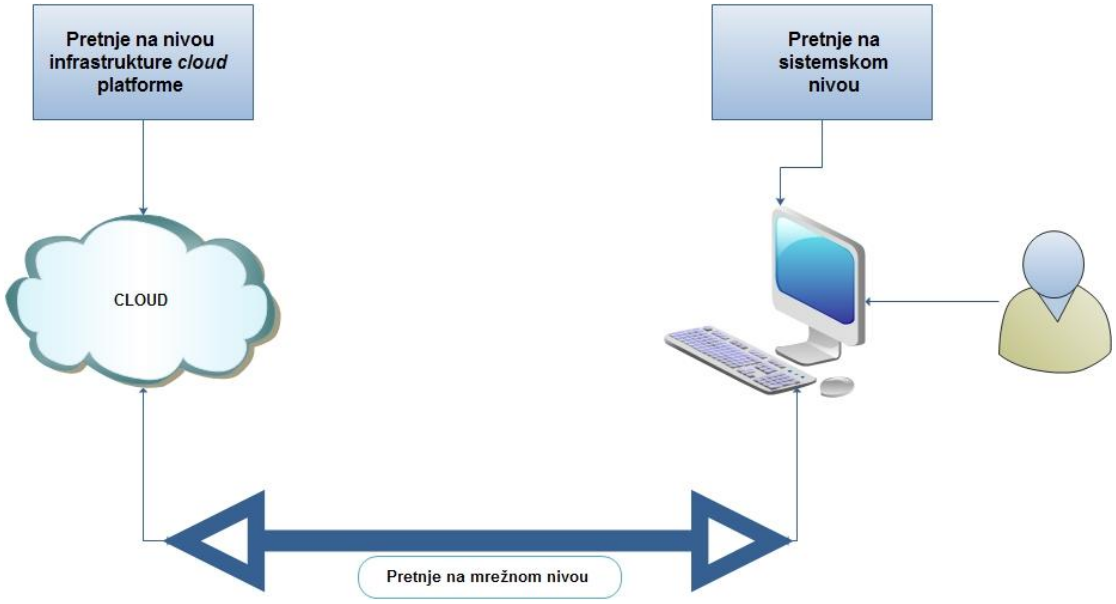
Ako se prilikom prijavljivanja na *cloud* sistem koristi API, zlonamerni korisnici mogu iskoristiti eventualne slabosti API-ja za neautorizovani pristup sistemu.

Kvalitet usluge (QoS) predstavlja slabo opisani problem kod *clouda*, jer većina provajdera obraća pažnju samo na brzinu i nisku cenu usluge. Međutim, treba postaviti pitanje kako QoS utiče na samu bezbednost sistema. Jednostavna greška u konfiguraciji neke od komponenti može uzrokovati znatne posledice, jer ta *cloud* konfiguracija može biti korišćena od strane više različitih servisa. Tehnički nedostaci, u kome se greške prenose sa servera do svake virtuelne mašine koja je kreirana na tom serveru se još više pogoršavaju kada se transfer dodatno obavlja i preko mobilnih virtuelnih mašina do drugih servera. Iz tog razloga, od primarne je važnosti da se takve greške prepoznaju na vreme i spreči dalje urušavanje sistema. Pouzdanost provajdera predstavlja značajan faktor koji zahteva dodatnu kontrolu osoblja koji imaju fizički pristup serverima i kontrolišu podatke.

Serveri u *cloud* sistemima predstavljaju osnovu infrastrukture koja pruža mnoge servise, kao što je skladištenje podataka ili servis elektronske pošte. Napadači mogu pristupiti sistemu ako bezbednosne polise servera nisu pravilno konfigurisane. Pogrešna konfiguracija se može dogoditi kod veb servera, kod upravljanja aplikacijama ili čak u samom platformskom kodu. *Cloud* serveri opslužuju više korisnika simultano kroz virtuelizaciju, čime se omogućuje deljenje istih hardverskih i softverskih resursa od strane različitih korisnika. Ovakva sposobnost sistema može dovesti do curenja informacija od jednog do drugog korisnika.

Što se fizičke lokacije servera tiče, važno je poštovati određene bezbednosne procedure. Važno je da prostorije gde se čuvaju serveri koji čine osnovu *cloud* sistema nemaju prozore, da pod bude antistatički, i da serverski ormari imaju propisano uzemljenje. Naravno, i pored svih preduzetih mera zaštite, važno je da postoje rezervne kopije podataka koje će se čuvati na nekom drugom, udaljenom mestu.

Kod zaštite podataka, bitno je da podaci budu enkriptovani, čuvani, kontrolisani i dostupni kada je to potrebno.



Slika 4.1 - *Cloud* komponente koje mogu biti podložne napadima

Na slici 4.1 prikazane su komponente sistema gde može doći do sigurnosnih problema. Svaka od komponenti, kao što su polise, klijenti, *cloud* infrastruktura i mreža, pokazuju slabosti na određene napade i zato zahtevaju određene strategije prevencije, prepoznavanja opasnosti i davanja odgovora na napade.

4.2. Analiza napada i odbrane od napada

U ovom odeljku biće prikazana klasifikacija napada. Predstavljeno je osam tipova napada, gde će svaki od njih biti opisan primerom. Takođe, za svaki od napada biće prikazana analiza uz pomoć poznatih tehnika zaštite.

4.2.1. Krađa servisa

Krađa servisa koristi nedostatke raspoređivača hipervizora. Napad se realizuje kada hipervizor koristi mehanizam raspoređivanja, koji u tom trenutku ne uspe da detektuje upotrebu servisa, već utrošak procesora pripisuje loše realizovanim virtuelnim mašinama. Na ovaj način, dalje je moguće da neki korisnici koriste servise na račun drugih korisnika. Ovakav vid napada dolazi do izražaja u javnim *cloud* plaformama gde se korisnicima naplaćuje vremensko korišćenje, a ne samo vreme utrošenog procesorskog vremena. Kako hipervizor raspoređuje virtuelne mašine, nedostaci u raspoređivanju mogu rezultovati u netačnom raspoređivanju. Ovi nedostaci dolaze do izražaja ako se koristi periodično proveravanje. Na taj način, neko ko zna kako sistem funkcioniše, može iskoristiti sistem za svoje potrebe, npr. kao putnik u vozu bez karte, koji se sakrije svaki put kada naiđe kontrolor karata. Kod ovakve vrste napada,

napadač osigurava da njegov proces korišćenja nikada nije aktivan u trenutku kada sistem proverava aktivnost.

Rešenje ovog problema su pružili Fangfei i autori u [35], i to modifikovanjem raspoređivača radi sprečavanja napada, bez umanjenja efikasnosti. Modifikovani raspoređivači su: 1. precizni raspoređivač, 2. uniformni raspoređivač, 3.*passion* raspoređivač i 4. Bernulijev raspoređivač. Osnovna razlika između ovih raspoređivača sastoji se u polisama raspoređivanja i praćenja, kao i u kalkulacijama vremenskih intervala. Eksperimenti koje su sproveli autori prethodno navedenog rada pokazuju bolje i fer raspoređivanje sa izmenjenim raspoređivačima. Modifikacije su se pokazale boljim od Xen hipervizora (koji se koristi kod Amazon EC2 i OpenStack *cloud* sistema).

4.2.2. DOS i DDOS napadi

Postoje mnoge vrste napada koji su poznati u okruženju sa klasičnom IT infrastrukturom, a sada se mogu primeniti i na *cloud computing*. Kako su podaci izmešteni na *cloud* server bez transparentne kontrole, sigurnosne pretnje se ne posmatraju na isti način kao ranije [36].

Jedan od takvih napada jeste napad odbijanjem usluge (Denial of Service – DoS) [37]. Svrha servera je da opsluži klijente koji mu šalju određene zahteve za isporukom nekog sadržaja koji se nalazi na serveru. DoS napadači ciljaju određeni server koji žele da napadnu izgledajući kao običan korisnik. Cilj napadača je da preplave server svojim zahtevima u toj meri da server više ne bude u stanju da opsluži prave klijente i samim tim postane potpuno nedostupan.

Sličan pristup se koristi i kod najvećeg broja napada na *cloud* sisteme i naziva se distribuirano odbijanje napada (*distributed denial of service* - DDOS), tj. DDOS napadi zasnovani na HTTP, XML, REST, ICMP ili UDP ECHO protokolima. Cilj je iskoristiti protokole koji mogu stvoriti veliku količinu dvosmernog saobraćaja (slanje zahteva uz zahtev za slanje povratnih informacija). Zbog nedostataka u sistemskom interfejsu, DDOS napade je lakše implementirati, a ekspertima teško zaustaviti takve napade [38]. DDOS napadi koji su zasnovani na XML i HTTP protokolima su destruktivniji od klasičnih DOS napada jer su ovi protokoli veoma zastupljeni u *cloud* sistemima, bez nekih mehanizama za zaobilaženje tih protokola. HTTP i XML su važni elementi, tako da bezbednost kod ovih protokola predstavlja važan faktor za obezbeđenje kvalitetnog razvoja *cloud* platforme.

Ovakvim ciljanim napadima se stvara velika količina saobraćaja, što može rezultovati zagušenjem protoka ciljanog sistema zbog velikih količina podataka koji se šalju velikom

brzinom (od 1 Gbps – 300 Gbps) [39]. Pored generisanja zagušenja, velika količina saobraćaja stvara i dodatne troškove [40].

Jedan od tipičnih distribuiranih napada naziva se *botnet*, koji se smatra svestranim i upravo se koristi za slanje velikih količina podataka na ciljanu lokaciju [41]. Kod *botnet* napada, napadači pokušavaju da naprave štetu dok izbegavaju otkrivanje svoje lokacije i identiteta kako bi smanjili mogućnost sopstvene detekcije i praćenja. Ovo se postiže indirektnim napadima preko drugih računara. Ovakvi hostovi se zauzimaju putem ilegalnih botnet mreža. Onaj koji upravlja takvom mrežom – *bot-master*, može davati komande i kontrolisati računare i servere u *cloud* platformi kako bi došao do osetljivih informacija i dobio neovlašćeni pristup resursima. Proteklih godina bilo je prijavljenih botnet napada kod poznatih *cloud* provajdera kao što su Amazon EC2 i Google AppEngine. Za napad je korišćen alat Zeus (Trojan.Zbot) koji se sastoji iz tri dela. Prvi je kontrolni panel, koji se instalira na serveru i upravlja botovima. Drugi deo je „Builder“, Windows aplikacija, koja služi za konfigurisanje bot programa. I treći predstavlja bot program, takođe Windows aplikacija, koja se nalazi na računaru žrtve. Zeus kontrolni panel je instaliran na Amazon EC2 [42]. Kompjuter je inficiran upotrebom prosleđenih komandi preko Google App Engine platforme [43], što je dalje napadačima omogućilo da ukradu podatke sa Raytheon *cloud* sistema [44]. *Cloud* okruženje predstavlja idealno mesto za ovu vrstu napada. *Cloud* poseduje puno resursa (protok, procesorska moć, skladišni prostor) kojima se lako pristupa. Napadač u tom slučaju može upotrebiti *cloud* server kao svoj komandni server ili može iznajmiti virtuelnu mašinu sa visokim performansama, a platiti ukradenim kreditnim karticama.

Do sada se puno radilo na otkrivanju računara koji su zloupotrebljavani na prethodno opisani način. Detekcija se obično oslanja na hipotezu jake korelacije između dolaznog i odlaznog saobraćaja potencijalno zaraženog računara. Takva korelacija može biti zasnovana na paketnom saobraćaju, ponašanju prilikom ulaska na sistem, učestalosti mrežne aktivnosti, vremenskim karakteristikama i periodičnosti mrežnog saobraćaja. Ipak, mnoge od ovih karakteristika je lako lažirati upotrebom kriptovanog saobraćaja, lažiranjem autentifikacije ili slučajnim kašnjenjima.

Jedna od najbolje isprobanih tehnika do sada jeste šema praćenja bot-mastera [45]. U prvom koraku prepoznaju se kriptografski ključevi botnet komunikacije za konfigurisanje operacija, a potom se vrši praćenje do bot mastera. Ovaj način obuhvata dizajn i implementaciju nove šeme identifikacije ključa i praćenje bot mastera preko zaraženih računara iza više *cloud* sistema.

Jedno od bezbednosnih rešenja za zaštitu od DDOS napada predloženo je od strane Karnwala i autora [46], gde je dat opis aplikacije „*Cloud* defender“, koja se sastoji iz pet delova:

- filter senzora,
- filter broja skokova,
- filter IP različitosti,
- „puzzle resolver“ filter i
- filter dvostrukog potpisa.

Prva četiri filtera detektuju HTTP DDOS napade, a peti filter detektuje XML-bazirane napade. REST-bazirani napadi su pomenuti, ali bez nekog konkretnog rešenja za njihovo zaustavljanje. Jedan od razloga je taj što su REST-bazirani napadi blisko povezani sa korisničkim interfejsom, koji može varirati od korisničkih pa do sistemskih aplikacija. Te aplikacije su po svojoj prirodi drugačije, na osnovu toga šta se od njih traži i čemu su namenjene, tako da ne postoji jedinstveno pravilo za implementaciju bezbednosnih mera na tom nivou. Rešenje se sastoji iz sledećih modula:

Senzor: Prati dolazeće poruke sa zahtevima. Ako senzor odredi da postoji povećanje zahteva od određenog korisnika, on ga označava kao sumnjiv.

Filter broja skokova: Služi za brojanje vrednosti broja skokova (koliko čvorova, da li poruka putuje od izvora do odredišta) i upoređuje sa predefinisanim brojem skokova. Ako postoji razlika u broju, to znači da postoji mogućnost da je zaglavlje poruke ili poruka promenjena na kompjuteru napadača i zato se označava kao sumnjiva.

Filter IP različitosti: Obeležava poruke sumnjivim ako se šalje veliki broj poruka sa iste IP adrese.

Dvostruki potpis: Udvostručuje XML potpise. Jedan potpis se nalazi u zaglavlju a drugi na kraju poruke. U slučaju napada, neophodno je proveriti oba XML potpisa.

„***Puzzle Solver***“: Ovaj filter funkcioniše po sistemu rešavanja slagalice, gde je rezultat uključen u zaglavlje SOAP (Simple Object Access Protocol) protokola. U slučaju napada (HTTP DDOS), *cloud defender* će pošiljaocu poruke poslati da reši slagalicu na zadatu IP adresu. Ako *cloud defender* primi rešenu slagalicu onda se poruka i IP adresa obeležava kao sigurna, u suprotnom se obeležava kao HTTP DDOS napad.

Nedostatak ovog sistema jeste što zahteva iscrpno praćenje. Iscrpno praćenje svih poruka bi znatno usporilo tok saobraćaja u mreži. Konačno, sistem nema mehanizam upravljanja čvorovima u slučaju detekcije napada.

Riquet i autori u svom radu [47] tvrde da “Ne postoji dovoljno dobro rešenje koje bi sprečilo DDOS napade”. Kako bi opravdali tvrdnju, autori su izveli eksperiment kako bi izvršili

evaluaciju efikasnosti trenutnih bezbednosnih rešenja protiv distribuiranih napada. Bezbednosna rešenja koja su korišćena u eksperimentu su SNORT [48] i komercijalni *firewall*. Autori zaključuju da postoje dva razloga zašto bezbednosni sistemi ne funkcionišu:
Ili je bezbednosno rešenje zastarelo jer nije ažurirano, ili se sistem oslanja na nepouzdane metode.

Nedavno je sve aktuelnije pitanje interne pretnje. Naime, napadači mogu da planiraju napad na *cloud* iznajmljivanjem velikih količina resursa samog *clouda* i potom sprovođenja napada na drugi deo *clouda* ili na određenu virtuelnu mašinu koja se nalazi u istom *cloudu* [49]. U ovom slučaju, pravilna konfiguracija *firewall* uređaja unutar same mreže *cloud* provajdera je ključni deo odbrane od napada.

4.2.3. Napadi izazvani ubacivanjem *malware* programa

Napadi ubacivanjem *malware* programa u *cloud* sistem se odnose na prikrivanje napada u vidu pružanja servisa žrtvi, a iza toga se krije neovlašćeni pristup korisničkim podacima.
Otkrivanje ovakve vrste napada je teško, jer ne samo da je potrebno detektovati ubacivanje zlonamernog programa u sistem, već je isto potrebno odrediti tačan čvor gde je napadač ubacio program [50].

Tokom napada u *cloud* okruženju pokušava se implementacija malicioznog servisa (SaaS ili PaaS) ili virtuelne mašine (IaaS). Napadač svojim servisom pokušava da prevari sistem tako da se novi servis tretira kao nova implementacija validne instance. *Cloud* sistem dalje vrši redirekciju maliciozne instance na virtuelne mašine i maliciozni kod se izvršava. Posledice napada obično obuhvataju izmenu ili krađu akreditiva, privatnih podataka korisnika, kao i neautorizovani pristup ili blokiranje resursa.

Za detektovanje ovakvih napada obično se koristi retrospektivna detekcija (ispitivanje operativne i sekundarnih memorija). Liu i autori predlažu novi retrospektivni pristup zasnovan na principima veza sa *portable executable file* (PE) formatom [51]. Ovakav pristup je implementiran u HADOOP [52] platformi i dokazano obezbeđuje veći postotak otkrivanja neželjenih programa, a smanjuje mogućnost lažnih otkrivanja. Osnovni principi po kojima ovaj pristup funkcioniše su:

- Većina legitimnih programa i *malware* fajlova su u izvršnom (.exe) formatu i izvršavaju se na Windows platformi.
- Broj legitimnih fajlova je veći od broja *malware* fajlova.
- Kreiranje/čitanje fajlova sa izvršnim formatom retko se dešava na korisničkom računaru.

Napadač može iskoristiti bilo kakav nedostatak u *cloud* sistemu bez potrebe da neke od ovih predispozicija budu ispunjene. Autori nisu naveli posledice ako neka od pretpostavki nije ispunjena, kao i kolika bi šteta bila načinjena ako ovakvih pretpostavki nema.

Zaštita od malicioznih napada se može implementirati automatskom proverom integriteta sistema pre upotrebe instance. Ovo se može postići čuvanjem *hash* vrednosti originalnog *image* fajla instance i upoređivanjem vrednosti sa *hash* vrednostima novih *image* fajlova novih instanci. Napadač bi onda morao da pronađe način kako da zaobiđe proveru *hash* vrednosti kako bi uspeo da ubaci maliciozni kod u sistem [53].

Još jedna protivmera napadima se naziva CloudAV i opisan je kao efikasan, tačan i brz sistem za detekciju *malware* programa [54].

CloudAV ima dva bitna servisa, i to:

Antivirus kao mrežni servis: Mogućnosti detekcije host antivirusa mogu biti mnogo veće ako se servis realizuje kao *cloud* mrežni servis. Na svakom host računaru ili mobilnom uređaju izvršava se proces koji detektuje nove fajlove a potom ih šalje mrežnom servisu u karantin radi dalje analize, što je efikasnije nego da se na svakom računaru izvršava kompleksni softver za analizu ostalih programa.

Zaštita N-verzije: Identifikacija malicioznog softvera je određena sredstvima za višestruku heterogenu detekciju, slično kao ideja programiranja N-verzije. Ipak, nedostatak je nešto veći broj lažno pozitivnih rezultata u poređenju sa jednostrukom detekcijom.

Autori dokazuju efikasnost programa kroz validaciju u *cloud* okruženju. CloudAV obezbeđuje bolju detekciju malicioznog softvera, poboljšava forenzičke mogućnosti, novi način detekcije kroz retrospektivnu detekciju i poboljšano upravljanje. Eksperimentom je utvrđeno da CloudAV pruža za 35% bolju detekciju pretnji u odnosu na obične antivirusne alate, kao i pokrivenost nad podacima u *cloud* sistemu od 98%.

Ipak, bezbednosna rešenja koja su namenjena *cloud* platformama obično imaju zajedničke probleme, kao što je bezbednosna pokrivenost, skalabilnost i privatnost. Kako *malware* može biti ugrađen u veliki broj fajlova i vrsti fajlova, napadači mogu premostiti takva rešenja, jer detekcioni sistemi obično pokrivaju manji broj vrsti fajlova i time se smanjuje pokrivenost detekcije. Eksportovanjem programa ili dokumenata na *cloud* radi istraživanja ne vrši se skaliranje i može stvoriti greške punjenjem *cloud* platforme izvršnim fajlovima. Konačno, slanjem izvršnih fajlova i dokumenata u *cloud* radi provere može stvoriti probleme sa privatnošću, jer uvek postoji rizik da je neki fajl sa osetljivim sadržajem takođe postavljen na *cloud*.

4.2.4. Napadi na virtuelne mašine

Kod ove vrste napada, napadačeva virtuelna mašina se izvršava zajedno sa virtuelnom mašinom žrtve, gde se utiče na keš memoriju procesora kako bi se došlo do žrtvine memorije. Ova vrsta napada zahteva da se obe virtuelne mašine izvršavaju na istom hardveru. Jedan od načina curenja podataka sa virtuelne mašine jeste putem merenja vremena potrebnog za računanje raznih operacija [55]. Promenom načina merenja može doći do curenja osetljivih informacija o vlasniku tih informacija ili čak i o *cloud* provajderu. Vremenske kanale je posebno teško kontrolisati zbog masovnog paralelizma kod *cloud* platformi. Takođe, ovakve napade je teško otkriti jer ne ostavljaju nikakav trag iza sebe. Klijenti obično nemaju ovlašćenja da provere vreme pristupa informacijama i odakle im je pristupano.

Ovakvi ciljani napadi na implementirane kriptografske algoritme predstavljaju veliku pretnju, tako da provajderi moraju obratiti posebnu pažnju prilikom dizajniranja bezbednosnog sistema [56].

Druga vrsta napada se sastoji od ispitivanja logova potrošnje energije u sistemu [57]. Umesto direktnog napada na softverski stek (sloj virtuelizacije), napadači mogu direktno sakupiti osetljive informacije o *cloud* sistemu preuzimanjem logova kojim se prati potrošnja električne energije. Ovakvi podaci služe za praćenje trenutnog statusa infrastrukture i za pružanje informacija radi povećanja energetske efikasnosti. U [58] se ispituje mogućnost izvlačenja vrednih informacija iz ovakvih logova, i njihov mogući uticaj na privatnost i bezbednost. U *cloud* okruženju mogu postojati na desetine hipervizora, gde svaki od njih može biti zadužen za ciljanu virtuelnu mašinu. Iz tog razloga, napadaču je potrebno dosta vremena dok utvrdi koji hipervizor je zadužen za ciljanu mašinu. Što je više vremena potrebno napadaču, to je veća mogućnost otkrivanja napada. Ako napadač može nekako doći do ovih logova, postoji izvesna mogućnost da napadač odredi servere koji opslužuju ciljanu virtuelnu mašinu. Ovo napadaču daje veće šanse da odredi tačan server pre otkrivanja. Trenutno ne postoji efikasan način zaštite od pomenutog napada.

4.2.5. Napadi malicioznih insajdera

Maliciozni insajderi mogu biti osobe koje su zaposlene kod provajdera, korisnik ili poslovni partner. Maliciozni insajderi mogu ukrasti važne poslove informacije o kompaniji ili samoj *cloud* infrastrukturi, tako da ovo može biti ozbiljna bezbednosna pretnja [59].

Maliciozni insajder se definiše kao „trenutni ili bivši zaposleni ili drugi poslovni partner koji je imao ili ima pristup mreži, sistemu ili podacima i namerno je zloupotrebio mogućnost pristupa kako bi negativno uticao na poverljivost, integritet ili dostupnost informacijama ili

informacionim sistemima kompanije“ [60]. Kod *cloud* platformi to je početni nivo napada koji dalje može voditi do nekoliko drugih vrsti napada, kao što su napadi na virtuelne mašine sporednim kanalima kao i napadi ubacivanjem *malware* programa.

Maliciozni insajder može pristupiti i memoriji virtuelnih mašina [61]. Ovakav pristup vodi do izvlačenja trenutno aktivnih procesa u sistemu i do privatnih informacija korisnika.

Za sada, jedini način za sprečavanje ovakvih napada jeste da se antivirusnim ili *firewall* programima i algoritmima ograniči pristup korisnicima ka deljenoj memoriji.

4.2.6. Phishing napadi

Phishing (pecanje) predstavlja pokušaj pristupa ličnim podacima korisnika putem tehnika socijalnog inženjeringa. Rezultat se postiže slanjem linkova veb strana u porukama elektronske pošte ili putem programa sa instant porukama. Takvi linkovi su na prvi pogled uobičajeni, koji vode do sajta za pristup bankovnom računu ili za verifikaciju podataka o kreditnoj kartici, ali zapravo se na taj način korisnici dovode do lažnih lokacija. Ovakvom obmanom, napadač može prikupiti osetljive informacije kao što su lozinke i podaci o kreditnim karticama. *Phishing* napade možemo podeliti u dve kategorije:

- Napadač ima *phishing* sajt na *cloudu* upotrebom nekog od servisa i
- Zloupotreba naloga i servisa putem klasičnih tehnika socijalnog inženjeringa.

Cloud security alliances (CSA) upozorava da *cloud* provajderi ne održavaju dovoljan nivo kontrole nad svojim sistemima kako bi izbegli hakovanja ili spamovanja. Da bi izbegli takve napade, CSA predlaže nekoliko mera kao što je striktni proces registracije, bezbedna provera identiteta i poboljšane tehnike praćenja [62]. Zbog poštovanja zakona koji regulišu privatnost korisnika, *cloud* provajderima nije moguće da prate ono što korisnici rade, tako da ako maliciozni pojedinac ili organizacija radi nešto štetno (*phishing* napad ili postavljanje malicioznog programa) upotrebom *cloud* servisa, ne može biti otkriven sve dok ga ne detektuje neki od softvera za regulisanje bezbednosti.

4.2.7. Napadi vraćanjem virtuelne mašine na prethodno stanje

Virtuelizovano okruženje na *cloud computing* platformi predstavlja najosetljivije područje koje je pogodno za napade. Hipervizor može isključiti virtuelnu mašinu (VM) bilo kada, može snimiti trenutno stanje procesora, diska i memorije, kao i kasnije nastaviti rad iste virtuelne mašine a da ciljana virtuelna mašina ne bude „svesna“ toga. Ove mogućnosti se uobičajeno koriste za održavanje virtuelnih mašina i sprečavanje grešaka. Ali, te iste mogućnosti otvaraju mogućnosti napadačima da aktiviraju napade na virtuelnu mašinu. Kod napada metodom

vraćanja virtuelne mašine na prethodno stanje, napadač može iskoristiti prethodno sačuvano stanje mašine, pokrenuti je bez znanja korisnika i potom izbrisati istoriju korišćenja, a zatim ponovo pokretati istu ili druge virtuelne mašine na isti način. Brisanjem istorije korišćenja, napadač ne može biti uhvaćen. Na primer, napadač može pokrenuti *brute force* napad kako bi pogodio lozinku za pristup virtuelnoj mašini, čak i ako OS te virtuelne mašine ima zabranu višestrukog pogađanja lozinke, jer nakon svakog pokušaja napadač može vratiti stanje virtuelne mašine na vreme pre pokušaja i pokušati ponovo, sve dok ne pogodi lozinku [63].

Jedna od mera protiv ovakvih napada se zove „Hyperwall“ [64]. Prevencija ovakvih napada se zasniva na tome da se hipervizoru onemogući da pauzira i nastavi rad virtuelne mašine. Na ovaj način, sistem bi morao da pita krajnjeg korisnika za dozvolu svaki put kada se sistem ponovo pokrene, migrira ili isključuje, što može biti nepraktično.

4.3. Evaluacija sistema za zaštitu cloud platformi

Tri *cloud* servis modela (SaaS, Paas i IaaS) ne samo da pružaju različite vrste servisa krajnjim korisnicima, već kriju i različite bezbednosne rizike. IaaS platforma se nalazi na najnižem sloju, što direktno daje najmoćniju funkcionalnost celom *cloud* sistemu. On omogućuje korisnicima da prilagode okruženje koje može imati razne virtuelne mašine sa proizvoljnim operativnim sistemima na njima. Napadači mogu iznajmiti virtuelne mašine, analizirati njihovu konfiguraciju, pronaći slabosti i zatim napasti virtuelne mašine drugih korisnika u istom *cloud* sistemu. IaaS servis omogućuje napadačima da izvršavaju *brute force* napade, gde je potrebna velika kompjuterska snaga. Kako IaaS podržava višestruke virtuelne mašine, time se obezbeđuje idealna platforma za DDoS napade, koji zahteva veliki broj napadačkih računara [65].

Gubitak podataka predstavlja važan sigurnosni rizik. Kod SaaS *cloud* modela, kompanije koriste aplikacije za obradu poslovnih podataka i čuvanje podataka o korisnicima u data centrima. Kod PaaS *cloud* modela, programeri koriste podatke za testiranje integriteta softvera tokom razvoja sistema. Kod IaaS *cloud* modela, korisnici kreiraju nove diskove na svojim virtuelnim mašinama i čuvaju podatke na njima. Ipak, podacima kod sva tri *cloud* modela može biti pristupano od strane neovlašćenih lica zaposlenih kod *cloud* provajdera, kao i od strane spoljnih napadača. Zaposleni kod provajdera mogu pristupati podacima slučajno ili namerno. Spoljni napadači dobijaju pristup bazama podataka u *cloudu* upotrebom različitih metoda, kao što je prisluškivanje mrežnog kanala ili preuzimanje sesije.

Klasične strategije mrežnih napada se mogu primeniti za narušavanje integriteta ova tri sloja u *cloud* sistemima. Na primer, napadom na veb-klijenta se iskorišćavaju nedostaci u

autentifikaciji i autorizaciji naloga *cloud* sistema. Virusi i trojanci mogu biti postavljeni na sistem i izazvati štetu. Hipervizor takođe može biti zloupotrebljen za manipulaciju virtuelnim mašinama.

U sledećem odeljku biće predstavljeni savremeni alati koji mogu sprečiti napade na *cloud* platforme.

4.4. Sistemi za detekciju upada

Uljezi, koji se predstavljaju kao obični korisnici, mogu pristupiti *cloud* infrastrukturi, pri čemu infrastruktura postaje nedostupna običnim korisnicima. Dokazano je da napadači mogu lako doći do podataka žrtvinih mašina kod IaaS servisa [66]. Ovakvi podaci mogu pomoći u napadima na *cloud* korisnike. Ti napadi se obično sastoje i iz DoS i DDoS napada kojima se ugrožava integritet i dostupnost podataka. Ovakve napade je moguće izbeći implementiranjem sistema za detekciju upada (IDS – Intrusion Detection Systems), čime se konstantno analizira mrežni saobraćaj, log fajlovi i ponašanja korisnika [67]. Sistem za detekciju upada se definiše kao sistem koji sakuplja i analizira podatke bezbednosnih razlika kako bi proverio da li postoji prekršaj mrežnih pravila. Detekcija upada se može klasifikovati u dve kategorije:

- Detekcija zloupotreba i
- Detekcija anomalija [68].

Detekcijom zloupotreba stavlja se naglasak na karakteristike podataka korisnika i upoređuju se sa rezultatima iz baze podataka. Sa druge strane, komponenta detekcije anomalija čuva korisničke navike u bazi podataka, gde se potom vrši upoređivanje sa trenutnim navikama. Ako postoji velika razlika u upoređivanju, onda sistem detektuje napad. Postoje tri vrste sistema za detekciju napada:

- sistem zasnovan na jednom računaru, gde se prati ponašanje jednog računara (host-based intrusion detection system - HIDS) i
- mrežni sistem, gde se analizira protok saobraćaja u mreži (network-based intrusion detection system - NIDS).
- hibridni sistem, koji kombinuje prethodna dva sistema.

4.4.1. SNORT

Snort [69] je sistem za mrežnu detekciju upada (NIDS). On ima sposobnost izvršavanja analize saobraćaja u realnom vremenu, kao i praćenja paketa u IP mrežama. Snort vrši analizu protokola i pretragu sadržaja. Ovi osnovni servisi mogu služiti za povećanje kvaliteta usluge ili smanjenje prioreta određenog dela saobraćaja kada se koriste aplikacije osetljive na

kašnjenje. Program može prepoznati razne vrste napada na mrežama, kao što su skeniranje portova, prikupljanje podataka o operativnim sistemima na host računarima, prelivanje podataka (*buffer overflows*), itd.

Snort ima tri načina rada: *sniffer, packet logger* i *network intrusion detection system (NIDS)* [70]. U *sniffer* režimu rada, program će čitati mrežne pakete i prikazivati ih u konzoli. U *packet logger* režimu rada program će beležiti podatke o svakom paketu na disk. U *network intrusion detection* režimu, program će pratiti mrežni saobraćaj i analizirati po pravilima koje je postavio administrator i preduzimati konkretne akcije.

Snort ima vrlo preciznu statistiku i podržava veliki broj protokola, tako da predstavlja jedan od najboljih alata za prepoznavanje napada u računarskim mrežama.

4.4.2. Wireshark

Wireshark [71] je program otvorenog koda koji služi za analiziranje paketa u računarskim mrežama. Koristi se kod rešavanja problema u mrežama, za analiziranje, razvoj softvera i komunikacionih protokola i u obrazovanju. On nije klasični sistem za detekciju upada na mreži zbog svoje pasivne uloge, ali može pomoći prilikom napada kako bi administrator otkrio o kakvoj vrsti napada je reč i koji je izvor napada.

Wireshark omogućuje nadgledanje celokupnog saobraćaja koji prođe kroz jednu mrežnu karticu, uključujući saobraćaj koji se odnosi na računar kome kartica pripada i ostali *multicast/broadcast* saobraćaj. Program podržava i praćenje saobraćaja u bežičnim mrežama.

Wireshark podržava razne protokole i tipove mreža, a sposoban je i za presretanje i snimanje VoiP poziva [72]. Dakle, program ne može manipulisati podešavanjima na mreži, ali može detaljno analizirati pakete i protokole.

4.4.3. GCCIDS

Jedno od predloženih sistema za praćenje *je Grid and Cloud Computing Intrusion Detection System* (GCCIDS) [73] koji je zasnovan na NIDS i HIDS sistemima. Sastoji se iz sistema koji otkriva napade koji pre toga nisu prepoznati od strane drugih NIDS i HIDS sistema. Sistem radi integrišući analize ponašanja kako bi se prepoznali napadi na *cloud* sistem. Osnovne komponente GCCIDS sistema jesu čvorište, servis, revizor događaja i sistem za skladištenje. Svako čvorište određuje lokalni događaj i uzbunjuje druga povezana čvorišta. Kako broj čvorišta raste tako rastu i troškovi, najviše zbog masovnih izračunavanja i međusobne komunikacije. Nije poznato da li čvorište u sistemu automatski obaveštava sva druga čvorišta o napadu čim ga prepozna ili postoji neki vremenski interval na koji se šalju obaveštenja. Biće

razmotren slučaj trenutnog uzbunjivanja sistema od 1000 čvorišta. Kada se dogodi napad, čvorište koje prepozna taj napad će uzbuniti svih ostalih 999 čvorišta, čime se generiše velika količina saobraćaja zbog međusobne komunikacije. Napadači mogu iskoristiti „širenje uzbune“ kao način da poremete mrežu. Napadači se mogu namerno ponašati kao uljezi u različitim vremenskim intervalima, čime određena čvorišta detektuju uljeze i obaveštavaju o tome sva druga čvorišta, čime se značajno povećava mrežni saobraćaj. U slučaju da čvorište ne obavesti odmah druga čvorišta, raste nekonzistentnost u sistemu. Čvorište koje prepozna napad ima saznanja o tome, dok druga čvorišta i dalje nemaju saznanja o tom događaju. Svako čvorište u sistemu ima sopstvenu bazu podataka gde se nalaze informacije o prethodnim napadima.

Kako bi prepoznao napade, ovaj sistem koristi tehnike prepoznavanja koje su zasnovane na znanju i na ponašanju. Tehnike zasnovane na znanju ne mogu prepoznati nove napade, jer prepoznavanje zavisi od predefinisanih pravila. Ali, ovo ograničenje se može ublažiti preko konstantnog ažuriranja lokalnih baza podataka sa informacijama o novim napadima. Za tehnike zasnovane na ponašanju, sistem koristi neuronsku mrežu za prosleđivanje obaveštenja (FFANN). Ovaj sistem sa neuronskom mrežom nije od velike koristi u početnoj fazi korišćenja zbog male količine podataka sa kojim bi radila. Međutim, kako vreme prolazi i količina podataka se povećava tako se rezultati poboljšavaju.

4.4.4. DCDIDP

Sledeći sistem koji se opisuje jeste *Distributed, Collaborative and Data driven Intrusion Detection and Prevention* (DCDIDP) sistem [74]. Kod ovog sistema kreira se jedinstvena baza podataka koja se koristi za detekciju napada. Detekciju napada vrši ID prevencioni modul. Arhitektura DCDIDP sistema ima tri nivoa:

- mrežna arhitektura,
- arhitektura hostova i
- globalna infrastruktura.

Mreža i hostovi imaju sopstvenu lokalnu bazu podataka koja se sastoji od pravila i polisa i doprinose jedinstvenoj bazi podataka. Jedinstvena baza podataka može deliti svoja saznanja o napadima sa drugim *cloud* sistemima. Osnovne karakteristike ovog sistema su:

- distribuiranost (polise se distribuiraju između hostova),
- kolaborativnost (hostovi su uvek sinhronizovani zahvaljujući konstantnoj međusobnoj razmeni informacija),
- upravljanje podacima (dinamička evaluacija pravila i pristupnih lista).

DCDIDP se može implementirati kod IaaS, PaaS i SaaS platforma i pruža efikasnu zaštitu od upada u sistem. Prethodno pomenuta saradnja između različitih *cloud* sistema zahteva posebno upravljanje i međusobno poverenje, što trenutni sistem ne obezbeđuje.

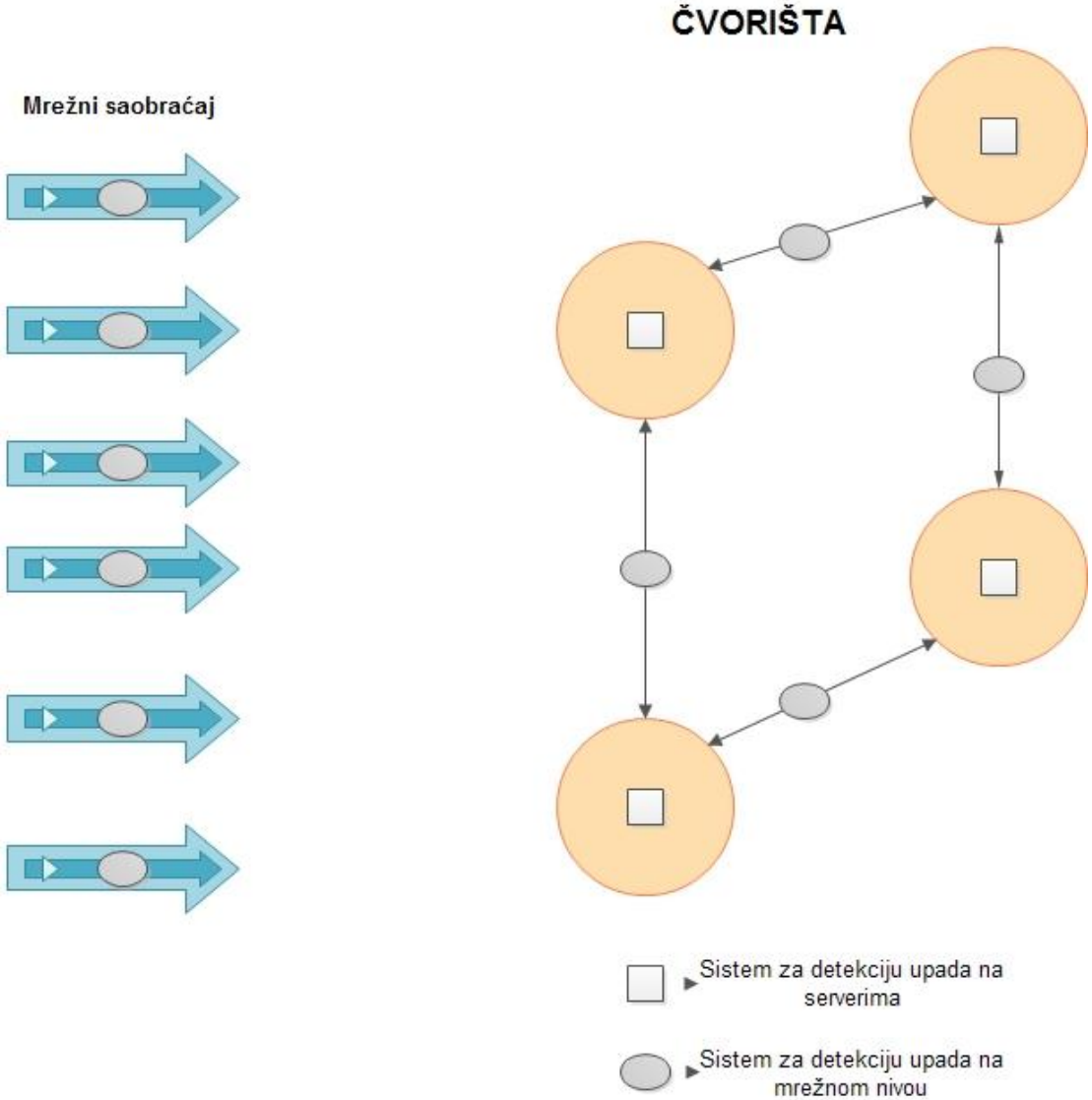
4.4.5. DIDMA

Distributed Intrusion Detection system using Mobile Agents (DIDMA) [75] predstavlja distribuirani sistem za prepoznavanje napada korišćenjem mobilnih agenata. Prednost ovog sistema leži u tome što izvršava decentralizovanu analizu podataka upotrebom mobilnih agenata, što ga čini skalabilnim sistemom. DIDMA koristi platformski nezavisne komponente, što je prednost u odnosu na prethodno opisane sisteme koji koriste komponente sa specifičnom platformom [76]. Sistem se sastoji iz četiri zasebne komponente:

- IDS kontrolni centar (IDS CC),
- agencija,
- aplikaciono specifični statički agent detektor,
- specijalizovani istražni mobilni agent.

IDS CC predstavlja centralni deo u upravljanju sistemom.

Na slici 4.2 je prikazana struktura IDS sistema, koja se sastoji od NIDS-a i HIDS-a. Svaki IDS zahteva komponente kao što je širenje uzbune, analizator, tehnika za prepoznavanje sumnjivog ponašanja i slično. U praksi postoje različita rešenja kojima se nastoji da sistem ima što manje komponenti a što bolje performanse. Najveći izazov kod razvoja ovih sistema jeste heterogenost infrastrukture i upotreba efikasnih komunikacionih protokola.



Slika 4.2 - Osnovna infrastruktura sistema za detekciju napada (IDS)

4.4.6. Autonomni sistemi

Autonomni sistem predstavlja vrstu sistema za detekciju upada koji radi sa predefinisanim pravilima. Na osnovu ovih pravila sistem se automatski konfiguriše, oporavlja, optimizuje i čuva, čime se smanjuje uticaj čoveka na funkcionisanje sistema. Pravila su određena metodama upravljanja ili veštačkom inteligencijom. Bez autonomnog računarstva nemoguće je upravljati distribuiranim sistemima sledeće generacije.

Jedan od takvih sistema je *Security audit as a Service* (SaaS) [77]. To je inteligentni autonomni sistem namenjen za detekciju incidenata i napada. Za ovog agenta se daje pretpostavka da je svestan toka instanci u razvijenom *cloud* sistemu. SaaS prikuplja podatke direktno sa izvora, vrši analizu toga, prikuplja informacije i potom ih distribuira. Centralni

deo sistema je zasnovan na analizi podataka koji se vrši na nivou ugovora servisa sigurnosti (SSLA). SaaS prepoznaje tri osnovna problema *cloud* platforme:

- Zloupotreba *cloud* resursa,
- Nedostatak bezbednosnog praćenja *cloud* infrastrukture,
- Loša izolacija deljenih resursa.

SaaS koristi veliki broj senzora pomoću kojih prepoznaje mnoge promene u sistemu. Ovi senzori primaju uputstva i bezbednosne polise od SSLA. Iako je SaaS autonomni agent, njegova bezbednosna politika je ipak zasnovana na predefinisanim pravilima, pa se time prepoznavanje napada ograničava na već poznate načine napada na sistem. SaaS ima veliki broj agenata i funkcija čime se stvara veliki saobraćaj u mreži zbog međusobne komunikacije i povećavaju troškovi procesorskih resursa.

Još jedan koncept potpuno autonomnog sistema predstavljen je u [78]. Ovaj sistem poseduje mogućnost samokonfigurisanja, samoizlečenja, samooptimizacije i samozaštite.

Potpuni autonomni sistem u svojoj arhitekturi mora posedovati autonomnog menadžera, koji je sposoban da pravi i izvršava planove implementacije. Plan bi trebalo da bude zasnovan na primljenim i poslatim informacijama.

Potpuno autonomni sistem predstavljen je u [79]. Arhitektura ovog sistema je zasnovana na IaaS platformi, a kontrola čvorišta je potpuno autonomna. Ova arhitektura koristi informacije koje dobija od čvorišta u realnom vremenu. Polise upravljanja su automatizovane i decentralizovane sa glavnog na sva sporedna čvorišta. Sistem ima nekoliko komponenti:

- *Cloud* kontroler – komponenta koja određuje koja virtuelna mašina odgovara svakom pojedinačnom klijentu,
- *Cloud* agent – inteligentni softver koji odgovara na upite *cloud* kontrolera. Pitanja se odnose na dostupne konfiguracije virtuelnih mašina za određeni period iznajmljivanja tih mašina,
- Repozitorijum virtuelnih mašina – interfejs kojim se upravlja virtuelnim mašinama i istovremeno kreira URL za svaku pokrenutu virtuelnu mašinu.

Cloud agent se sastoji iz nekoliko delova, kao što je menadžer zahteva, menadžer polisa, menadžer sposobnosti i skladištenja podataka. Ključni element zbog koga se ovaj sistem razlikuje od drugih IaaS sistema jeste što sadrži decentralizovano upavljanje polisama, dok ostali sistemi imaju *master-slave* arhitekturu koja može predstavljati usko grlo sistema. Kod ovog pristupa, ako se nešto dogodi sa master elementom, sistem može otkazati delimično ili čak u potpunosti. Ovakav problem se rešava upravo decentralizovanim pristupom. Polise se prenose sa glavnog na lokalna čvorišta. Zbog brzih promena u *cloud* arhitekturi, neophodno je da decentralizacija bude potpuno autonomna. Decentralizacijom upravljanja polisama se

postiže povećana pouzdanost i bezbednost, jer ako je jedno čvorište ugroženo, ostala čvorišta mogu biti nesmetano kontrolisana.

Slična arhitektura sistema zaštite je opisana u [80]. Naime, autori su razvili sistem za dinamičko i autonomno upravljanje resursima na *cloud* platformi uz uslove da je sistem moguć, skalabilan i fleksibilan. Predstavljen je pristup koji radi po principu decentralizacije zadataka od strane agenata. Sistem ima dve osnovne karakteristike:

Upravljanje resursima se deli na nezavisne zadatke i svaki zadatak izvršava autonomni agent i konfigurisanje vrši agent preko analiza odluka upotrebom PROMETHEE metode [81].

Simulacija sistema dokazuje postizanje skalabilnosti kroz distribuirani pristup čime se smanjuje kompleksnost izračunavanja. Ovakav sistem je moguće upotrebiti kod velikih data centara.

Dinamička alokacija resursa obezbeđuje fleksibilnost kroz sposobnost dodavanja ili menjanja konfiguracija.

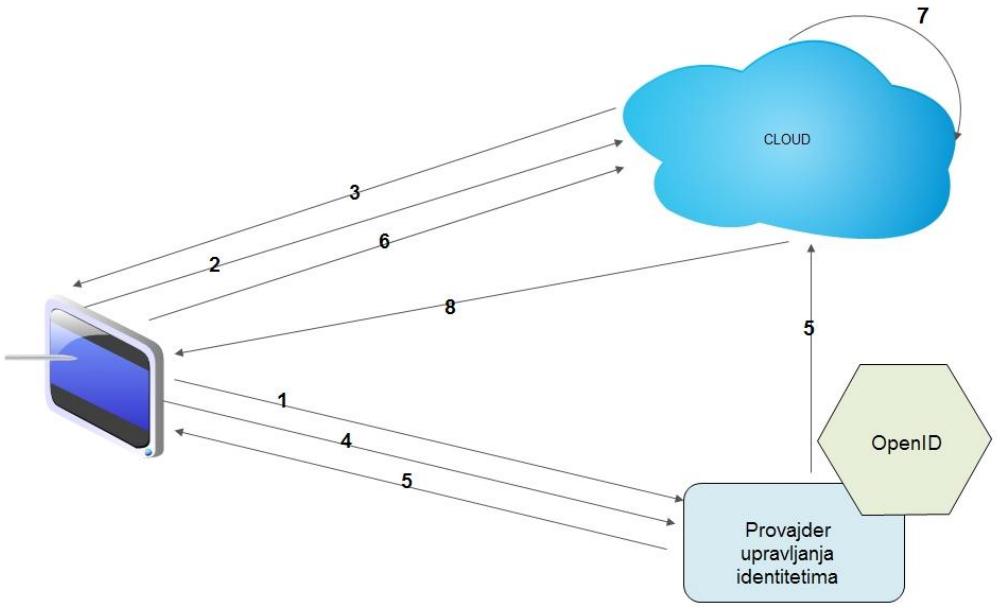
Ipak, sistem nije potpuno automatizovan, jer zahteva ručno postavljanje veličine hard diska i memorije na osnovu procene utroška.

4.5. Sistem federalnog upravljanja identitetima

Osnova bezbednosti kod *cloud* sistema je usko povezana sa identitetima koji se koriste za pristup *cloud* infrastrukturi. Komponenta za upravljanje identitetima (Identity management - IDM) održava integritet identiteta kroz njihov životni ciklus kako bi bili dostupni servisima, uz očuvanje bezbednosti i privatnosti [82]. Koncept federalnog upravljanja identitetima (Federated Identity Management - FIM) opisuje upravljanje identitetima omogućujući subjektu da uspostavi vezu između njegovih identiteta, gde svaki od njih može biti iskorišćen za različite servise. Uspostavljanje logičke veze između identiteta naziva se federacija identiteta. Federacija predstavlja grupu organizacija koje su ostvarile međusobno poverenje kako bi uspostavile poslovnu saradnju. Proces jednog prijavljivanja za više različitih sistema odjednom predstavlja primer federalnog identiteta [83]. Osnovna mana ovakvog koncepta jeste mnogo veća potencijalna šteta u slučaju kompromitovanja sistema za prijavljivanje. Drugi problem jeste nedostatak mehanizma za dinamičku federaciju kod FIM sistema [84].

Na slici 4.3 prikazana je funkcionalnost IDM sistema: prijava na sistem, zahtevanje aplikacije ili podatka, zahtevanje tokena radi ID verifikacije, generisanje tokena, provera i na kraju slanje zahteva za pristup aplikaciji/podacima iz cloud sistema. Opis je dat u koracima. U prvom i drugom koraku korisnik se prijavljuje na sistem upotrebom svojih pristupnih parametara i istovremeno zahteva pristup aplikaciji ili podacima. U trećem koraku *cloud*

potražuje token od sistema zaštite za određenog korisnika radi dalje autentifikacije. U četvrtom koraku korisnik zahteva token od IDM sistema. IDM generiše token i deli ga sa korisnikom i *cloud* platformom (peti korak). Korisnik zatim šalje token ka *cloud* platformi kako bi dovršio proces autentifikacije (šesti korak). U sedmom koraku *cloud* upoređuje token korisnika i token IDM sistema. Ako se tokeni podudaraju *cloud* dozvoljava pristup korisniku.



Slika 4.3 - Sistem federalnog upravljanja identitetima

Postoje tri priznata standarda za generisanje tokena u IDM sistemu. To su:

- SAML,
- OpenID i
- Information Card.

Kako bi se implementirala provera identiteta, SAML se sastoji od skupa tehničkih standarda. Uz pomoć SAML standarda implementiran je sistem jedinstvenog prijavljivanja na *cloud* platforme. Primenjuje se kod mnogih sistema jer primenjuje striktne bezbednosne zahteve. OpenID je sličan SAML standardu, ali ima manji set funkcija uz jednostavnije izražavanje podataka o identitetu. Kod ovog standarda, identiteti su predstavljeni kao skup kartica sa različitim podacima, koji zajedno daju potpun identitet.

U radu [85] se predlaže autorizacioni sistem za *cloud* okruženje, korišćenjem SAML standarda. Predlaže se pristup sistemu bez upotrebe treće strane. Sistem se sastoji iz četiri komponente:

1. Servis za manipulaciju zahtevima – predviđen da autentifikuje korisnika i izda token korisniku.

2. Kontrolor atributa – dizajniran za sprovođenje zahteva za atribute softverskih proizvoda i za primenu odgovarajuće politike privatnosti.

3. Direktorijum servis – davanje prostora korisniku.

4. Mehanizam autentifikacije – predviđen za autentifikaciju korisnika preko korisničkog imena i lozinke.

Ovim se pruža dobra autorizacija ali bez autentifikacije. Nakon autentifikacije korisnika, nije jednostavno ukrasti identitet tokom sesije. Ipak, sistem ne obezbeđuje mehanizam za obezbeđenje legitimiteta osobe povezane na sistem. Dakle, bilo koja osoba sa ispravnim korisničkim imenom i lozinkom može pristupiti sistemu. Multiplatformski autorizacioni sistem generiše odvojeni provajder identiteta za svakog korisnika, dok svaki provajder koristi iste polise sigurnosti. Ova arhitektura pomaže korisnicima da menjaju polise po potrebi.

5. ENERGETSKI EFIKASNO ALOCIRANJE RESURSA U CLOUD COMPUTING OKRUŽENJU

Smanjenje potrošnje električne energije predstavlja ključni korak u smanjenju troškova data centara. Porastom popularnosti *cloud computinga* neophodno je ispitati različite metode za smanjenje potrošnje električne energije u *cloud* okruženju. U istraživanjima je različitim analizama utvrđeno da se utrošak energije može smanjiti i do 14%, a ukupni troškovi i do 26% [86].

Kako se uvećava korišćenje virtuelizacionih servisa, *cloud computing* platforme su sve popularnije. Potražnja za *cloud* infrastrukturama kao sto je Amazon EC2 je konstantno u porastu [87], a upotreba privatnih *cloud* sistema je sve popularnija opcija mnogim institucijama.

Virtuelizacija se definiše kao proces apstrakcije originalne fizičke strukture, kao što je hardverska platforma, operativni sistem, uređaji za skladištenje podataka ili drugi mrežni resursi [88]. Dakle, mašine, aplikacije, mrežni resursi i servisi su potpuno odvojeni od fizičkih ograničenja. Svaki fizički računar u *cloud* sistemu može imati više aktivnih virtuelnih mašina, koje se što se korisničke strane tiče ponašaju kao i svaka druga fizička mašina.

Operativni troškovi data centara su svake godine u porastu. Iz ovih razloga, neophodno je razmišljati o načinima smanjenja potrošnje električne energije na svaki mogući način. U ovom poglavlju biće ispitane različite mogućnosti smanjenja utroška energije kroz prilagođavanja polisa upotrebe virtuelnih mašina. Takođe, kao posebna celina, biće navedene mogućnosti uštede energije koja se troši na softver za zaštitu *cloud* platformi, i to kroz njegovu optimizaciju i povećanje efikasnosti.

U radu [89] je analizirano aktiviranje i rad kompjuterskih servisa u klaster računarima, dok se istovremeno vodilo računa o smanjenju potrošnje, i to kombinacijom algoritama minimizacije i skaliranjem voltaže centralnih procesora. U radu [90] autori su se fokusirali na razvoj sličnog sistema kojim se kombinuje raspoređivanje i skaliranje voltaže centralnih procesora. Ovim načinom se smanjuju troškovi, ali potencirano je rešenje trošenja energije na nivou procesora, a ne na nivou celog klastera.

Cloud platforme za upravljanje resursima i virtuelnim mašinama imaju višestruke prednosti:

1. Skalabilnost,
2. Dostupnost,
3. Fleksibilnost i

4. Bolja izolacija resursa.

Imajući u vidu navedene prednosti, postoji veliki interes u istraživanju različitih vrsta virtuelizacije, kao što je virtuelizacija desktop okruženja, servera, aplikacija, skladišta podataka i mreža.

U literaturi postoji više metoda koje su autori koristili kako bi unapredili *cloud* platformu u smislu energetske efikasnosti.

Jedan od načina uštede postiže se konsolidacijom virtuelnih mašina [91]. Na ovaj način doprinosi se smanjenju potrošnje energije u virtuelnom okruženju. To se postiže tako što se maksimizuje broj neaktivnih fizičkih servera konsolidacijom virtuelnih mašina na što je moguće manji broj fizičkih servera. Ovo se postiže trenutnom migracijom virtuelnih mašina uz pomoć sinhronizacije, uz minimalni dodatni utrošak energije. Nedostatak ovog pristupa predstavlja kršenje QoS pravila tokom sinhronizacije i migracije, zbog primetnog smanjenja performansi sistema. Jedan od problema jeste i povećanje opterećenja mreže *cloud* sistema zbog agresivnog pristupa konsolidacije virtuelnih mašina.

U radu [92] autori su pokušali da pronađu rešenje za prethodno opisani problem mrežnog opterećenja tokom konsolidacije virtuelnih mašina. U radu je predložena nova šema postavljanja virtuelnih mašina zasnovana na heurističkom algoritmu iz dve etape. Na taj način bi se optimizovale performanse mreže, uz smanjenje potrošnje energije, čime bi se napravio kompromis između energetske efikasnosti i performansi mreže.

Slični pokušaj smanjenja potrošnje uz upotrebu heurističkog algoritma opisan je i u radu [93].

U radu [94] autori idu korak dalje, pa osim što se razmatra ušteda i efikasno balansiranje na serverima, već se dodatno nastoji uštedeti energija isključivanjem mrežnih elemenata.

5.1. Komponente cloud platformi

Opšte komponente koje svaka *cloud* platforma treba da ima su:

- Hardver i operativni sistem,
- Računarska mreža,
- Hipervizor (Xen, KVM, itd.),
- Kopije virtuelnih mašina,
- Interfejs (alat za određivanje parametara i pokretanje virtuelnih mašina),
- *Cloud* upravljački sistem (neka od *cloud* platformi, kao što je Open Nebula).

Navedene komponente predstavljaju osnovu za konstruisanje sveobuhvatne *cloud* platforme [95].

5.2. Polise za alokaciju resursa

Postoje i drugačiji pristupi, gde se vrši optimizacija procesa alokacije [96]. Ova strategija obuhvata modelovanje *cloud* sistema u smislu problema arhiviranja, sa fizičkim računarima kao arhivama i virtuelnim mašinama kao objektima koji se postavljaju u njih. Upotrebom ove metode pokušana je konsolidacija virtuelnih mašina na što je moguće manji broj fizičkih servera kako bi se minimizovala potrošnja električne energije.

Opisane alokacije resursa se koriste u popularnim *cloud* platformama kao što je OpenNebula i Eucalyptus.

Postoji više polisa koje se primenjuju kao pravilo u alokaciji resursa *cloud* sistema:

1. Round Robin,
2. Striping,
3. Packing,
4. Balansiranje opterećenja slobodnih procesora,
5. Balansiranje opterećenja slobodnih procesorskih jezgra,
6. Potrošnja energije po procesorskom jezgru i
7. Cena koštanja po procesorskom jezgru.

5.2.1. Round Robin

Round Robin predstavlja najjednostavniji sistem alokacije resursa. Za svaku virtuelnu mašinu, vrši se sekvencijalna iteracija kroz dostupne računare dok se ne nađe server koji ima dovoljno resursa za hostovanje virtuelne mašine. Kada dođe do nalaženja, virtuelna mašina se dodeljuje serveru. Za sledeću virtuelnu mašinu, vrši se dalja sekvencijalna iteracija od mesta gde se stalo i ponovo se bira prvi računar koji može opslužiti virtuelnu mašinu. Ovaj proces se nastavlja sve dok se ne alociraju sve virtuelne mašine.

Round Robin metodom se vrši pokušaj podjednakog postavljanja virtuelnih mašina na serverima. To je trenutno podrazumevana polisa raspoređivanja kod Eucalyptus *cloud* platforme.

5.2.2. Striping

Kod ove polise raspoređivanja, za svaku novu virtuelnu mašinu prvo se odbacuju svi fizički serveri koji nemaju dovoljno resursa za smeštanje virtuelnih mašina. Od preostalih servera, traži se onaj server koji ima najmanji broj virtuelnih mašina. Kada se pronađe takav server, vrši se raspoređivanje nove virtuelne mašine na serveru. Striping predstavlja jednu od polisa koja je ugrađena u OpenNebula *cloud* platformu.

5.2.3. Packing

Packing polisa funkcioniše na suprotan način od *Striping* polise. Za svaku novu virtuelnu mašinu, prvo se odbace serveri koji nemaju dovoljno resursa za dodeljivanje novih virtuelnih mašina. Od preostalih servera, traži se onaj koji ima najveći broj virtuelnih mašina kod sebe. Kada se pronađe, virtuelna mašina se dodeljuje tom serveru. Proces se ponavlja dok se ne alociraju sve virtuelne mašine na čekanju. Na ovaj način, pokušana je alokacija što je moguće više virtuelnih mašina na što manji broj fizičkih servera. *Packing* predstavlja jednu od mogućih polisa kod OpenNebula i Eucalyptus platforme.

5.2.4. Balansiranje opterećenja slobodnih procesora

Ova polisa predstavlja napredniju verziju prethodno opisane *Striping* polise raspoređivanja. Za svaku novu virtuelnu mašinu, prvo se odbacuju serveri koji nemaju dovoljno resursa za postavljanje novih virtuelnih mašina. Od preostalih servera, pronalazi se onaj sa najvećim brojem jezgara u procesoru. Kada je pronađen, virtuelna mašina se dodeljuje takvom serveru. Ova polisa teži minimizaciji opterećenja procesora na računarima. To je jedna od opcionih polisa kod OpenNebula *cloud* platforme, gde postoji pod nazivom *Load Aware* polisa.

5.2.5. Balansiranje opterećenja procesorskih jezgra

Srazmerna polisa balansiranja opterećenja predstavlja napredniju verziju prethodne polise. Za svaku novu virtuelnu mašinu prvo se odbacuju računari koji nemaju dovoljno resursa za postavljanje novih virtuelnih mašina. Od preostalih servera, pronalazi se onaj koji ima najveću razmeru slobodnih procesorskih jezgara u odnosu na zauzeta jezgra. Kada se pronađe, vrši se alociranje virtuelne mašine. Ova polisa teži ka minimizaciji opterećenja procesora.

5.2.6. Potrošnja energije po procesorskom jezgru

Ova polisa [97] predstavlja predlog koji teži energetskoj efikasnosti. Za svaku novu virtuelnu mašinu, prvo se odbacuju računari koji nemaju dovoljno resursa da prihvate nove virtuelne mašine. Od preostalih računara, traži se onaj koji će potrošiti najmanje energije po odabranom procesorskom jezgru, na osnovu podataka o naponu svakog računara. Prilikom računanja potrošnje, daje se pretpostavka da se isključuje računar koji trenutno nema virtuelnih mašina kako ne bi trošio električnu energiju i pravio nepotrebne troškove. Ovom polisom se konstantno traže jezgra koja će za zadatu virtuelnu mašinu trošiti što je manje moguće energije, čime se smanjuje ukupna potrošnja sistema.

Ovo je naravno teoretska pretpostavka, jer autori još uvek nisu implementirali predloženi model.

5.2.7. Cena koštanja po procesorskom jezgru

Ova polisa [98] predstavlja predlog koji teži smanjenju troškova. Za svaku novu virtuelnu mašinu, prvo se odbacuju računari koji nemaju dovoljno resursa za dodeljivanje novim virtuelnim mašinama. Od preostalih računara, nalaze se oni koji će napraviti najmanje dodatnih troškova po procesorskom jezgru, i to na osnovu cene koštanja električne energije i potrebnog napajanja za ciljani računar. Kada se pronađe računar koji ispunjava prethodno navedene zahteve, koristi se za aktiviranje sledeće virtuelne mašine.

Kao i u prethodnom pristupu, ovo je još jedan od predloga autora, ali bez konkretne implementacije.

5.3. Energetska efikasnost hipervizora

Kako su *cloud computing* data centri vremenom sve veći, tako se sve više pažnje posvećuje većoj energetskoj efikasnosti takvih centara. Iz tog razloga, virtuelizacija predstavlja opciju kojom se mogu maksimalno iskoristiti kapaciteti aktivnih servera tako što će na jednom serveru biti aktivno više virtuelnih mašina.

U ovom odeljku biće analizirane Xen, KVM i VMWare ESXi hipervizori, kao i njihov uticaj na efikasnost virtuelizacije. Dalje, biće upoređen njihov uticaj na konačnu potrošnju servera kako bi se izveo zaključak koja od ovih tehnika virtuelizacije daje najbolje rezultate.

Mnoge kompanije i organizacije danas imaju velike koristi od *cloud computing* platformi. Njegova korisnost se proširila na gotovo sve oblasti istraživanja i primene. Zbog toga se u ovu oblast ulaže sve više sredstava i proširuju postojeći kapaciteti. Takođe, proširuju se i istraživanja koja se odnose na što veću energetsku efikasnost samih data centara, jer je njihov udeo u potrošnji električne energije značajan i dostiže oko 1.3% ukupne svetske potrošnje električne energije [98]. Na hardverskom nivou, istražuju se nove arhitekture u smislu ubrzanja (Tesla K20, Intel Phi) i proizvodnje procesora koji troše malu količinu energije u odnosu na svoj radni učinak (ARM). Na srednjem nivou, između softvera i hardvera, pojavljuje se virtuelizacija kao napredni pristup izvršavanja više aktivnih računara – instanci, na jednoj mašini – serveru.

Xen hipervizor se koristi za virtuelizaciju resursa i upravljanje virtuelnim mašinama. On predstavlja osnovu brojnih aplikacija, kako komercijalnih tako i onih otvorenog koda [99]. Sastoji se od: Xen hipervizora, CREDIT raspoređivača, Domen 0 gosta (Dom 0) i Domen U

gosta (DomU) koji podržava pokretanje gostujućih sistema uz paravirtuelizaciju ili punu virtuelizaciju. Ovaj hipervizor predstavlja softverski sloj koji može raditi direktno na hardveru, ispod bilo kod operativnog sistema. Odgovoran je za raspoređivanje procesora i deljenje memorije između virtuelnih mašina. Prilikom pokretanja Xena, hipervizor prvi preuzima kontrolu nad sistemom, a potom učitava prvi gostujući OS.

KVM (Kernel-based Virtual Machine) je hipervizor otvorenog koda koji podržava proces potpune virtuelizacije. Sastoji se iz izmenjenog kernel drajvera uključenog u Linux platformu što mu daje sve prednosti Linux kernela i hardverski podržane virtuelizacije. KVM se sastoji iz dva modula: kernel modul i *user space* modul. Kernel modul je drajver koji upravlja virtuelnim hardverom, gde svaka virtuelna mašina ima svoj adresni prostor alociran od strane Linux raspoređivača prilikom pokretanja. *User space* modul se brine o I/O operacijama u virtuelizaciji.

Vmware ESXi predstavlja deo većeg softverskog paketa pod imenom Vmware vSphere. To je hipervizor koji podržava direktnu komunikaciju sa hardverom (takozvani tip 1 hipervizor). Sve virtuelne mašine ili gostujući operativni sistemi se instaliraju na ESXi serveru. Kako bi instalirali, upravljali i pristupali takvim virtuelnim serverima, neophodan je drugi programski paket pod imenom vSphere *client* ili vCenter. Osnova ESXi hipervizora je Vmkernel koji upravlja virtuelnim mašinama i omogućuje im da pristupe hardveru obezbeđenjem raspoređivanja procesora, upravljanje memorijom i obradom podataka [100].

U jednom od istraživanja [101], upoređivan je stepen potrošnje osnovnog sistema prilikom izvršavanja određenih operacija sa stepenom potrošnje prilikom aktiviranja hipervizora. Takođe, poređenje je vršeno i između dve vrste procesora – Intel Xeon i AMD Opteron. Dobijeni rezultati ukazuju na značajnu razliku između rada osnovnog sistema i rada sa hipervizorima. Sama potrošnja energije prilikom promene hipervizora je prilično mala. Najveća razlika je prilikom upoređivanja potrošnje između servera zasnovanih na AMD i Intel platformi. Rezultati pokazuju da je kod AMD platforme potrošnja veća u odnosu na Intel i to u pojedinim situacijama i do 25% [102].

U drugom istraživanju [103], autori su detaljno istražili uticaj Xen i KVM platforme na energetsku efikasnost u *cloud* sistemima. Osnova za istraživanje bila su 3 servera – jedan je radio bez primene virtuelizacionih tehnika, drugi sa Xen hipervizorom, a treći sa KVM hipervizorom. Autori su došli do nekoliko zaključaka, i to:

- Kada su serveri uključeni, ali aktivne virtuelne mašine nisu zauzete, potrošnja se razlikuje od servera do servera.

- Potrošnja osnovnog servera i servera sa Xen hipervizorom i aktivnim virtuelnim mašinama je gotovo identična (razlika je manja od 0.5% u korist fizičkog servera).
- Kod servera sa instaliranim KVM hipervizorom, potrošnja je bila veća za oko 10%, što predstavlja značajno veću potrošnju.

Ovakvo ponašanje KVM hipervizora je najverovatnije prouzrokovano većim zauzećem procesora dok je server u praznom hodu. Druga istraživanja su pokazala i veće zauzeće memorije prilikom rada u odnosu na Xen [104].

Kada višejezgarni server radi sa višeprocesnim aplikacijama, fizička mašina može potrošiti više energije od virtuelizovanih servera. To se dešava zbog loše višejezgarne optimizacije na fizičkoj mašini. Sa druge strane, Xen i KVM mogu distribuirati fizička procesorska jezgra u virtuelna, čime se demonstrira očigledna prednost virtuelizacije u poboljšanju ravnomernog korišćenja resursa [105].

U radovima [106] i [107] autori su upoređivali performanse različitih hipervizora u istom okruženju. Nakon sprovedenih testova, utvrđeno je da se Xen hipervizor pokazao najoptimizovaniji, uz prikazane bolje performanse od hipervizora kao što su VMware ESXi, KVM ili MS Hyper-V hipervizora.

Tabela 5.1 - Prikaz karakteristika skalabilnosti hipervizora

Hipervizor	Xen 4.6	KVM 1.2.0	ESXi 6.0
Arhitektura	IA-32, x86, x64, ARM	ARM, IA-64, PowerPC, S/390, x86-64	X86, x64
Max CPU na VM	512	160	128
Max Host memorija	16 TB	-	12 TB
Max VM memorija	1TB	32 TB	4 TB
3D akceleracija	Da	Da	Da
Licenca	GPL	GPL/LGPL	U privatnom vlasništvu

U tabeli 5.1 prikazana je skalabilnost svakog od pomenutih hipervizora. Njihove mogućnosti se međusobno razlikuju. U zavisnosti od potreba i instaliranih *cloud* platformi, svaki od provajdera se može odlučiti koji od hipervizora će koristiti. Upoređivanjem karakteristika i rada hipervizora u različitim okruženjima, može se zaključiti da se Xen hipervizor pokazao kao najefikasnije rešenje.

Kako bi se ispunile različite potrebe kklijenata, svaki provajder može implementirati više hipervizora, tako da klijenti imaju izbora prilikom konfigurisanja sopstvenog sistema. Danas je neizbežno da, pored osnovnih karakteristika vezanih za skalabilnost hipervizora, provajderi

obraćaju pažnju i na stepen efikasnosti svakog hipervizora. Hipervizor koji ima najoptimalnije performanse utiče i na smanjenje potrošnje energije celokupnog *cloud* sistema, što svakako ne treba zanemariti.

5.4. Uticaj bezbednosnih mehanizama na energetsku efikasnost cloud platformi

Ubrzani rast globalnog skladištenja podataka umanjuje postignutu efikasnost kod praćenja Murovog zakona u računarstvu. Problem efikasnosti se dodatno povećava primenom bezbednosnih mehanizama.

Udvostručavanje tranzistora kod procesora na svakih 18-24 meseca, koje je opisano Murovim zakonom, praćeno je od strane industrije već decenijama unazad. Smanjenje radne voltaže je pomoglo održavanju konzistentnosti tranzistora na jednom čipu, dok se brzina konstantno uvećavala. U poslednje vreme primetno je usporavanje udvostručavanja, tako da se može kazati da je važenje Murovog zakona pri kraju.

Sa druge strane, primećen je eksponencijalni rast u količini podataka iz različitih izvora. Ova velika količina podataka prevazilazi mogućnosti skladištenja i obrade podataka, što dovodi do povećanja neefikasnosti. Ova razlika se znatno uvećava kada se uzme u obzir da je bezbednost nezaobilazni deo implementacije svakog sistema koji utiče na performanse i potrošnju sistema. Dakle, ne samo da sve vrste ličnih uređaja zahtevaju određeni nivo sigurnosti, već postoje i velike količine privatnih podataka u *cloud* sistemima koji zahtevaju zaštitu. Današnji bezbednosni pristupi se obično primenjuju po *ad hoc* sistemu: svaki računarski uređaj je zaštićen nekim hardverskim bezbednosnim implementacijama, kao i antivirusnim i *firewall* programima. Mrežni saobraćaj je obezbeđen različitim sigurnosnim protokolima i softverom za filtriranje spam poruka. Kako obim podataka bude rastao, tako će rasti i udeo u potrošnji energije koja odlazi na bezbednost. Ovi bezbednosni modeli ne mogu biti dizajnirani nezavisno, tako da će u dizajnu sistema značajan deo imati bezbednosne komponente koje moraju biti energetski efikasne.

Iz navedenih razloga, neophodno je sagledati udeo potrošnje električne energije u današnjim sistemima i analizirati ono što je do sada urađeno na ovom polju [108], [109], [110], kao i mogućnostima za dalje unapređenje ove oblasti.

Ono što trenutno najviše može uticati na povećanje potrošnje jeste antivirusni softver, filtriranje spam poruka (u *cloudu*) i bezbednost u računarskim mrežama (bezbednosni protokoli, koji se primenjuju na svaki podatak).

5.4.1. Filtriranje spam poruka

Oko 90% svih poruka elektronske pošte predstavljaju spam (neželjene) poruke. Ako se ne prepoznaju na vreme, ove poruke mogu prouzrokovati dodatne probleme u poslovanju, kao i običnim korisnicima. Zbog velikog obima, filtriranje spam poruka samo po sebi može potrošiti veliku količinu energije. Na svom putu od pošiljaoca do primaoca, spam poruka može proći kroz nekoliko različitih filtera. Prvi filter može biti na serverima globalnih servisa (Yahoo, Google, Outlook), dok poslednji može biti spam filter Internet provajdera korisnika.
Na osnovu određenih istraživanja u [111] moguće je izračunati ukupnu potrošnju energije na određene bezbednosne sisteme koji se koriste u *cloud* sistemima.
Prema izveštaju kompanije McAfee [112], procenjuje se da je u 2008. godini ukupno poslato oko 66 biliona spam poruka, a ukupna potrošnja energije u svetu koja odlazi na spam i sve vezano za spam je oko 33 milijarde kWh. Od toga, procenjeno je da je udeo potrošnje spam filtera negde oko 16%. Za 2013. godinu procenjuje se da je ukupan broj poslatih spam poruka smanjen na oko 44 biliona poruka [113]. Iz toga možemo zaključiti da se ukupna potrošnja energije smanjila na oko 22 milijarde kWh. Smanjenje u odnosu na 2008. godinu je značajno, ali je potrošnja električne energije još uvek velika.
Iz navedenog možemo zaključiti da je potrošnja spam filtera (PSF):

$$PSF = \frac{\%potrošnje * ukupnaPotrošnja}{100}$$

$$PSF = \frac{16 * 22.000.000.000}{100} = 3.520.000.000 \quad \text{kWh}$$

Dalje, iz ovoga možemo izračunati da je ukupna potrošnja po spam poruci (PPP):

$$PPP = \frac{UkupnaPotrošnja}{BrojSpamPoruka}$$

$$PPP = \frac{22.000.000.000}{44.000.000.000.000} = 0.0005kWh,$$

tj. 0.5W po spam poruci.

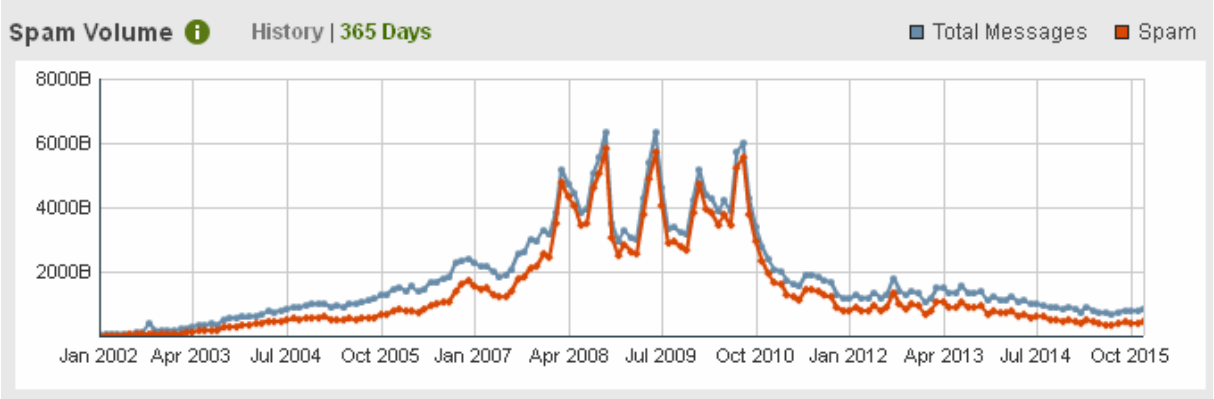
Iz ovog dalje zaključujemo da je potrošnja spam filtera po poruci, po prethodno definisanoj formuli

$$PSF = \frac{\%\,potrošnje * ukupnaPotrošnja}{100}$$

$$PSF = \frac{16 * 0.5}{100} = 0.08W$$

po jednoj spam poruci.

Na sledećoj slici prikazano je kretanje ukupnog broja poslatih spam poruka u poslednjih 13 godina.



Slika 5.1 - Globalni obim poslatih spam poruka [114]

U navedenom grafikonu primećuje se pad broja poslatih spam poruka. Međutim, to ne znači da su provajderi sve uspešniji u borbi protiv spam poruka, već da su se pošiljaoci preorijentisali na socijalne mreže, budući da sve veći broj korisnika interneta koristi upravo socijalne mreže kao što su Facebook i Twitter. Umesto e-mail poruka vrši se slanje instant poruka između korisnika, tako da je za same socijalne mreže izuzetno teško da detektuju i spreče spam poruke.

5.4.2. Antivirusi

Antivirusni programi se već godinama koriste na PC računarima. Poznato je da ovi programi mogu zauzeti značajnu količinu kompjuterskih resursa. Iako je prilično teško odrediti koliki je udeo antivirusnog softvera u ukupnoj potrošnji električne energije, to se može pretpostaviti na osnovu uticaja na performanse sistema. Pošto antivirusni softver vrši praćenje rada operativnog sistema, praćenje rada aplikacija i manipulaciju fajlovima (kopiranje, premeštanje i sl.), njegov uticaj na sistem nije beznačajan.

U tabeli 5.2 su prikazani rezultati istraživanja antivirusa koji ukazuju na zauzeće sistemskih resursa [115].

Tabela 5.2 - Prosečan uticaj antivirusnog softvera na performanse sistema

Ispitivanja	Rezultati
Zauzeće memorije	30 MB
Veličina instalacije	400 MB
Povećanje vremena ponovnog pokretanja operativnog sistema	18.25 s
Početna brzina skeniranja	26.58 MB/s
Standardna brzina skeniranja	150.7 MB/s
Kašnjenje u pokretanju aplikacija	0.46 s

*rezultati mogu variratu u odnosu na korišćeni antivirusni softver i konfiguraciju računara

Kod mobilnih platformi, jedno istraživanje pokazuje da se upotrebom antivirusnog softvera na mobilnim uređajima može povećati potrošnja energije i do 40% prilikom konstantnog pingovanja uređaja (paketi od 1KB, 100 paketa u sekundi) [116].

5.4.3. Bezbednost hardvera

Mnogi sigurnosni mehanizmi su implementirani na hardverskom nivou u vidu primitiva kako bi bili što efikasniji. Kako snaga procesora eksponencijalno raste, tako se stvaraju uslovi za veću razmeru implementacije sigurnosnih mehanizama u sam hardver. Većina prethodnih istraživanja koja se fokusiraju na energetsku potrošnju kompjuterskih uređaja razmatraju samo najveće potrošače, kao što je CPU i memorija. Povećanje kompleksnosti hardverskih sigurnosnih mehanizama ukazuje na to da potrošnja energije na bezbednost treba da bude važan faktor prilikom dizajniranja takvog hardvera [117].

Operativna memorija je jedna od komponenti računarskih sistema koja je najviše istraživana radi primene energetski efikasnih polisa upotrebe. Istraživanje u radu [118] ispituje uvećanu potrošnju energije nakon implementiranja strategija autentifikacije i enkripcije u memoriju kako bi se sprečilo „curenje" podataka. Autori zaključuju da se potrošnja energije uvećava prilikom enkripcije za 1-12%, u zavisnosti od korišćenih mehanizama, dok se prilikom autentifikacije potrošnja uvećava i do 66%.

5.4.4. Bezbednost računarskih mreža

Pored implementiranja bezbednosnih protokola na platformama, još jedan od potrošača energije predstavlja obezbeđenje svakog podatka prilikom slanja kroz računarsku mrežu.

Integritet i bezbednost svih podataka na Internetu obezbeđuje se primenom raznih sigurnosnih protokola, različitim mehanizmima za kriptovanje podataka, digitalnih potpisa i sl.

Jedno od istraživanja prikazuje povećanje utroška energije primenom sigurnosnih protokola [119].

Najkorišćeniji algoritmi zaštite troše oko jednog mikrodžula po bajtu podataka, dok algoritmi koji koriste razmenu ključeva koriste oko 1 milidžula po bajtu podatka. Procenjenjuje se da je potrebno oko 50mJ energije mobilnom klijentu da izvrši komunikaciju od 1MB preko SSL protokola [120].

Ukupno posmatrano, procenjuje se da sigurnosni mehanizmi čine oko 30% ukupne potrošnje energije. Naravno, kako se količina podataka na Internetu konstantno povećava, očekuje se dodatni rast potrošnje energije za zaštitu tih podataka.

6. ENERGETSKI EFIKASNA KLIJENTSKA PLATFORMA NAMENJENA UČENJU

U tradicionalnom režimu učenja zasnovanom na vebu, izgradnja i održavanje sistema se nalaze u unutrašnjosti obrazovne institucije ili preduzeća, što stvara dosta problema, kao što je mnogo potrebnih investicija, ali bez povratka kapitala i bez razvojnog potencijala. *Cloud computing* koji uvodi mehanizam servisa na zahtev može obrazovnoj ustanovi iznajmiti infrastrukturu koja se može koristiti za učenje. Tako se ustanova oslobađa dodatnih troškova i time obezbeuje znatno efikasniji sistem za elektronsko učenje.

Nakon odabira serverskih rešenja, koja mogu biti lokalno rešenje ili implementacija putem *cloud* sistema, postavlja se pitanje odabira klijenata uz pomoć kojih će studenti pratiti kurseve, raditi seminarske radove, testove, polagati ispite...

Tu postoji više rešenja. Cilj je odabrati ono rešenje koje je najekonomičnije za obrazovnu ustanovu. Dakle, oprema mora biti cenovno povoljna i energetski efikasna.

Prvo rešenje jeste zadržavanje tradicionalne tehnologije i nabavka PC računara. Tradicionalni PC uređaji imaju visoke troškove održavanja, pogotovo kod velikih preduzeća kao što su banke, telekomunikacioni operateri i sl. Podaci na računarima se čuvaju na lokalnim diskovima, koji su podložni uticaju virusa. Od savremenih sistema zahteva se visoka pouzdanost i energetska efikasnost. Jasno je da PC računari ne mogu zadovoljiti ove potrebe.

Drugo rešenje jeste primena tankih klijenata. Tanki klijenti predstavljaju energetski efikasno rešenje, ali kako bi radili na predviđen način neophodno je da se oslanjaju na druge serverske ili *cloud* sisteme. Poslednje rešenje jeste upotreba *smartphone* i/ili tablet uređaja. Za njihovu upotrebu u edukativne svrhe neophodna je i određena softverska podrška (aplikacija ili platforma) od strane obrazovne institucije.

6.1. Evaluacija tankih klijenata prilikom korišćenja u desktop Cloud okruženju

U ovom odeljku vršiće se istraživanje tankih klijenata koje mogu uspešno zameniti personalne računare, pogotovo u domenu upotrebe *cloud* servisa. Faktori koji su uzeti u obzir obuhvataju hardver, udaljeni sistem upravljanja, pouzdanost, kompatibilnost platforme za virtuelizaciju i potrošnja električne energije.

Sa razvojem virtuelizacije za *cloud* okruženje sve je šira upotreba tankih klijenata koji zamenjuju personalne računare. Desktop virtuelizacija daje rešenje ovih problema. Krajnji korisnici mogu pristupiti virtuelnom desktop okruženju putem tankih klijenata, koji zahtevaju

manje održavanja od personalnih računara. Pored toga, podaci korisnika se čuvaju na serverima koji imaju visoki nivo redudantnosti. Centralizovano upravljanje virtuelnog desktopa omogućuje brži odgovor IT službe za održavanje sistema.

Kao ulazni deo desktop *cloud* okruženja, tanki klijenti igraju važnu ulogu u iskustvu klijenta, što je važno za evaluaciju karakteristika tankih klijenata. Prvi pristup upoređivanja predstavlja jednostavno upoređivanje performansi različitih klijenata. Međutim, programi za merenje performansi se nalaze na serverima, čime se u stvari mere performanse virtuelnog desktopa a ne tankih klijenata. U radu [120] vršeno je merenje hvatanjem mrežnih paketa između tankog klijenta i njegovog servera tokom standardnog merenja performansi aplikacija.

Da bi se što preciznije izmerile performanse klijenata, mora biti obuhvaćeno više faktora za merenje. Ono što je neophodno obuhvatiti jeste hardver, sistem udaljenog upravljanja, kompatibilnost periferija, performanse, pouzdanost, kompatibilnost virtuelizacione platforme i potrošnja električne energije.

Na tržištu postoji puno modela tankih klijenata raznih proizvođača, kao što su Dell, HP, Start, Cloud Times, Huawei i ZTE.

Tanki klijenti su dizajnirani da budu laki, da ne zauzimaju puno prostora i da imaju skromnu hardversku konfiguraciju. Obično su zasnovani na x86 i ARM arhitekturi. Postoji mnogo proizvođača ARM procesora, zbog svog načina licenciranja, ali neki od njih koji se najviše koriste kod tankih klijenata su Huawei, Armlogic i Maxcell. Instalirani operativni sistemi su obično neki od prilagođenih Windows i Linux platformi. Različita hardverska i softverska rešenja otežavaju kompatibilnost ovih klijenata. Iz tog razloga, potreban je jedinstveni sistem upravljanja u okviru službe IT administracije.

Dalje, tanki klijenti mogu biti prilagođeni različitim platformama za virtuelizaciju. VMware, Citrix Xen i Hyper-V predstavljaju trenutno najprihvaćenija rešenja u toj oblasti. Pored njih, postoje i druge instance određenih proizvođača, kao što su Oracle VDI, Redhat VDI i ZTE iRAI. Kako bi se klijenti povezali sa serverom, neophodno je na lokalnim operativnim sistemima instalirati odgovarajući softver. Zbog navedenih specifičnosti, neophodno je ispitati i uporediti kompatibilnost i pouzdanost navedenih sistema.

Tanki klijenti se koriste u različite svrhe – za kancelarijske poslove, *call* centre, servise i u različitim centrima za održavanje i podršku.

Tanki klijenti se upoređuju na osnovu svojih dimenzija, načina i mogućnosti hlađenja uređaja, eksternih interfejsa, konfiguracije hardvera (CPU, memorija, skladištenje podataka, grafička i mrežne kartice).

Udaljeno upravljanje predstavlja osnovnu razliku u odnosu na tradicionalne personalne računare. Sistem udaljenog upravljanja se koristi za konfiguraciju i promenu softvera tankih klijenata. Sistem udaljenog upravljanja bi trebalo da obezbedi funkciju automatskog nalaženja tankih klijenata i dodeli ih određenim radnim grupama na osnovu utvrđene strategije, npr. po IP adresi ili korisničkom imenu. Jedna od dodatnih potreba jeste udaljeno uključivanje i isključivanje uređaja. Kako bi se osigurala bezbednost informacija, ovaj sistem bi trebalo da ima mogućnost kontrole pristupa interfejsima (prilikom priključenja USB memorija i sl.).

Kompatibilnost perifernih uređaja se ispituje priključenjem opreme kao što su skeneri, štampači, mikrofoni, slušalice, kamere, itd.

Tanki klijenti predstavaljaju ulaz ka virtuelnom desktop okruženju, i većina posla se obavlja od strane virtuelizovanog okruženja na serverskoj strani. Međutim, prilikom rada sa multimedijalnim sadržajem, virtuelni procesor tokom dekodiranja video sadržaja i slanja slika troši prilično serverskih i mrežnih resursa, a kao rezultat dobija se vrlo loš vizuelni prikaz. Iz tog razloga, razvijen je protokol multimedijalne redirekcije (multimedia redirection - MMR), uz pomoć koga se transportuje originalni kompresovani multimedijalni sadžaj od servera do klijenta, a zatim se dekodira i izvršava lokalno. MMR omogućuje korisnicima da imaju kvalitetan audio i video prijem.

Alati za testiranje performansi tradicionalnih računara nisu pouzdani ni primenjivi na tanke klijente. Ako se softver pokreće na virtuelnom desktopu, najveće opterećenje će preuzeti server dok će uticaj na klijente biti minimalan. Pokretanjem alata na lokalnom operativnom sistemu se postiže željeni efekat. Performanse klijenata se zatim mogu proveriti pokretanjem video materijala u formatu sa visokom rezolucijom, kontinuirano u dužem vremenskom periodu.

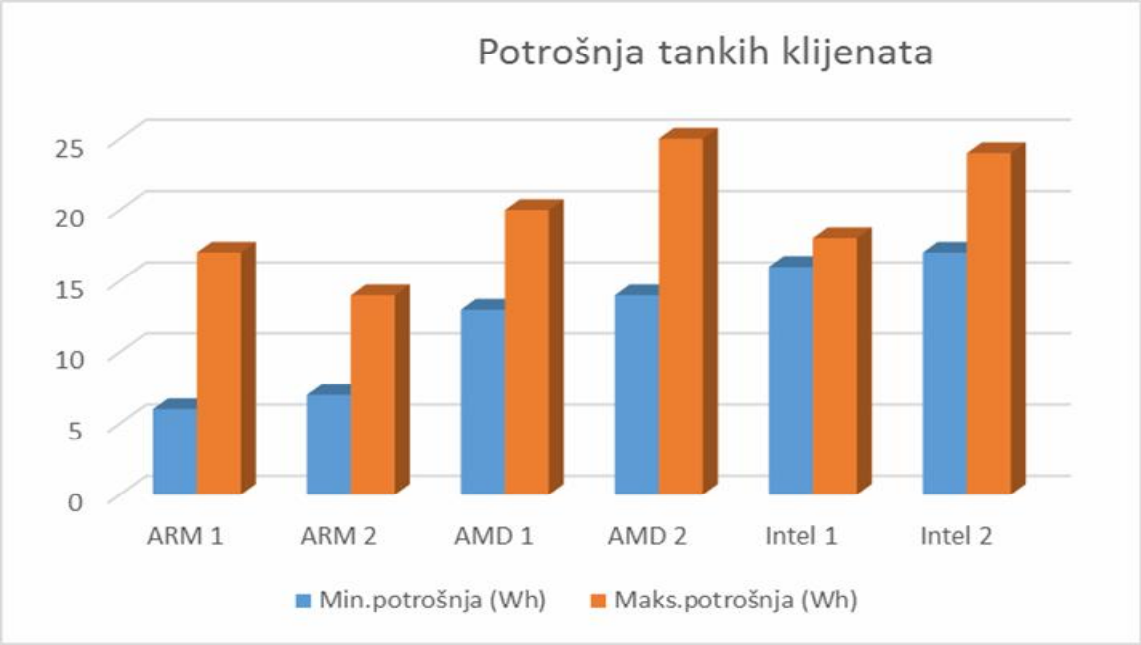
Kompatibilnost virtuelizacione platforme tankih klijenata predstavlja bitan deo evaluacije sistema, koji se odnosi na virtuelizacionu platformu, virtuelne desktop klijente, lokalni operativni sistem i konfiguraciju hardvera.

Niska potrošnja energije je jedna od najbitnijih karakteristika tankih klijenata. U realnom okruženju, klijenti se mogu naći u različitim situacijama. Kod potrošnje energije razlikujemo tri različita stanja, kao što je status isključen, povezanost na virtuelni desktop i prikaz video materijala, gde se vrši merenje potrošnje.

Nakon utvrđivanja načina upoređivanja klijentskih rešenja, može se pristupiti njihovom upoređivanju. Kod pitanja energetske efikasnosti najvažnija stvar jeste ukupna potrošnja u radu tankih klijenata (tabela 6.1). Na slici 6.1 prikazana je minimalna i maksimalna potrošnja za tanke klijente različitih proizvođača.

Tabela 6.1 - Analiza potrošnje tankih klijenata

Arhitektura	ARM 1	ARM 2	AMD 1	AMD 2	Intel 1	Intel 2
Min.potrošnja (Wh)	6	7	13	14	16	17
Maks.potrošnja (Wh)	17	14	20	25	18	24



Slika 6.1 - Uporedni pregled potrošnje tankih klijenata

Na testu se uočava da klijenti čija je arhitektura zasnovana na ARM procesorima troše ispod 7.5Wh. Ovi klijenti su znatno kompaktniji od drugih, troše malo energije i imaju nisku disipaciju toplote. Oni klijenti koji nisu zasnovani na ARM arhitekturi, uz dobru energetsku optimizaciju, dostižu potrošnju do 15Wh, dok ostali troše više od 15Wh. U obzir nije uzeta i (neophodna) potrošnja monitora, već samo potrošnja tankih klijenata.

6.2. Analiza energetski efikasnih klijenata namenjenih učenju

Iako u literaturi postoji priličan broj radova koji analiziraju rešenja kojima bi povećali energetsku efikasnost [121], [122], [123], kako u opštim *cloud* modelima, tako i kod primene u edukaciji, koliko je autoru poznato, ne postoji sveobuhvatni pregled i analiza energetski efikasnih rešenja koji obuhvataju različite klijentske uređaje (PC, *Thin* i *smartphone*), pogotovo u edukaciji. Kako bi bilo moguće izračunavanje količine utrošene električne energije, prvo se mora utvrditi kolika je dužina trajanja jednog kursa u našim uslovima. Naime, kada je potrebno izračunati potrošnju energije i odrediti potencijalnu energetsku efikasnost mogućih rešenja, određivanje potrošnje na mesečnom ili godišnjem nivou nije izvodljivo. Dakle, postoje periodi kada se kompjuterski resursi uopšte neće koristiti (kada se ne odvijaju nastavne aktivnosti). Najtačnije moguće rešenje koje bi prikazalo realnu potrošnju i efikasnost se dobija samo ako se tačno odredi koliko se vremenski dati resursi koriste. Ako znamo da po planu i programu jedan kurs (predmet u semestru) traje 15 nedelja, i da se u svakoj nedelji održi u proseku po 4 časa u trajanju od po 45 minuta iz tog predmeta, onda dobijamo da su računarski resursi u funkciji:

$$\frac{15*4*45}{60} = \frac{2700}{60} = 45$$

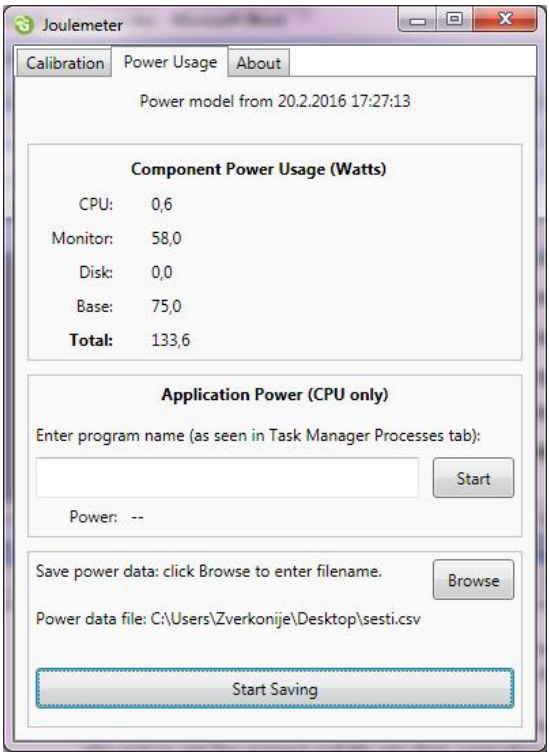
sati, po jednom jednosemestralnom predmetu.

Nakon što znamo koliko su klijenti aktivni, moguće je napraviti projekciju potrošnje 100 klijenata i na taj način pronaći najefikasnije rešenje.

Pored toga, bilo je neophodno kreirati okruženje za novu mobilnu platformu. Iz tog razloga razvijena je platforma pod imenom mTester. Prilikom evaluacije elemenata potrebnih za povezivanje sistemskog dela mTester platforme, moralo se voditi računa o više stvari. Osnovni uslov jeste mogućnost efikasnog povezivanja sa klijentskom stranom platforme koja je zasnovana na Android operativnom sistemu. Takođe, sistemski deo bi trebao biti zasnovan na platformama koje su jeftine za postavljanje i održavanje. Iz tog razloga, izabrano je *open-source* rešenje u vidu Apache servera i MySQL baze podataka.

U tabeli 6.2 prikazano je upoređivanje predloženih rešenja za elektronsku učionicu. Dato je poređenje računara starije generacije (PC), zatim savremenijih rešenja u vidu tankih klijenata i *smartphone*/tablet uređaja. Merenje potrošnje PC-ja i tankih klijenata obavljeno je kombinacijom uređaja za neprekidno napajanje sa sopstvenom baterijom UPS (Uninterruptible power supply) APC BR1000G [124] i softverskog rešenja za procenu

potrošnje na osnovu senzora i modela potrošnje pod nazivom Joulemeter [125]. Joulemeter je deo Microsoft Research projekta koji je iskorišćen kao osnova za primenu sličnih modela za procenu potrošnje energije u okviru paketa Visual Studio. Joulemeter može proceniti potrošnju energije računara i pojedinih komponenti računara (slika 6.2), virtuelne mašine, pa čak i pojedinog programa ili procesa konvertovanjem zauzeća resursa u realno korišćenje energije na bazi realističnog modela potrošnje zasnovanog na automatskom učenju. Joulemeter omogućuje čitanje podataka sa spoljnih senzora, a može se oslanjati i na očitavanja potrošnje baterije kod laptop računara. Još jedna od prednosti ovog alata je mogućnost čuvanja trenutnih vrednosti u csv formatu, što dalje omogućuje prikaz rezultata u vidu grafikona (slika 6.3).



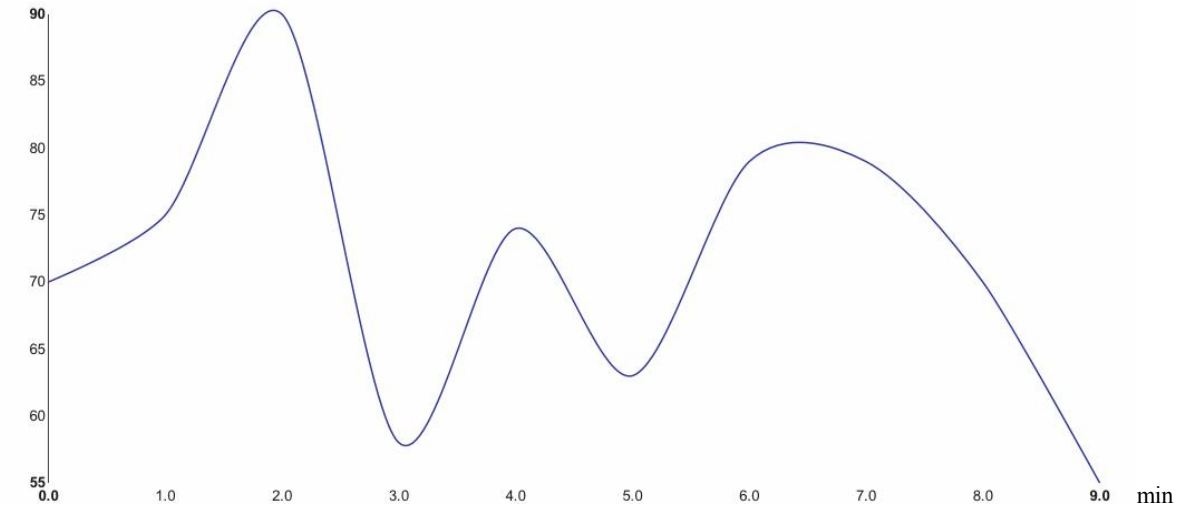
Slika 6.2 - Prikaz potrošnje komponenti računara pomoću softverskog alata Joulemeter

Za smartphone uređaje bilo je neophodno pronaći softversko rešenje koje je zasnovano na Android platformi. Iako postoji puno programa koji prate potrošnju programa i procesa, kao i njihov udeo u potrošnji baterije, ne postoji veliki broj relevantnih aplikacija koje mogu interpretirati potrošnju energije pojedinih hardverskih komponenti u realnom vremenu, njihovu ukupnu potrošnju, kao i tekstualni i grafički prikaz izmerenih vrednosti. Jedno od najpotpunijih rešenja koje ispunjavaju navedene uslove je PowerTutor [126]. Pomoću ovog alata moguće je izmeriti potrošnju i potom sačuvati dobijene vrednosti tokom upotrebe aplikacije za mobilno učenje mTester u uslovima praćenja prenosa video predavanja preko mreže, pregleda ostalih materijala i rešavanja testova.

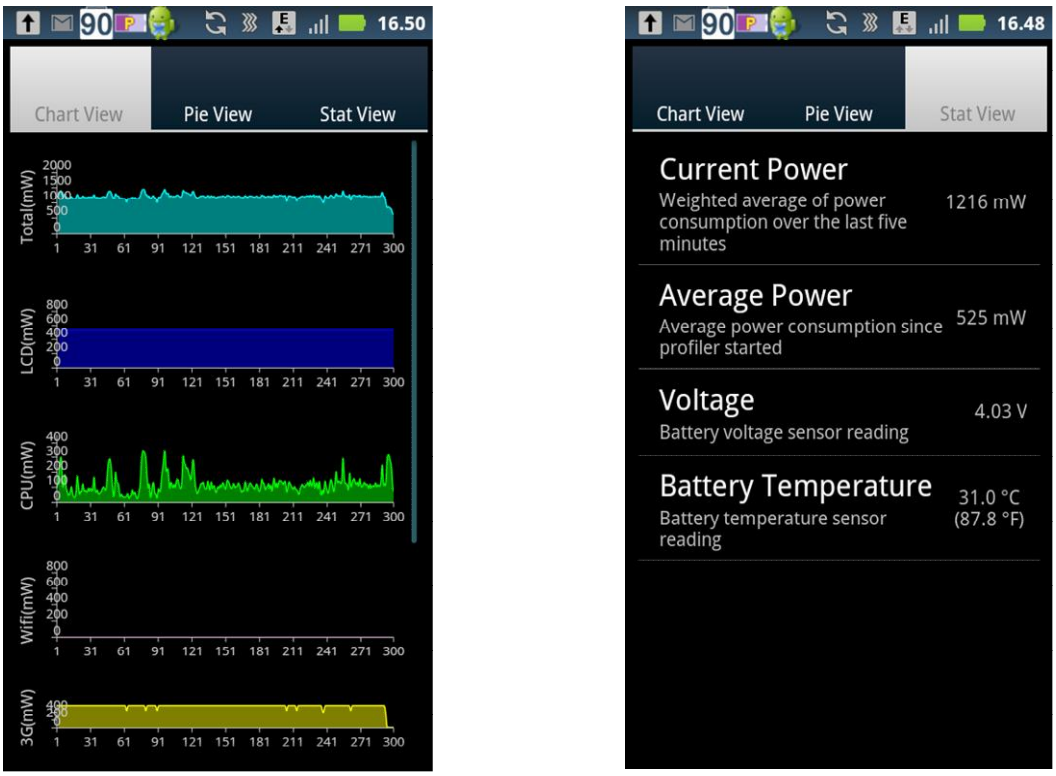
Tabela 6.2 - Analiza potrošnje klijenata i servera namenjenih elektronskoj učionici

Uređaji	PC	Tanki klijenti	Smartphone uređaji
Broj klijenata	100	100	100
Potrošnja klijenta* (W)	129.3	39	1.28
Uk.potrošnja (kWh) – (p.klijenta*br.klijenata)*45 /1000	581.85	175.5	5.76
Ukupna potrošnja za godinu dana (kWh) (svi predmeti na stud.programu RI)	20352.5	6142.5	201.6
Godišnja emisija štetnih gasova CO_2 (0.7kg/kWh)	14246,75	4299,75	141,12

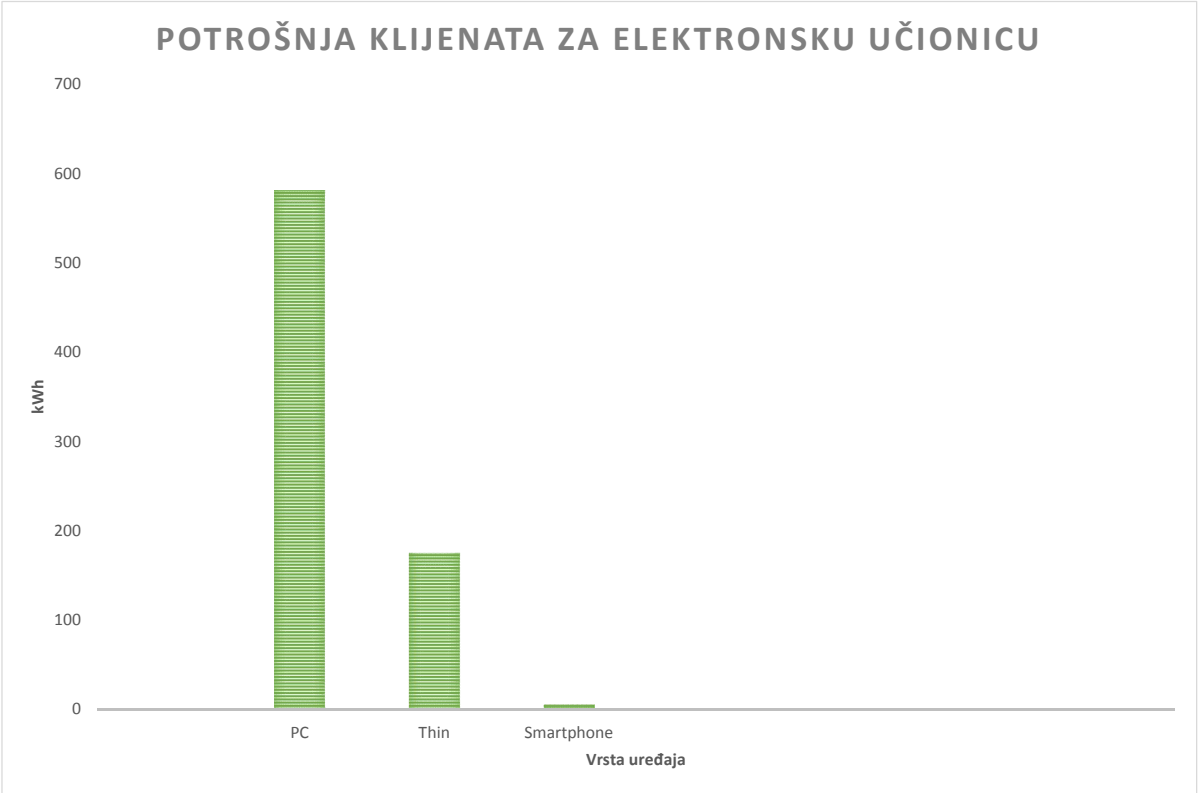
*Prosečna potrošnja prilikom praćenja video predavanja i rešavanja testova



Slika 6.3 - Potrošnja električne energije PC računara (bez monitora)



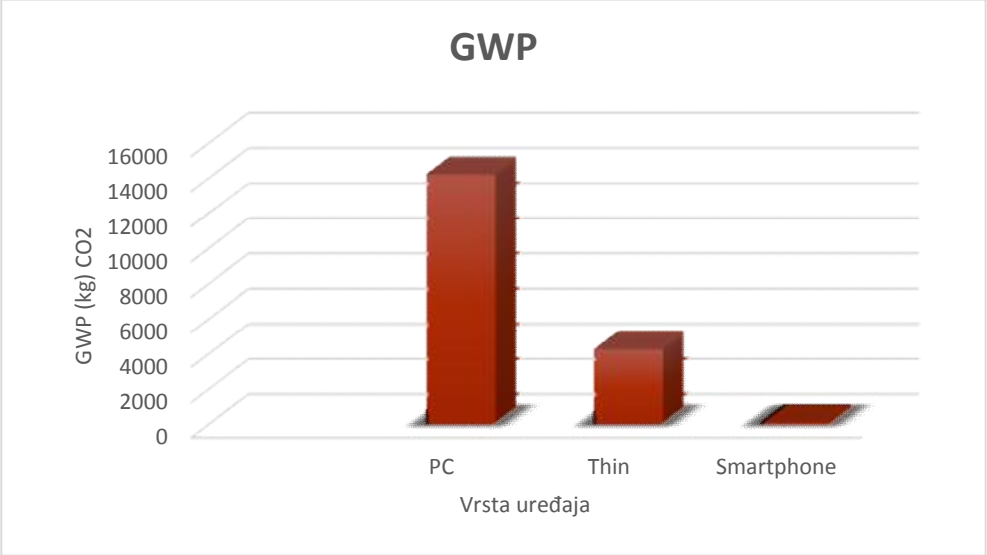
Slika 6.4 - Potrošnja energije po komponentama Android uređaja (levo) i prosečna potrošnja (desno)



Slika 6.5 - Uporedni pregled potrošnje električne energije klijenata za elektronsku učionicu

Iz navedenih pokazatelja primećuje se da je primena PC računara potpuno neisplativa, i kao takvu je ne treba uzeti u obzir za bilo kakvu narednu implementaciju u elektronskoj učionici ili računarskoj laboratoriji. Upotrebom tankih klijenata dobijamo optimalne rezultate.

Potrošnja nije velika, a pruža određeni komfor upotrebe kao i na PC računarima. Upotreba mobilnih uređaja daje najbolje rezultate. Potrošnja električne energije je u ovom slučaju najmanja i najefikasnija (slika 6.4 i 6.5). Ali, tu ostaje problem kompatibilnosti sa određenim protokolima koji se obično koriste prilikom virtuelizacije i povezivanja sa *cloud* sistemima. Pored toga, komfor može biti problem, jer su ovakvi uređaji znatno manjih ekrana od uporednih konfiguracija. Zato je primena tablet uređaja poželjna, ali uz određenu softversku implementaciju koja će se koristiti u edukativne svrhe. Iz prikazanih rezultata se može razmatrati uticaj na klimatske promene ispitivanjem globalnog potencijala zagrevanja [127] (global warming potential - GWP) upotrebom PC-ja, tankih klijenata i tablet uređaja. GWP je definisan kao deo Kyoto protokola, i koristi se kao merna jedinica za izražavanje količine emisije gasova staklene bašte i generalno se izražava u funtama (lbs) ili kilogramima po proizvedenom kWh energije. U suštini, što je manji GWP predloženi sistem će ostaviti manje traga kroz emisiju štetnih gasova u proizvodnji električne energije. Posredno, sistem će imati manju štetnost na životnu sredinu.



Slika 6.6 - Ispitivanje globalnog potencijala zagrevanja (global warming potential - GWP) upotrebom PC-ja, tankih klijenata i tablet uređaja za elektronske učionice

Uz implementaciju novih klijentskih uređaja namenjenih učenju, na slici 6.6 se može videti da se uticaj na okolinu može značajno smanjiti, i to smanjenjem potrošnje električne energije što direktno utiče na smanjenje emisije štetnih gasova.

6.3. Finansijska analiza opremanja elektronske učionice

Nakon analiziranja potrošnje električne energije, dobijene rezultate uporedićemo sa cenom koštanja nabavke neophodne opreme. Kombinovanjem cene električne energije i cene koštanja opreme u 2015. godini, dobija se procena ukupnih troškova za jednu elektronsku učionicu.

Za nabavku 100 PC računara, biće odabrani računari skromnijih performansi čija je cena oko 370$. Za svaki od PC računara neophodan je i monitor. Odabran je Samsung LCD monitor - LS22C150NS/ZA, zbog svoje relativno niske cene od 130$ i niske potrošnje od 18W po radnom satu. Iz navedenog dobijamo da je cena koštanja kompletnog PC računara 500$. Dakle, za 100 PC računara potrebno je utrošiti 50.000$.

Kod izbora tankih klijenata, jedno od odabranih rešenja može biti HP t510 tanki klijent. Njegova cena je oko 300$. Na ovu cenu je neophodno dodati i monitor, takođe Samsung LCD monitor - LS22C150NS/ZA, čime dobijamo konačnu cenu pojedinačne konfiguracije od 400$. Za 100 ovakvih uređaja ukupni troškovi su 40.000$.

Konačno, kod primene sistema mobilnih uređaja svakom studentu je neophodan samo uređaj koji ima odgovarajuće performanse radi održavanja stabilne i konstantne veze sa serverom ili *cloudom*, kao i odgovarajuća softverska podrška. Ovakav pristup gde studenti koriste sopstvene uređaje u svrhu obrazovanja je poznat kao BYOD [128] (Bring-Your-Own-Device) pristup. BYOD trend je sve više rasprostranjen u obrazovnim ustanovama, jer sve veći broj studenata donosi svoje mobilne uređaje na predavanja. To je prepoznato kao mogućnost upotrebe tih uređaja u predavanjima i učenju. Ipak, donošenje mobilnih uređaja u učionicu može predstavljati izazov, jer se mora prepoznati željeno tehnološko rešenje koje će uskladiti potrebe i mogućnosti obrazovne ustanove i studenata. Tu treba imati u vidu potrebnu mrežnu infrastrukturu, bezbednost, IT i softversku podršku i eventualne prekide prilikom rada u učionici. Jedan od najvažnijih uslova koje je neophodno ispuniti kako bi saradnja između ustanove i studenata bila moguća je softverska podrška. Ona se sastoji u izradi pozadinskog sistema i aplikacije koja bi se koristila u elektronskoj učionici i bila deo strategije elektronskog učenja. Pristup BYOD za instituciju ujedno donosi manje početnih troškova, jer obrazovna ustanova (uglavnom) nema obavezu obezbeđenja klijentskih uređaja.

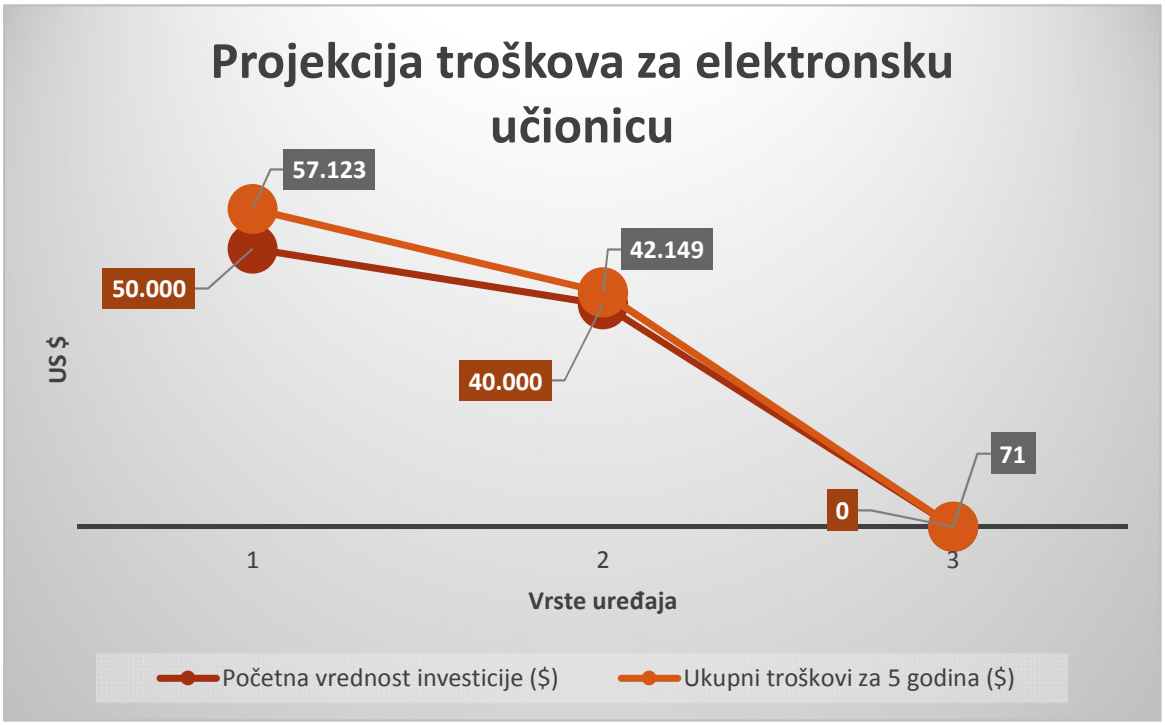
U tabeli 6.3 je prikazana zbirna kalkulacija troškova klijentskih uređaja i potrošnje električne energije. U navedenu kalkulaciju nije uključen trošak realizacije i potrošnje mrežne opreme

(ruteri, svičevi, kablovi) jer se konkretna realizacija može značajno razlikovati u zavisnosti od uslova i potreba svake obrazovne ustanove.

Tabela 6.3 - Kalkulacija troškova za elektronsku učionicu

Tip organizacije elektronske učionice	Početna vrednost investicije ($)	Ukupna potrošnja za godinu dana (kWh)*	Vrednost potrošnje za godinu dana ($)	Ukupni troškovi za 5 godina ($)
PC računari (1)	50.000	20352,50	1424,68	57.123,40
Tanki klijenti (2)	40.000	6142,50	429,98	42.149,90
Smartphone uređaji (3)	-	201,6	14,11	70,56

*Kalkulacija je urađena na osnovu prosečne cene kWh od $0.07.



Slika 6.7 - Uporedni pregled troškova za elektronsku učionicu u zavisnosti od odabranih platformi

Iz prethodnih analiza primećuje se da su najneisplativija rešenja upravo ona koja su do sada bila podrazumevana, a to je upotreba PC računara. Uz srednjeročnu analizu troškova izvodi se zaključak da je poželjno i najefikasnije koristiti smartphone/tablet klijentske uređaje, jer su troškovi minimalni (slika 6.7).

7. *CLOUD* I SISTEMI ZA ELEKTRONSKO UČENJE

Veb komponenta učenja predstavlja osnovni deo koncepta učenja na daljinu. Učenje zasnovano na vebu često se naziva i *online* učenje. Termin elektronsko učenje (E-Learning) se često odnosi na učenje čiji se sadržaj isporučuje putem Interneta ili preko drugih digitalnih medija ili elektronskih uređaja. Aplikacije i procesi učenja obuhvataju upotrebu računara, virtuelnih učionica i digitalne saradnje. Učenje na daljinu predstavlja situaciju u kojoj je nastavnik odvojen od studenata. Ta razdvojenost može biti lokacijska, vremenska, ili kombinovana. Kursevi za učenje se isporučuju u formi teksta i slika u elektronskom obliku, animacija, slajdova, video materijala, video konferencija, itd. Dakle, sam proces učenja na daljinu je prilagođen individualnim potrebama polaznika, pa se proces razvoja učenja, koji se kreće od elektronskog ka mobilnom učenju (*mobile learning* ili *m-learning*), upravo dodatno prilagođava individualnim potrebama polaznika. Još jedan od nedostataka klasičnog sistema jeste taj što samo postavljanje lekcija na Internet ne uči polaznike. Jedno od rešenja je upotreba softvera za učenje, kao što su ITS sistemi (*Intelligent Tutoring Systems*) koji se sastoje od softverskih modula sa implementiranim principima učenja, kognitivnih nauka, računarske lingvistike, matematike i veštačke inteligencije [129]. Kako bi se stepen uspeha u učenju povećao, osim individualnog prilagođavanja neophodno je primeniti simulacije koje odgovaraju dešavanjima u realnom okruženju. Time se doprinosi povećanju stepena obučenosti polaznika i primena stečenog znanja u realnom okruženju.

Dakle, proces učenja se može sagledati u nekoliko koraka [130]:

1. Pribavljanje informacija,
2. Učenje,
3. Obrada naučenog i primena,
4. Nastavak procesa učenja i unapređenja,
5. Učenje koje zatim postaje individualni proces interakcije između studenta i njegovog okruženja, čime se konstruiše subjektivna realnost studenta.

Elektronsko učenje je proces učenja zasnovan na internetu, koristeći internet tehnologije za projektovanje, implementaciju, upravljanje, podršku i proširenje učenja, koje neće zameniti tradicionalne metode obrazovanja, ali će u velikoj meri poboljšati efikasnost obrazovanja. Kako elektronsko učenje ima dosta prednosti kao što su fleksibilnost, diverzitet, merenje, otvaranje i tako dalje, ono postaje primarni način za učenje [131].

U sada već tradicionalnom elektronskom učenju zasnovanog na vebu, izgradnja i održavanje sistema se nalaze u unutrašnjosti obrazovne institucije ili kompanije, što ostavlja dosta prostora za probleme kao što su potrebe za značajnim investicijama, ali bez kapitalnih dobiti za njih, što dalje dovodi do nedostatka razvojnog potencijala. Nasuprot tome, modeli elektronskog učenja zasnovanog na *cloudu* uvode mehanizam razmere efikasnosti, odnosno izgradnja sistema je poverena *cloud computing* dobavljačima, koji mogu omogućiti da provajderi i korisnici ostvare takozvanu *win-win* situaciju: s jedne strane, kompanije dobavljači mogu da koriste svoje tehnološke prednosti za izgradnju sistema učenja sa stabilnijim performansama, više sveobuhvatnih funkcija i sigurnijim karakteristikama. U međuvremenu, dobavljači mogu da preuzmu odgovornost tako da postoji povraćaj uloženih sredstava. S druge strane, korisnici mogu biti oslobođeni izgradnje i održavanja sistema i mogu se posebno fokusirati na primenu sistema za elektronsko učenje u cilju poboljšanja kvaliteta nastave i upravljanja. Dakle, pojava *cloud computinga* stvara nove ideje za dalji razvoj elektronskog učenja.

Zbog mogućnosti primene *cloud computinga* u oblasti obrazovanja, ispitano je dosta problema, kao što su tehnologija i morfologija za buduće obrazovanje na daljinu uz pomoć *clouda* [132], nastavni informacioni sistemi, integracija resursa učenja, razvoj nastavnih sistema.

U integraciji elektronskog učenja i računarskih mreža, naglasak je stavljen na izgradnju hardverskih i softverskih platformi za elektronski sistem učenja, funkcionalnu strukturu, upravljanju bezbednošću računarskih mreža i obuka, integraciji informacionih tehnologija u nastavi, itd.

Iz prethodno navedenog možemo videti da su do sada istraživači pretežno usmerili istraživanja na sledeća dva aspekta:

- Integracija interneta i elektronskog učenja i
- *Cloud computing* u oblasti obrazovanja.

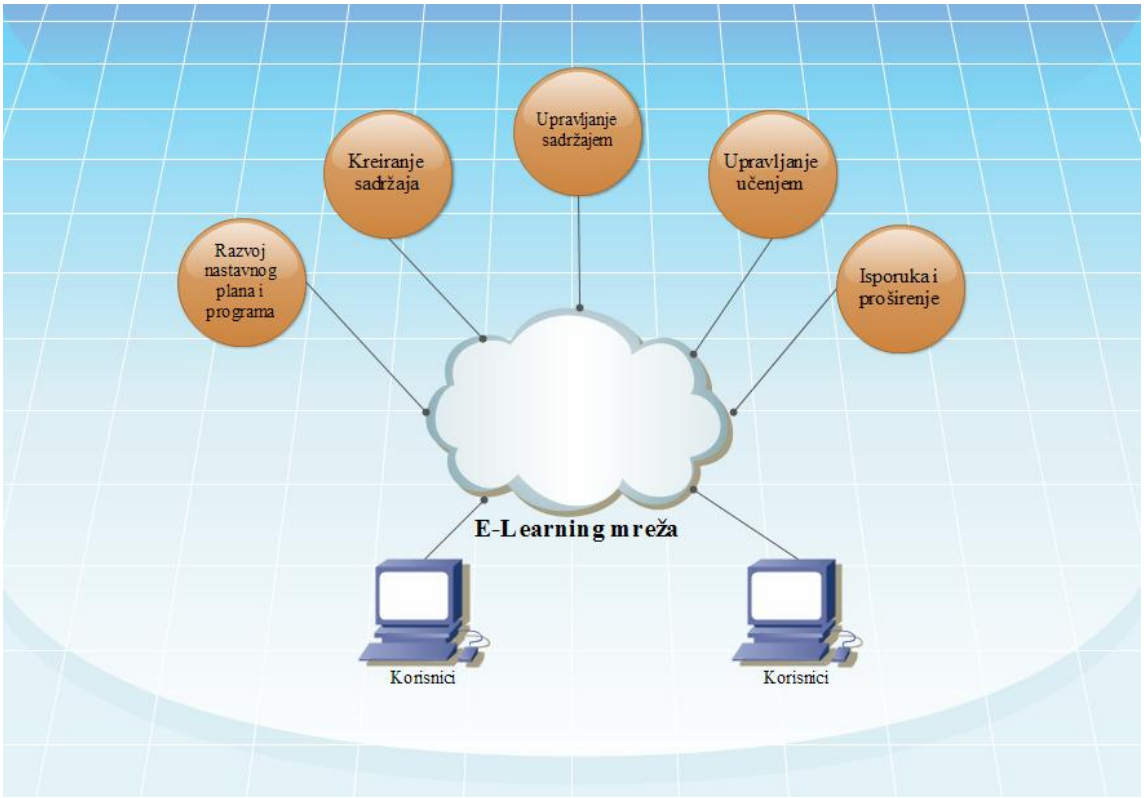
7.1. Tradicionalni model elektronskog učenja

Tradicionalna mreža za elektronsko učenje nalazi se u kampus mreži ili intranetu i održava je visokoškolska ustanova ili preduzeće. Standardna platforma koja se koristi u edukaciji jeste Moodle [133] platforma za učenje. Ona omogućuje kreiranje kurseva, kreiranje grupa, postavljanje sadržaja koje studenti mogu preuzimati i pregledati, zatim rešavanje kvizova i sl.

Za razvoj i implementaciju pomenute platforme, neophodno je obuhvatiti sedam osnovnih celima:

1. Inteligentna IP mrežna infrastruktura,
2. Razvoj nastavnog plana i programa,
3. Kreiranje sadržaja,
4. Upravljanje sadržajem,
5. Upravljanje učenjem,
6. Isporuka i
7. Proširenje.

Inteligentna IP mrežna infrastruktura predstavlja osnovu digitalne platforme kampus mreže, uglavnom kako bi se ispunili uslovi u pogledu kvaliteta usluge. Podaci, multikast tehnologija, bezbednost, upravljivost, keširanje, tehnologija distribucije sadržaja, obezbeđuje visoko dostupnu i kontrolisanu infrastrukturu. Tradicionalna mreža za elektronsko učenje (E-Learning) je prikazana na slici 7.1 [134].



Slika 7.1 - Tradicionalna E-Learning mreža

Za potrebe elektronskog učenja kreiraju se programi sa multimedijalnim edukativnim sadržajem.

Centar za upravljanje sadržajima se sastoji iz dva dela:

- Server za upravljanje sadržajem i
- Distribucija sadržaja.

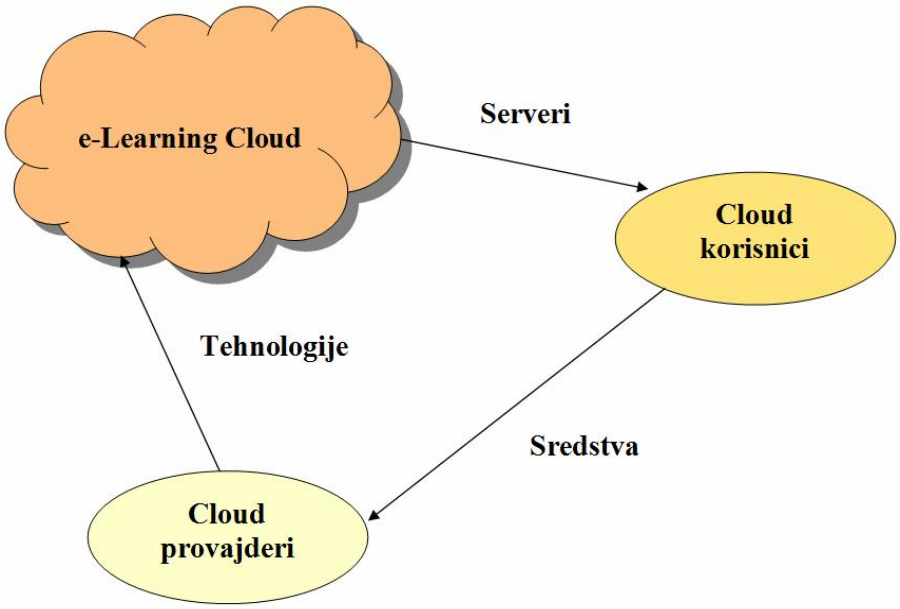
Server za upravljanje sadržajem koristi napredne koncepte za internet učenje, fokusirajući se na prenos multimedijalnog sadržaja, učenje hipertekst sadržaja, komunikaciju među učenicima i učenje.

Sistem za distribuciju sadržaja je dostupan zbog upravljanja nad velikim brojem fajlova i multimedijalnih kanala, zajedno sa sadržajima u pristupnim tačkama, tako da korisnici u pristupnim tačkama mogu smanjiti „usko grlo“ protoka u WAN mreži, čime se poboljšava sistem.

7.2. *Cloud computing* model za elektronsko učenje

Tradicionalne mreže za elektronsko učenje, kao i sve aplikacije unutar te mreže, su napravljene i održavane od strane škola ili kompanija. Dakle, troškovi investiranja u opremu, razvoja i održavanja snose škole ili kompanije. Ali, ako se konstruisanje, razvoj, upravljanje i održavanje ovakvih sistema umesto školi ili kompaniji poveri *cloud* provajderima, i ako se sistem otvori većem krugu korisnika putem interneta, onda se troškovi znatno smanjuju, kako za obrazovne institucije, tako i za provajdere. Ovakav poslovni model se naziva *cloud computing* model za elektronsko učenje.

Cloud computing model funkcionisanja je prikazan na sledećoj slici.



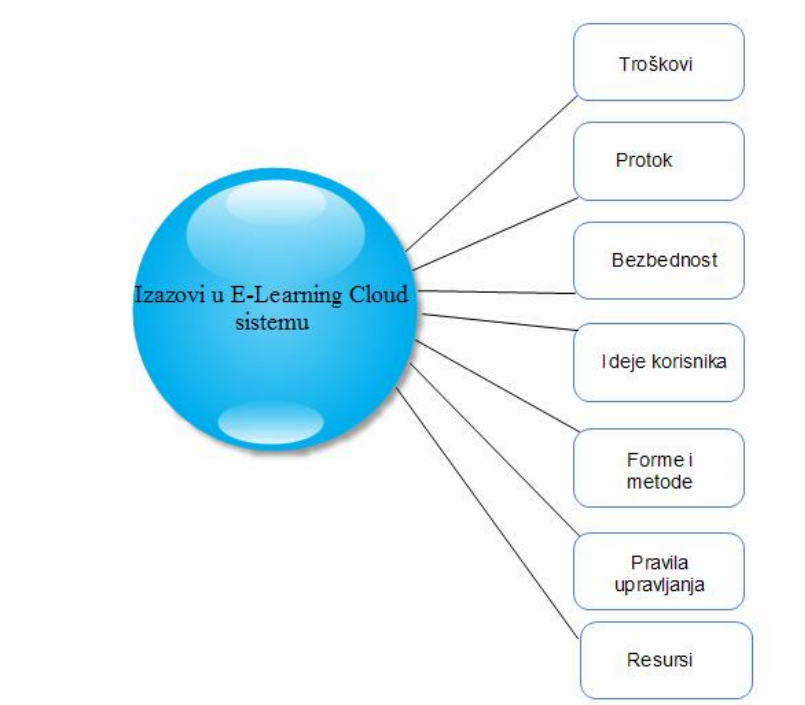
Slika 7.2 - Model elektronskog učenja zasnovan na *cloud computing* platformi

Kod *cloud computing* modela elektronskog učenja, *cloud* provajder je odgovoran za izgradnju i održavanje *cloud* sistema i pružanje tehničke podrške. *Cloud* korisnici plaćaju provajderu za usluge u *cloud* sistemu, usluge kojima se pristupa na zahtev. Na slici 7.2, tokom ciklusa,

serveri daju podršku korisnicima, korisnici svojim finansijskim sredstvima daju podršku provajderu, a tehnologije obezbeđuju podršku za *cloud*, što predstavlja poslovni ciklus.
Cloud namenjen elektronskom učenju predstavlja migraciju *cloud computing* tehnologije u oblast elektronskog učenja, što je budućnost infrastrukture za učenje.

7.3. Izazovi u cloud tehnologiji namenjenoj elektronskom učenju

Cilj razvijanja elektronskog sistema učenja je zadovoljenje zahteva korisnika. Provajderi su odgovorni za razvoj sistema, gde leži osnova komercijalne vrednosti ove tehnologije. C*loud* tehnologija elektronskog učenja predstavlja izazov novijeg datuma i zato predstavlja predmet pažnje i provajdera i korisnika. Slika 7.3 prikazuje neke od očiglednih problema kojima je potrebno posvetiti posebnu pažnju.



Slika 7.3 - Izazovi u *E-Learning cloud* sistemu

Cloud computing okruženje predstavlja efikasnu paradigmu jer obezbeđuje gotovo trenutnu isporuku servisa na zahtev bez potrebe za ljudskom interakcijom sa obe strane, heterogenost platformi za pristup servisima, balansiranje resursa za usluživanje velikog broja korisnika, elastičnost i merljivost servisa, čime se omogućuje plaćanje samo onoga što je i korišćeno. Ipak, postoje i određene stvari na koje treba obratiti posebnu pažnju kada je učenje na daljinu u pitanju:

Bezbednost i privatnost - Kako podaci mogu biti distribuirani na mnogim serverima, van domašaja korisnika, postoji potreba za kontrolom resursa uz pomoć različitih metoda. Kako bi se očuvala poverljivost korisnika, potrebno je koristiti različite vrste sertifikata i protokola.

Dostupnost i očuvanje podataka – Neophodna je mogućnost konstantnog pristupa sistemu uz upotrebu redundantnih sistema kako bi se izbeglo preopterećenje mreže, gubitak podataka i sl.

Skalabilnost – Obavezna je upotreba sistema za konstantno praćenje korišćenja resursa, radi praćenja trenutnih zahteva korisnika i eventualnih pokušaja zloupotrebe korišćenja sistema.

Energetska efikasnost – Neophodno je vršiti što je veću moguću uštedu električne energije upotrebom procesora sa malom potrošnjom, adaptivnih sistema i optimizovanih algoritama.

Upoređujući tehnologije učenja i prihvatanja znanja, neosporno je da učenje zasnovano na vebu ima više prednosti nego konvencionalno učenje u učionicama. Prednosti ovakvog načina učenja jeste smanjenje troškova, ne postoji potreba za fizičkim prostorijama, a student može pratiti predavanja kada god to njemu odgovara. Materijal za učenje je lako ažurirati, a predavač može lako ubaciti i multimedijalne sadržaje kako bi olakšao razumevanje svojih materijala za određeni kurs.

Pored navedenih, postoje i određeni nedostaci koji moraju biti prepoznati i rešeni pre potpune integracije elektronskih i mobilnih platformi u akademski program. Ove platforme je naravno moguće primeniti i primenjuju se, ali treba obratiti pažnju na stvari kao što je skalabilnost na infrastrukturnom nivou. U slučaju povećanog pristupa studenata, odjednom se angažuje velika količina resursa i na taj način značajno povećavaju troškovi. Ključni problem jeste efikasno iskorišćenje raspoloživih resursa. Na primer, serveri i kompjuterske laboratorije su gotovo neiskorišćene u toku noći i prilikom raspusta. Sa druge strane, dati resursi su potpuno zauzeti pri kraju semestra. Konačno, obrazovna ustanova mora plaćati i dodatne troškove za održavanje mreže, računarskih sistema, softver i sl.

Iz navedenih razloga, nameće se potreba za promenom starog pristupa upotrebe kompjuterskih resursa, gde edukaciona ustanova kupuje svu neophodnu opremu, infrastrukturu i softver. Ta promena dovodi do rešenja upotrebom *cloud computing* tehnologije, čime bi se smanjili troškovi upotrebe kompjuterskih resursa.

7.4. Aplikacije namenjene elektronskom učenju u cloud computing okruženju

Jedan od načina povećanja kvaliteta jeste razvoj i povezivanje aplikacija namenjenih učenju na daljinu sa *cloud* sistemom, kao i korišćenjem mogućnosti virtuelizacije *cloud* sistema, čime se smanjuju troškovi i vreme potrebno za instalaciju i održavanje sistema.

Jedna od metoda koje se koristi u praksi jeste upotreba IaaS *cloud* tehnologije radi upotrebe virtuelnih mašina od strane studenata [135]. Sledeći primer je BlueSky [136], čija arhitektura poseduje više komponenti namenjenih efikasnom upravljanju E-Learning servisima, pravovremenoj alokaciji resursa, i sl. Još jedno od predloženih rešenja jeste CloudIA [137]. To je rešenje koje obezbeđuje kreiranje i konfiguraciju virtuelnih mašina na zahtev, na način da studenti imaju sopstveno Java servlet okruženje, koje se sastoji od MySQL, Tomcat, PHP i Apache web servera. Uz ovakav pristup, studentima ostaje više vremena za razvoj i testiranje svojih aplikacija u servlet kontejneru. U [138], autori su predstavili novi servisni model koji povećava efikasnost unutar virtuelnog personalizvanog okruženja za učenje. Ovaj sistem je namenjen za prijavu odabranih sadržaja za učenje, kao i kreiranje personalizovane virtuelne učionice. Provajderima sadržaja se omogućuje da registruju svoje aplikacije na serveru, čime se studentima daju dodatni izvori za učenje i savladavanje odabranog kursa. Sledeće rešenje za lično i virtuelno učenje koji se oslanjaju na *cloud* predstavljaju poznati Google servisi YouTube i GoogleDocs.

Primeri savremenih komercijalnih i nekomercijalnih LMS sistema su Blackboard Learn [139], Moodle [140], Claroline [141] i Canvas [142]. Blackboard Learn predstavlja virtuelno okruženje namenjeno učenju kroz kurseve. Ovaj komercijalni softver je zasnovan na vebu i omogućuje upravljanje kursevima. Može poslužiti kao dodatak klasičnom načinu držanja kurseva, ali ima sve elemente neophodne za potpuno održavanje online kurseva. Moodle je softver za učenje na daljinu uz pomoć veba, i ujedno je sistem za upravljanje kursevima. Moodle je nekomercijalna platforma i predstavlja jedno od najpopularnijih rešenja namenjenih elektronskom učenju. Claroline je *open source* platforma namenjena edukaciji. Claroline omogućuje postavljanje dokumenata u bilo kom formatu, uređivanje foruma, kreiranje testova, slanje obaveštenja putem email poruka, postavljanje domaćih zadataka, ocenjivanje, itd. Canvas je komercijalni LMS sistem koji je zasnovan na sopstvenoj *cloud* platformi. Platforma podržava masovne otvorene *online* kurseve (Massive Open Online Course - MOOC), čime je kompanija nastojala da olakša eksperimentisanje sa novim

pedagoškim metodama i novim načinima upotrebe multimedijalnih okruženja radi poboljšanja procesa učenja.

Navedeni sistemi imaju prednosti koje se ogledaju u mogućnosti lake dostupnosti materijala za učenje, postavljanje softverskih alata koji olakšavaju učenje, kreiranje diskusionih grupa, itd.

Postoje i određeni radovi koji upoređuju efikasnost *online* i tradicionalnih metoda učenja [143]. Jedan od najreprezentativnijih istraživanja jeste [144], gde su se autori fokusirali na uticaj tehnologije i olakšanu upotrebu i ubrzavanje procesa učenja. Dalje je analiziran određeni nivo apstrakcije, IaaS ili PaaS model, koji treba dostaviti studentima kako bi im se olakšalo savladavanje odabranog kursa.

Pored navedenih prednosti, postoje i ograničenja. Ona se ogledaju u visokoj ceni desktop ili laptop računara, velikom prostoru koji desktop računar zauzima, kao i pristup materijalima za učenje sa fiksne lokacije [145].

8. MOBILNA REŠENJA NAMENJENA UČENJU

Elektronsko učenje obuhvata učenje zasnovano na vebu, učenje uz pomoć računara, u virtuelnim učionicama. Kao prirodni nastavak unapređenja učenja pojavila se potreba usklađivanja sa *cloud computing* sistemima. Ovo poglavlje je posvećeno sistemima za učenje uz pomoć mobilnih aplikacija, uz podršku *cloud* sistema.

Napredak u bežičnim komunikacijama doneo je nove mogućnosti implementacije novih strategija obrazovanja i to kombinacijom fizičkih okruženja za učenje sa resursima digitalnog sveta. Uz pomoć ovih novih tehnologija, svakom studentu se pruža mogućnost da uči uz pomoć digitalnog sadržaja kome može pristupiti bežičnim putem. Pojava prenosnih uređaja i bežičnog umrežavanja dovodi do promene i unapređenja koncepta elektronskog učenja (e-learning) na koncept mobilnog učenja (m-learning). Ipak, trenutna ograničenja mobilnih uređaja predstavljaju mane novog koncepta učenja. Ti nedostaci se najviše ogledaju u količini raspoložive memorije, veličini ekrana, i sl. Postojeći koncepti elektronskog učenja se ne uklapaju u potpunosti u novi model učenja. Iz tog razloga, neophodne su određene modifikacije kako bi se veća količina literature za učenje nalazila na samom korisničkom uređaju. Razlog ovakvom pristupu informacijama leži u mogućim problemima sa mrežom kojoj uređaj pristupa. Iako mobilni uređaji mogu imati pristup mobilnim mrežama gotovo svuda, problem može nastati zbog nestabilnosti mreže, slabljenja signala i sl. Sve to prouzrokuje smanjenje protoka podataka ili potpuni prekid komunikacije, pa je zato neophodno imati unapred dostupnu literaturu na samom uređaju. Tu literaturu koju student tek treba pročitati, preslušati ili odgledati aplikacija namenjena mobilnom učenju može automatski preuzimati i skladištiti bez odobrenja korisnika u uslovima kada je komunikacija između sistema i uređaja dobra, kada je signal kvalitetan i protok stabilan. Iako navedeni nedostaci postoje, oni ne mogu značajno ugroziti proces učenja i prilagođavanje učenja potrebama studenata koji prate kurseve [146].

Prilikom dizajniranja mobilne aplikacije namenjene učenju, treba voditi računa o načinu na koji će se aplikacija praviti. Naime, metode kojim se programeri vode za pravljenje platforme za elektronsko učenje ne mogu biti iste kao metode za pravljenje platforme za mobilno učenje. Osnovne razlike između ova dva načina učenja ogledaju se u kontekstu upotrebe uređaja preko kojih se pristupa aplikacijama za elektronsko i mobilno učenje. Dakle, ključ pravljenja uspešne platforme za mobilno učenje jeste razumevanje konteksta (tabela 8.1).

Tabela 8.1 - Poređenje konteksta upotrebe

Kontekst	Elektronsko učenje	Mobilno učenje
Platforma za pristup	Desktop ili laptop (veliki ekran)	mobilni uređaj sa malim ekranom (smartphone ili tablet)
Položaj studenta	Sedeći	Sedeći, stojeći, u šetnji, ležeći
Ometanja	Malo	Mnogo
Vreme učenja	Manji broj dužih perioda	Veći broj kraćih perioda
Ulazni uređaj	Tastatura i miš	Ekran osetljiv na dodir
Mesto učenja	U kancelariji ili kući	Bilo gde, bilo kada (u autobusu, parku, krevetu)

Dakle, iz tabele možemo zaključiti da se načini učenja mogu značajno razlikovati, pa je s toga neophodno pristupati dizajniranju ovih platformi na drugačiji način. Dakle, kod mobilnog učenja je neophodno prvo dizajnirati mobilnu platformu za učenje, pa tek sadržaj, dok kod elektronskog učenja taj proces može biti obrnut.

8.1. Primena pedagoških metoda u obrazovanju upotrebom mobilnog sistema

Napredak u bežičnim komunikacijama doveo je do novih mogućnosti u implementiranju novih strategija obrazovanja. Nove strategije obrazovanja treba zasnivati na kombinaciji digitalnog sveta i fizičke lokacije učenja.

Zahvaljujući novim tehnologijama, studenti su u mogućnosti da uče iz digitalnog sadržaja sa bilo kog mesta gde postoji mogućnost mrežnog povezivanja. Usavršavanje mobilnih uređaja i bežičnog umrežavanja je dovelo do transformacije koncepta elektronskog učenja u koncept mobilnog učenja.

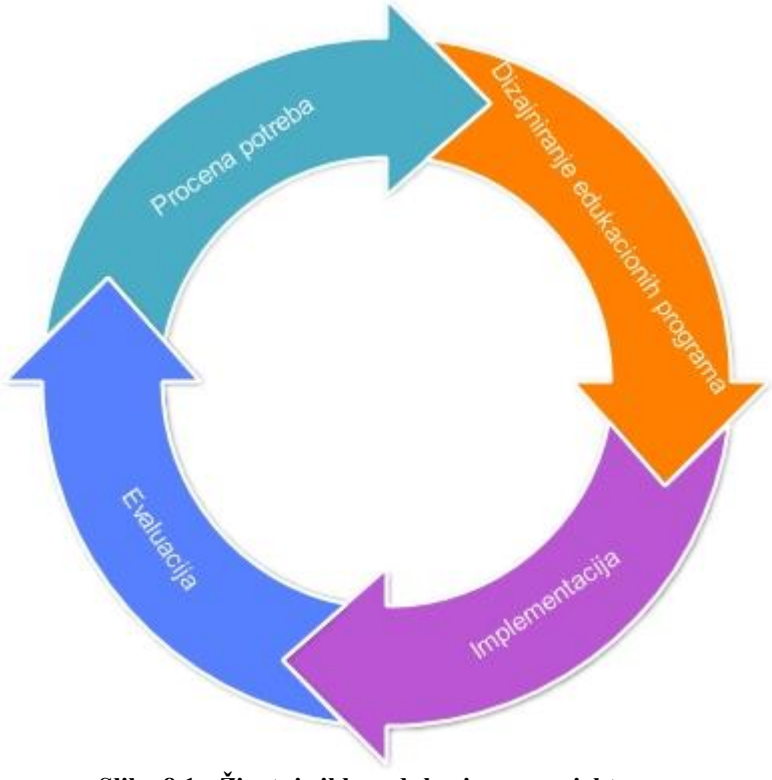
Mobilno učenje predstavlja pogodan način učenja, jer je takav servis dostupan na bilo kojoj lokaciji. Saradnja je obično jedna od prednosti ovakvog načina učenja, kao i trenutno deljenje sadržaja svim učesnicima, što dalje može dovesti do brzih povratnih informacija. Još jedna od pogodnosti jeste i laka prenosivost sadržaja, čime se zamenjuju tradicionalne knjige [147].

Samim definisanjem mobilnog učenja menja se fokus sa mobilnih uređaja na mobilne učenike, i od dizajnera se zahteva da ne dizajniraju instrukcije za novu klasu mobilnih tehnologija, već da prošire svoje poglede na način na koji mobilnost utiče na učenje. U literaturi postoje određena rešenja za učenje u mobilnom okruženju [148]. Ono što može biti ograničavajući faktor u potpunom razvoju ovog koncepta jesu performanse i veličina mobilnih uređaja.

Pored isključivo tehničke strane razvoja platforme, neophodno je konstantno voditi računa i o uravnoteženoj implementaciji pedagoških metoda [149]. Primenom uravnoteženog rešenja dobija se maksimalno produktivno i efikasno rešenje. Dakle, moderna primena informaciono komunikacionih tehnologija u obrazovanju zahteva pažljivu evaluaciju i adekvatnu primenu u skladu sa pedagoškim principima [150]. Dosadašnja istraživanja ukazuju na neophodnost poboljšanja principa učenja kod primene mobilnog učenja. Razvoj mobilnog softvera mora biti pažljivo planiran. Butoi, A., Tomai, N., & Mocean, L. (2013) su predstavili iterativni proces razvoja programa za edukaciju [151], koji je prihvaćen kao osnovni pedagoški princip za razvoj mobilne aplikacije. Iterativni proces razvoja programa za edukaciju se sastoji iz četiri koraka:

1. Procena potreba,
2. Dizajniranje edukacionih programa,
3. Implementacija i
4. Evaluacija.

Ovaj proces je grafički prikazan na slici 8.1.



Slika 8.1 - Životni ciklus edukacionog projekta

U prvom koraku definišu se pedagoške potrebe koje moraju biti definisane i implementirane u mobilnom okruženju. Na strani instruktora, to može predstavljati količinu informacija koje će biti prikazane svakom od studenata. Sadržaj mora biti lako razumljiv i prihvatljivog obima. Drugi korak definiše dizajn programa za edukaciju. Ovaj korak određuje način na koji se

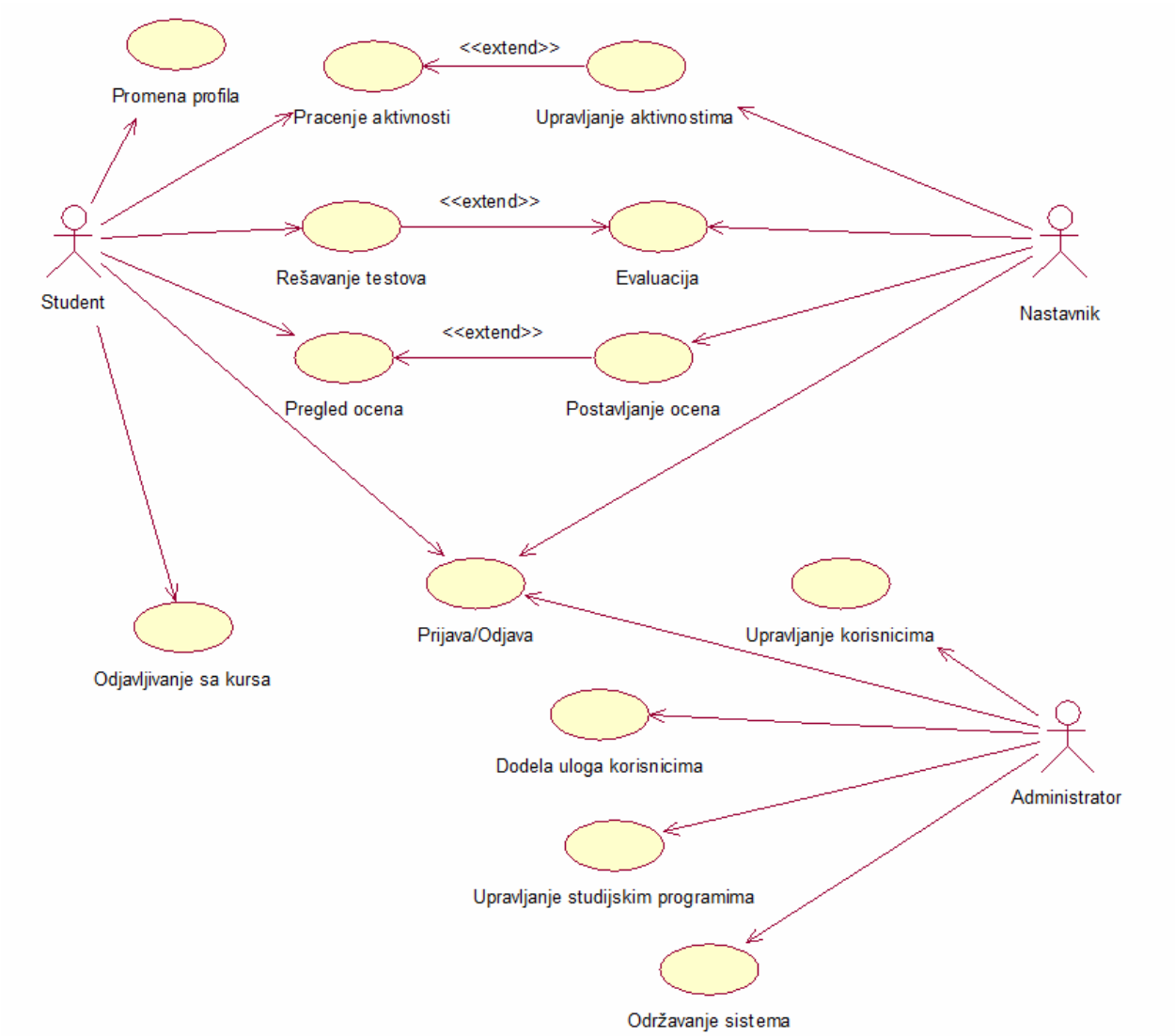
prikazuje sadržaj. Kada se koristi mobilno okruženje, poželjno je primeniti interaktivnost, sa savremenim dizajnom koji će se svideti studentima. U trećem koraku razvoja aplikacije, neophodno je iskoristiti prednosti pedagoških metoda. Nakon što student pregleda sadržaj, aplikacija treba da mu predstavi praktičan problem koji će moći da reši. U mobilnom okruženju, ovo se može postići različitim testovima, koji mogu biti povezani čak i sa okruženjem studenta. U četvrtom koraku se definiše evaluacija. Evaluacija se ispunjava ocenjivanjem aplikacije i sadržaja. Zahvaljujući povratnim informacijama vrši se dalje usavršavanje prototipa, sadržaja i testova, uz bolju upotrebu pedagoških metoda.

Sa daljim razvojem mobilnog učenja, neophodno je osloniti se na koncept mobilnog učenja i mobilne aplikacije koja integriše *cloud computing* i mobilno učenje.

8.2. Dizajniranje sistema

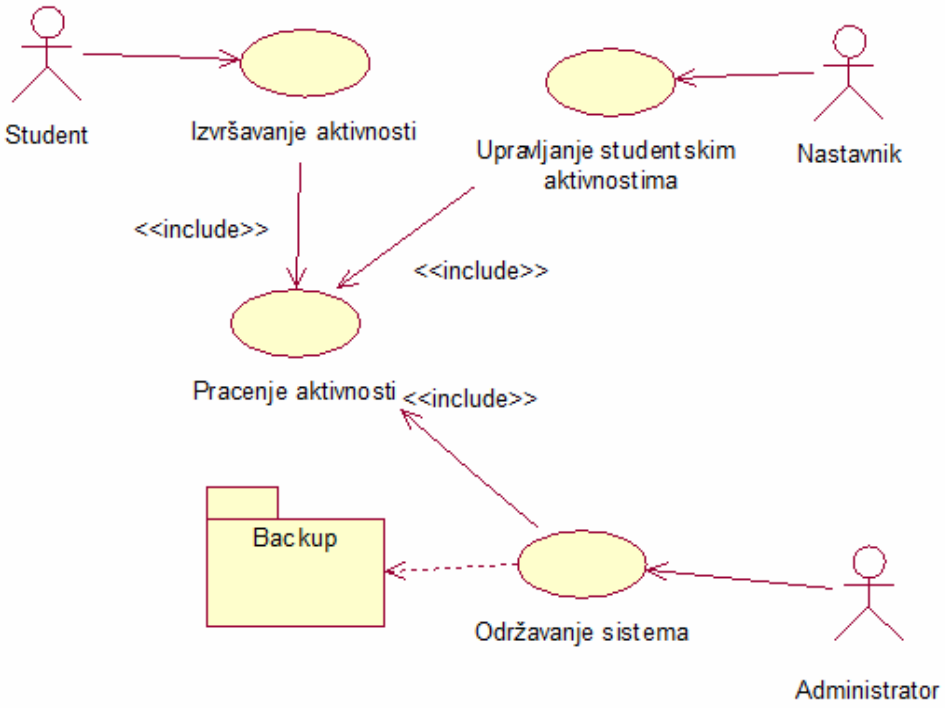
Kod dizajniranja sistema i softvera, poželjno je koristiti dijagrame kako bismo definisali sve delove sistema i njihove uloge. U tu svrhu se koriste *Use Case* dijagrami. Use case dijagrami se koriste kako bi se prikazala funkcionalnost sistema. On prikazuje relacije različitih entiteta između njih i koristi koristi se za prikaz koraka koji definišu interakciju između učesnika – aktora (eng. *actors*) i sistema radi postizanja određenog cilja. Aktor može biti osoba, organizacija ili spoljni sistem koji ima ulogu (rolu) i jednoj ili više interakcija sa sistemom [152].

U projektu izrade mobilnog sistema namenjenog učenju, aktori predstavljaju primarne korisnike sistema. Na sledećoj slici prikazan je osnovni use case dijagram mobilnog sistema.



Slika 8.2 - Glavni use case dijagram projekta

Kao što dijagram prikazuje, postoje tri korisnika: učesnik kursa (student), nastavnik kursa (nastavnik) i administrator sistema (administrator). Na sledećoj slici prikazan je use case dijagram uzajamne zavisnosti aktora.



Slika 8.3 - Prikaz uzajamne zavisnosti aktora

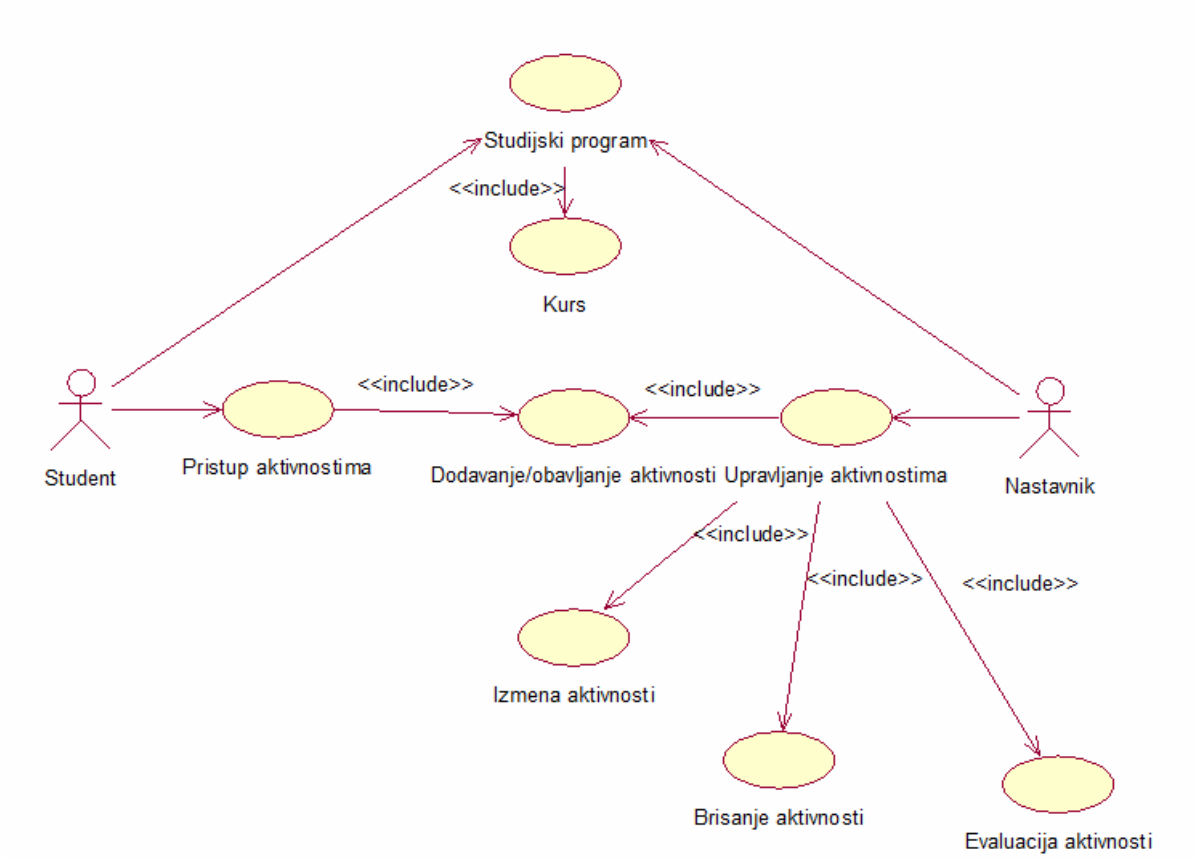
Aktor student je zadužen za praćenje aktivnosti u okviru kursa. To može biti pregled novih poglavlja, polaganje testova, itd. Nastavnik je odgovoran za upravljanje aktivnostima. Ovo obuhvata dodavanje, menjanje i brisanje aktivnosti namenjenih studentima. Sve aktivnosti studenata se prate, kako bi aktor nastavnik imao uvida u rad i napredak studenata. Sa druge strane, administrator je zadužen za upravljanje i održavanje sistema, praćenje aktivnosti ostalih korisnika, kao i pravljenje rezervnih kopija sistema i baza podataka. Svi aktori su zavisni od aktivnosti koje obavljaju ostali korisnici. Za svakog od aktora će biti prikazan use case dijagram sa detaljnim aktivnostima koje obavljaju.

Aktor nastavnik je odgovoran za upravljanje aktivnostima u okviru sopstvenog kursa. Sve njegove aktivnosti direktno utiču na aktivnosti aktora student. Iz tog razloga, njihove aktivnosti je neophodno prikazati zajedno na jednom use case dijagramu.

Kako bi pravilno definisali use case dijagram, potreban nam je određeni alat kojim definišemo korake obavljanja aktivnosti u dijagramu. Taj alat se naziva use case izveštaj (eng.report). On pokriva grafičku use case reprezentaciju u formi detaljnog izveštaja. Kroz predstavljeni izveštaj je jednostavnije razumeti sve aktore i aktivnosti koje aktori obavljaju. Ovako se jasno definišu najvažniji koraci koji se obavljaju u sistemu.

Tabela 8.2 - Use case izveštaj za aktore nastavnik i student

Naziv use case dijagrama	Upravljanje aktivnostima
Aktori	Nastavnik Student
Tok događaja	1. Izbor studijskog programa 2. Izbor kursa 3. Dodavanje aktivnosti (predavanje/test) uz predefinisane odgovore i postavljanje na sistem 4. Student obavlja aktivnosti 5. Pregled aktivnosti studenata 6. Evaluacija aktivnosti 7. Izmena/Brisanje aktivnosti
Uslov	Prijavljen na sistem sa određenim korisničkim imenom i lozinkom

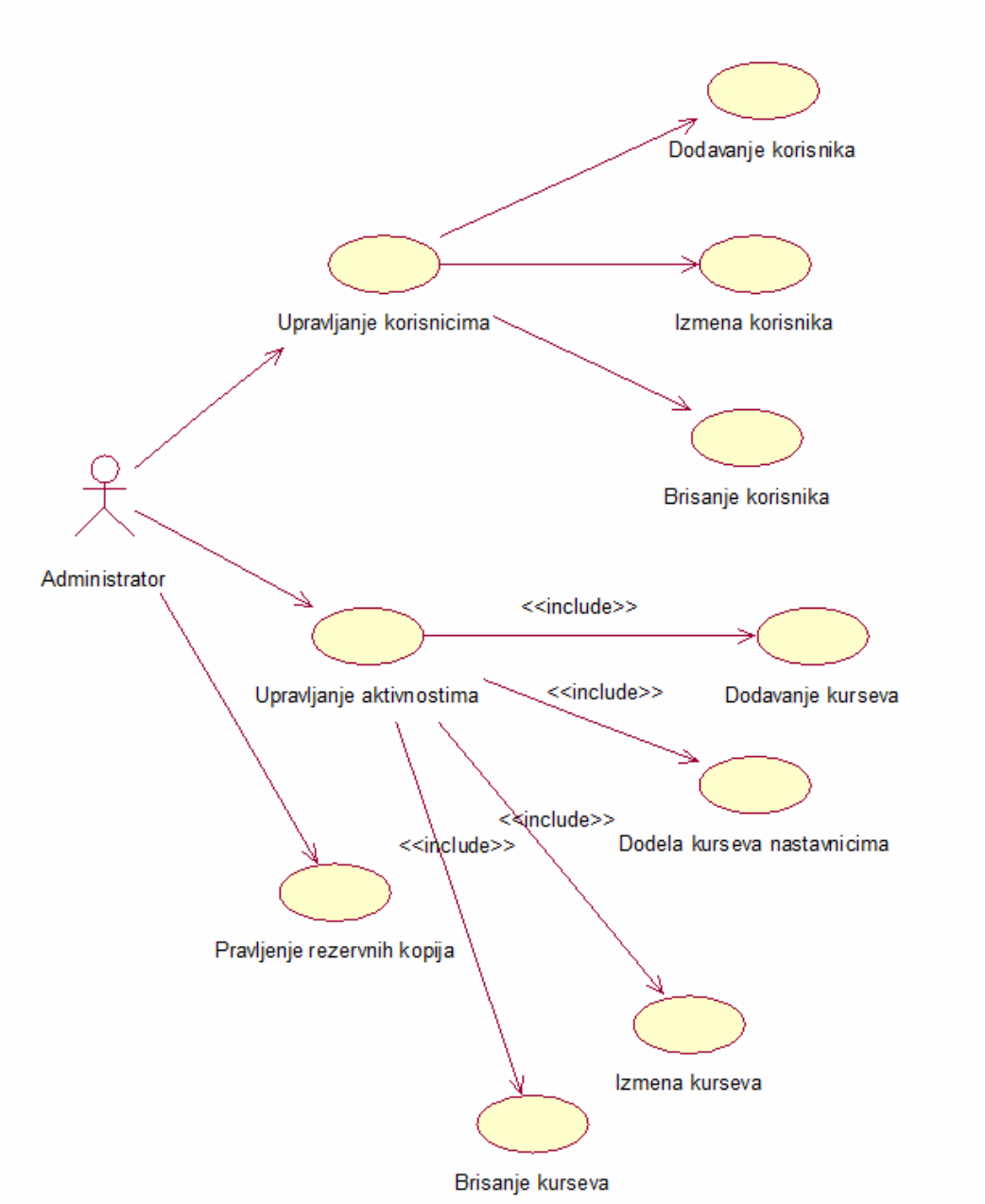


Slika 8.4 - Use case relacije aktora nastavnik i student

U tabeli 8.3 i na slici 8.5 prikazane su detaljne aktivnosti administratora. Administrator je odgovoran za definisannje i upravljanje studijskim programima i kursevima, korisnicima i njihovim rolama. On prati aktivnosti korisnika.

Tabela 8.3 - Use case izveštaj aktora administrator

Naziv use case dijagrama	Upravljanje aktivnostima
Aktor	Administrator
Tok događaja	1. Upravljanje korisničkim nalozima (Dodavanje, izmena, brisanje) 2. Pravljenje kurseva 3. Dodela kurseva nastavnicima 4. Izmena kurseva 5. Brisanje kurseva 6. Praćenje aktivnosti i pravljenje rezervnih kopija
Uslov	Prijavljen na sistem sa određenim korisničkim imenom i lozinkom



Slika 8.5 - Use case aktora administrator

Administrator je takođe odgovoran za dodavanje, izmenu i brisanje kurseva u sistemu. On može dodeliti nastavnike svakom definisanom kursu. Nakon završetka školske godine, administrator može napraviti rezervnu kopiju kurseva i obrisati sve aktivnosti u kursevima.

8.3. Arhitektura mobilnog sistema

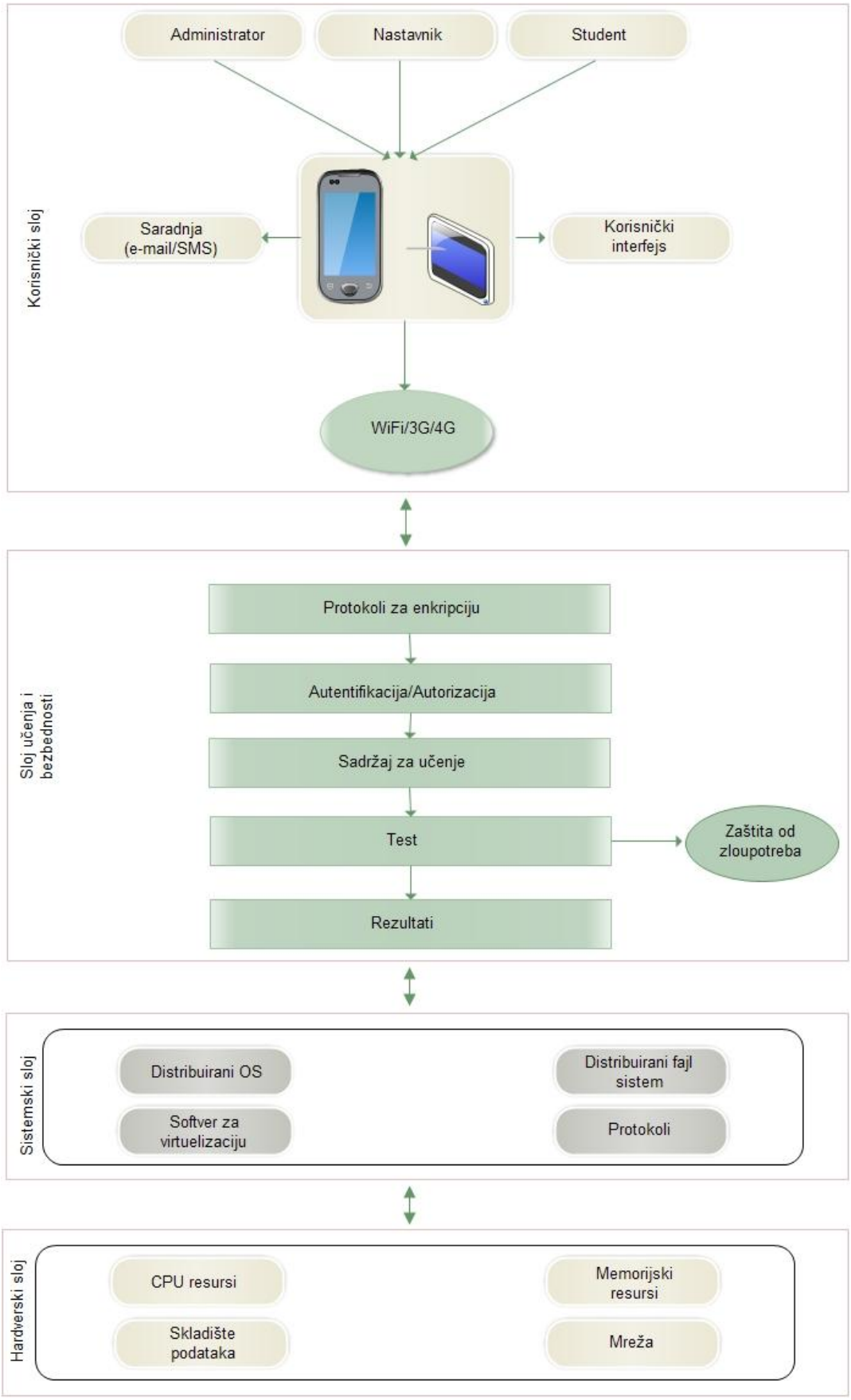
Integrisano *cloud* rešenje predstavlja idealnu podršku za mobilne sisteme za edukaciju. Kako bi razvili potpuno funkcionalni sistem, neophodno je izvršiti konceptualno raslojavanje, na način prikazan na slici 8.5. Sistem je podeljen na četiri celine.

Prvi sloj je hardverski sloj, koji se sastoji od fizičkog sistema i predstavlja osnovu *cloud* sistema. Sistem sadrži procesorske i memorijske resurse, skladište podataka i uređaje za transfer podataka između *cloud* sistema i korisnika.

Drugi sloj je sistemski sloj koji se, sa jedne strane sastoji od osnovnog softvera koji je zadužen za virtuelizaciju resursa, a sa druge predstavlja operativni sistem i protokole koji omogućuju rad sistema i komunikaciju između korisnika i *cloud* aplikacije za mobilno učenje.

Treći sloj obuhvata sloj učenja i bezbednosti, koji obuhvata protokole za zaštitu komunikacije, integrisani proces autentifikacije i autorizacije korisnika, sadržaje za učenje, sadržaje za proveru znanja studenata, deo za čuvanje informacija o napredovanju studenata i komponentu sistema koja sprečava zloupotrebe prilikom rešavanja testova.

Četvrti sloj je korisnički sloj. On definiše mobilne uređaje, saradnju između učesnika i aplikaciju za mobilnu platformu.



Slika 8.5 - Arhitektura *m-Learning* platforme zasnovane na *cloudu*

8.4. Aplikacije namenjene mobilnom učenju

U literaturi se može naći više praktičnih rešenja namenjenih učenju u mobilnom okruženju.

Quizzer [153] je mobilna aplikacija prvenstveno namenjena rešavanju kviz pitanja. Postoje dve vrste kvizova: kviz koji zahteva tačnu lokaciju studenta i kviz koji ne zahteva lokaciju. Iskorišćavanjem lokacijskih senzora pametnog telefona moguća je interakcija sa okolinom studenta. Ova mobilna aplikacija podržava multimedijalna pitanja u vidu slika i audio pitanja, kao i repozitorijum pitanja od kojih student može birati na koja će davati odgovore.

KnowleMobiLe [154] aplikacija je zasnovana na servisno orijentisanoj arhitekturi. Sastoji se iz više delova, a za učenike su najvažnije mogućnost navigacionog interfejsa, sa mogućnošću pretrage po ključnoj reči, na osnovu lokacije, kombinovano i interaktivnom pretragom. Ova mobilna aplikacija se oslanja na sopstvenu bazu podataka pojmova, zatim *Google maps* platformu, sadrži adrese do *Wiki* stranica i stranica sa slikama prethodno definisanih pojmova u okviru aplikacije.

MEEFLS [155] mobilna aplikacija je prvenstveno namenjena za podučavanje studenata iz oblasti elektrotehnike i elektronike. Aplikacija poseduje jednostavan interfejs, gde studenti mogu pristupati lekcijama i raditi testove iz oblasti koje izaberu. Na taj način u svakom trenutku mogu proveriti stečeno znanje. Još jedna od opcija jeste pregled video materijala koji su povezani sa određenim lekcijama.

Sortko [156] je Android mobilna aplikacija koja pomaže studentima u učenju algoritama za sortiranje podataka. Aplikacija se sastoji iz više modula: modul za interaktivno sortiranje koji proverava ispravnost postavljenog algoritma od strane studenta, modul za pružanje kontekstualne pomoći studentima prilikom rešavanja problema, motivacioni modul koji dodeljuje poene studentima na osnovu tačnosti i postupka rešavanja zadatka.

Blackboard Learn je veoma poznata komercijalna aplikacija namenjena edukaciji iz raznih oblasti koja je zasnovana na veb platformi, ali se može pristupiti i mobilnom aplikacijom ***Blackboard Mobile Learn***. Aplikacija podržava učestvovanje u forumima, dobijanje obaveštenja prilikom novih aktivnosti u kursu, pregled lekcija i rešavanje testova.

M-learning auto assessment [157] predstavlja mobilnu aplikaciju jednostavnog dizajna namenjena praćenju lekcija i rešavanju testova na osnovu predefinisanog sadržaja.

MOBL21 je mobilna aplikacija namenjena učenju. Poseduje pregledan interfejs, mogućnost čitanja materijala u trenutku kada korisnik nije prijavljen na sistem, dodavanje *flashcards*

termina kao podsetnika i rešavanje kvizova. Posebna prednost jeste opcija interakcije sa učesnicima slanjem SMS poruka.

M2Learn je *framework* koji se može koristiti za razvoj mobilnih *context-aware* aplikacija namenjenih učenju. Osnovne karakteristike *frameworka* su upravljanje tehnologijama lociranja (GPS), identifikacija objekata putem RFID tehnologije, kao i mogućnost povezivanja sa platformama za elektronsko učenje.

Moodle je veoma popularna platforma koja obezbeđuje integrisani sistem personalizovanog okruženja namenjenog učenju. Veb platformi je moguće pristupiti i pomoću mobilne aplikacije ***Moodle Mobile 2***. Aplikacija podržava pregled kurseva i ocena, primanje instant notifikacija, dodavanje slika, audio, video i drugih materijala sa mobilnog uređaja.

Sledi kratka analiza navedenih aplikacija.

Quizzer omogućuje interaktivno pravljenje testova, gde rešavanje pitanja može biti zavisno od lokacije (*location aware*), ali program ne pruža mogućnost čitanja i učenja iz materijala kurseva. Sa druge strane, *KnowleMobiLe* povezivanjem raznih servisa omogućuje interaktivno učenje, ali ne i testiranje studenata. Aplikacije kao što su *MEEFLS* i *Sortko* dizajnirane su samo za određenu grupu studenata, i nemaju osobine mobilnog LMS sistema, pa samim tim nisu nezavisne od vrste studija. Kod *Blackboard* aplikacije preporučuje se da instruktori prilikom manipulacije kursevima i testovima pristupaju preko veb platforme [158], a ne preko mobilne aplikacije, što samo po sebi može predstavljati ograničenje. *M-learning auto assessment* poseduje i opcije učenja i testiranja, ali je vrlo siromašan kada je u pitanju interaktivnost sa korisnicima. MOBL21 omogućuje interaktivno učenje, ali ima uočenih poteškoća u prilagođavanju interfejsa različitim mobilnim uređajima, kao i ograničenu mogućnost snimanja glasa u okviru *polling* modula. M2Learn ima prednost upotrebe raznih vrsta senzora kojim se poboljšava kvalitet učenja, ali se mogućnost integrisanja sadržaja u blogove i forume oslanja na eksternu platformu za elektronsko učenje. Moodle Mobile 2 aplikacija se potpuno oslanja na Moodle veb platformu. Ne poseduje instruktorsku podršku za manipulaciju kvizovima direktno iz aplikacije, dok se postavljanje fajlova na kurs vrši indirektno, preko privatnog repozitorijuma fajlova.

Svaka od ovih aplikacija je dobar sistem i omogućava ostvarivanje ciljeva definisanih od strane autora. Međutim, analiza ovih sistema pokazuje da se ti ciljevi razlikuju od aplikacije do aplikacije.

Analizom karakteristika prethodno opisanih aplikacija došlo se do prepoznavanja poželjne arhitekture sistema. Sistem bi trebalo da ima razne funkcionalnosti koje se mogu lako iskoristiti. Neophodno je da bude dizajniran što je "opštije" moguće, da pruži osnovne

funkcionalnosti koje nisu ograničene na samo jedan domen [159] Takođe, poželjno je da poseduje jednostavan korisnički interfejs, jer komplikovana upotreba može obeshrabriti prihvatanje novog mobilnog sistema. Pored toga, potrebno je uvažiti i mogućnosti studenata da ispune zahteve koje bi novi sistem uslovljavao (dostupnost jeftinih *smartphone* i tablet uređaja). Na osnovu svega navedenog, došlo se do prepoznavanja željenih elemenata sistema za mobilno učenje (niže je data tabela 8.4). To bi bila aplikacija na Android platformi, koja bi imala mogućnost lociranja uređaja studenata, postavljanja materijala i lekcija, rešavanja kvizova na kraju svake lekcije, mogućnost davanja audio odgovora i video prenos predavanja. Pošto nijedan od ovih sistema ne podržava sve ovo, razvijena je sopstvena mobilna platforma za učenje na VPŠSS u Blacu pod imenom mTester [160].

Tabela 8.4 - Mobilne aplikacije namenjene edukaciji

Aplikacija	Autor	Platforma	Moduli	Kviz	Lociranje	Snimanje zvuka	Video strimovanje
Quizzer	University of Duisburg-Essen, Germany	Android	Da	Ne	Da	Ne	Ne
KnowleMobiLe	Nanyang Technological University, Singapore	Java	Da	Da	Ne	Ne	Ne
MEEFLS	Universiti Teknologi MARA, Perlis , Malaysia	J2ME, XML, Adobe Flash	Ne	Da	Da	Ne	Da
Sortko	Faculty of Electrical Engineering and Computing Zagreb, Croatia	Android	Ne	Ne	Da	Ne	Ne
Blackboard Mobile Learn	Blackboard Inc	iOS Android BlackBerry webOS	Ne	Da	Da	Ne	Da
m-learning auto assessment	Computer Science Department, University of Alcal, Madrid, Spain	Java	Ne	Da	Da	Ne	Ne
MOBL21	Emantras Inc	iOS Android BlackBerry	Ne	Da	Da	Ograničeno	Da
M2Learn	Spanish University for Distance Education Madrid, Spain	Windows Mobile Android	Da	Da	Da	Ne	Da
Moodle Mobile 2	Moodle Pty Ltd	iOS Android Windows Phone	Ne	Da	Da	Ograničeno	Da
mTester	Business School of Applied Studies, Blace, Serbia	Android	Ne	Da	Da	Da	Da

mTester je Android mobilna aplikacija razvijena na Visokoj poslovnoj školi strukovnih studija u Blacu, Srbija. Zamišljena je kao mobilna LMS platforma, i nije ograničena na određenu oblast ili studijski program. Aplikacija podržava postavljanje i pregled lekcija, praćenje predavanja strimovanjem video sadržaja, zatim rešavanje vremenski ograničenih kviz pitanja, pregled datih odgovora, kao i mogućnost davanja odgovora audio snimanjem. Ovo je posebno pogodno prilikom postavljanja pitanja otvorenog tipa, gde se student ne može precizno izraziti tekstualnim putem ili gde bi odgovor morao biti obiman.

8.5. Mobilna LMS aplikacija mTester

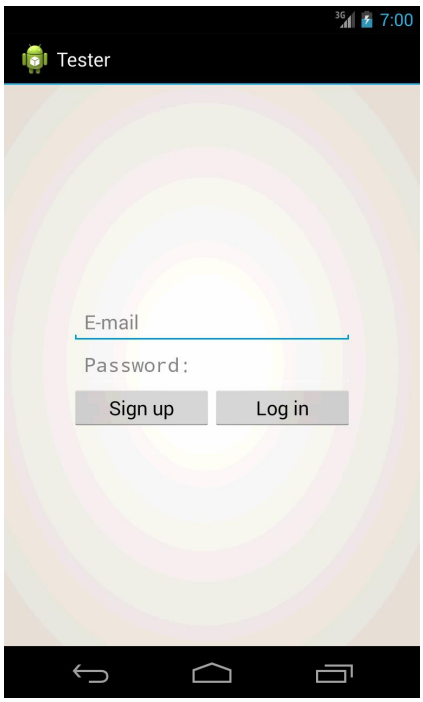
Razvijena aplikacija, pod imenom mTester, sadrži takve mehanizme koji zadovoljavaju sve navedene korake pedagoških metoda. Aplikacija je osmišljena uz uvažavanje iterativnog procesa razvoja programa. Kako bi studenti prihvatili novu mobilnu aplikaciju u najkraćem vremenskom roku, neophodno je obratiti pažnju na intuitivno korisničko iskustvo (User Experience - UX) i *look-and-feel* korisnički interfejs (User Interface - UI) [161]. Student se mora osećati prijatno prilikom korišćenja aplikacije, gde se svaka komanda nalazi tamo gde student očekuje da se nalazi, i da mu nisu potrebne značajnije smernice za upotrebu sistema. Zahvaljujući tome, omogućena je laka manipulacija i snalaženje u aplikaciji, uz zaštitu korisničkih podataka i postignutih rezultata. Predstavljeni prototip se sastoji iz dva dela: aplikacionog i sistemskog (*frontend* i *backend*). Aplikacioni deo platforme je podeljen na korisnički i instruktorski.

8.5.1. Korisnički deo platforme

Korisnički deo predstavlja grafički deo aplikacije sa kojima korisnici imaju direktan kontakt i mogućnost manipulacije u vidu preuzimanja materijala sa kurseva, puštanje video materijala ili rešavanja testova. Predavačima se omogućuje upravljanje kursevima, testovima i drugim materijalima koji su deo Tester platforme. Korisnički deo je implementiran upotrebom *Android Eclipse* platforme i *Android Developer Tools frameworka*.

Kako bi bio izvšen proces registracije i autorizacije, studenti treba da poseduju sopstvenu e-mail adresu, na koju sistem šalje e-mail poruku radi potvrde registracije. Na ovaj način se izbegava moguća zloupotreba aplikacije od automatizovane registracije nepostojećih korisnika.

Sistem razlikuje tri grupe korisnika: administrator, nastavnik i student. Svaka od grupa ima određene privilegije, u zavisnosti od dodeljenih polisa.



Slika 8.6 - Ekran za registraciju i autorizaciju korisnika

Prilikom pokretanja aplikacije, korisnik se mora prijaviti na sistem. Nakon unosa, sistem proverava ispravnost e-mail adrese i lozinke iz baze podataka. Na osnovu unetih parametara vraća se jedinstveni identifikator korisnika. Na osnovu identifikatora korisnika prepoznaje se status korisnika u sistemu (administrator, nastavnik ili student). Nakon toga, sistem šalje korisniku profilnu sliku i kreira se nova istanca klase Bundle koja se koristi za prenošenje podataka između različitih aktivnosti u aplikaciji, prenose se identifikator korisnika i njegov status.

Listing 8.1. Klasa Bundle

```
private void loger() {
try {
myMethods.connect();// Kod za konekciju na MySQL bazu

boolean check2 = logIn(txtEmail.getText().toString(), txtPassword
                .getText().toString());// Kod koji proverava ispravnost e-mail adrese i sifre, u
bazi podataka
if (check2 == true) {
        myMethods.downloadBase();

        id = myMethods.getUserId(txtEmail.getText().toString(),
                        txtPassword.getText().toString());
        status = myMethods.getStatusLogIn(id);

        String imagePath = Methods.IMAGE_PATH + myMethods.getImage(id);
        new DownloadImageTask().execute(imagePath);

        Bundle extras = new Bundle();
```

```
            extras.putInt("0", id);
            extras.putString("1", status);

            clearall();
            vrti = true;

            Intent openCourse = new Intent(
                        "com.project.tester.courses.Course");
            openCourse.putExtras(extras);
            startActivity(openCourse);

}else{
            vrti = true;
}

            myMethods.disconnect();// Kod koji se diskonektuje sa baze
} catch (Exception e) {
            // TODO: handle exception
            vrti = true;
}
}
```

Nakon prijavljivanja na sistem vrši se prenošenje podataka preko Bundle instance i startovanje nove aktivnosti, čime korisnik prelazi na sledeći ekran sa različitim grupama kurseva. U zavisnosti od vrste kurseva ili studijskih programa, moguć je odabir određene grupe kurseva.

Slika 8.7 - Biranje grupe kurseva koje će student slušati

Niže je dat segment koda kojim se kreira novi adapter (veza između komponenti interfejsa i izvora podataka koji popunjava vrednosti na interfejsu), a zatim se vrši njegovo aktiviranje. Zatim se definiše metoda koja se poziva klikom na neki element liste, čime se poziva element i jedinstveni identifikator kursa. Nakon toga se kreira nova instanca klase Bundle radi prenosa podataka u novu aktivnost i iniciranje sledećeg ekrana gde se prikazuju kursevi unutar odabrane grupe.

Listing 8.2. Kreiranje adaptera i nove instance klase Bundle

```
myAdapter = new MyAdapter(CoursForAll.this, itemImg, title, desc, editImg, deleteImg,
"cours");
listView.setAdapter((ListAdapter) myAdapter);
listView.setOnItemClickListener(new OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> adapter, View view,
                        int position, long arg) {
                Object listItem = listView.getItemAtPosition(position);

                int k_id = myMethods.getId("Kurs", listItem.toString());

                Bundle extras = new Bundle();
                extras.putInt("0", Cours.id);
                extras.putString("1", listItem.toString());
                extras.putInt("2", k_id);

                Intent openGroup = new Intent("com.project.tester.groups.Group");
                openGroup.putExtras(extras);
                startActivity(openGroup);
        }
});
```

Kada student odabere određenu grupu, prikazuje se ekran gde može odabrati koje kurseve želi da prati. Npr., ako student odabere grupu „Računarstvo i informatika“, u okviru te grupe može pratiti kurseve „Java programiranje“, „Softverski inženjering“ ili „Engleski jezik“. Unutar svakog od kurseva predavač je u mogućnosti da postavi određene materijale koji su interesantni za taj kurs. Ti materijali mogu biti prezentacije, knjige, audio i video sadržaji.

Slika 8.8 - Prikaz kursa sa postavljenim materijalima za predavanja

U sledećem delu definisan je način za dodavanje materijala na kurs, uz prikaz progress bara kako bi bio prikazan status datoteke koja se prenosi. U drugom delu koda definisano je više metoda: metoda koja se poziva kod okončanog transfera dela datoteke i koji inkrementira progress bar za određenu veličinu, metoda koja se poziva kod početka postavljanja datoteke (progress bar se podešava na nulu i meri se veličina datoteke koju treba dodati), metoda u slučaju prekida dodavanja datoteke i metoda kojom se definiše otkazivanje dodavanja datoteke od strane korisnika.

Listing 8.3. Dodavanje materijala na kurs

```
client.upload(fileName, new FTPDataTransferListener() {
        int increment = 0;
        @Override
        public void transferred(int arg0) {
                increment = increment + arg0;
                progressBar.setProgress(increment);
        }
        @Override
        public void started() {
                progressBar.setMax((int) file.length());
                progressBar.setProgress(0);
        }
        @Override
        public void failed() {
                progressBar.setProgress(0);
        }
        @Override
        public void aborted() {
                progressBar.setProgress(0);
        }
});
```

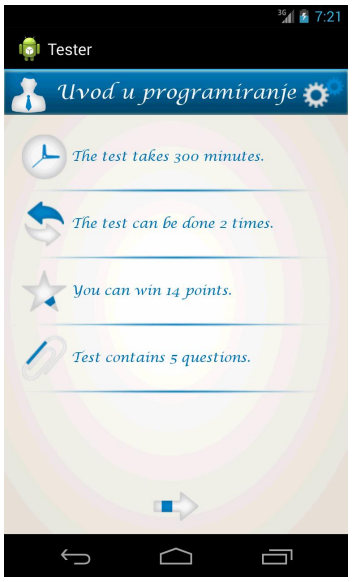
Klikom na prvu ikonicu sa spajalicom pojavljuju se materijali za odabrani kurs. Sledeća rutina omogućuje prikaz interfejsa, uz omogućavanje prikaza opcija za menjanje i brisanje elemenata u slučaju ako je na sistem prijavljen administrator ili nastavnik.

Listing 8.4. Prikaz interfejsa

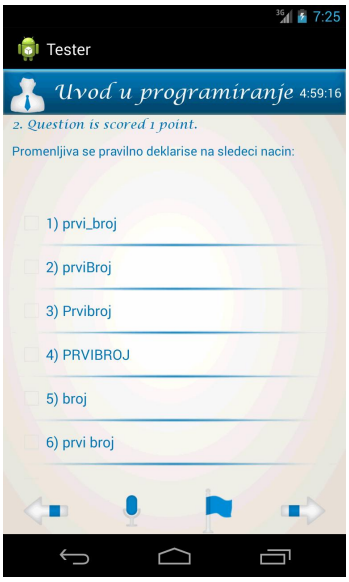
```
for (String s : contets) {
        if (!s.equals("")) {
        itemImg.add(BitmapFactory.decodeResource(getResources(),
                        R.drawable.tester_content));

        if (myContent[i].equals(String.valueOf(Group.id))) {
                editImg.add(BitmapFactory.decodeResource(
                                getResources(), R.drawable.tester_edit));
                deleteImg.add(BitmapFactory.decodeResource(
                                getResources(), R.drawable.tester_delete));
        } else {
                editImg.add(null);
                deleteImg.add(null);
        }

        title.add(s);
        desc.add(grammar(Integer.parseInt(numbOfMaterial[i])));

        i++;
        }
}
```

Slika 8.9 - Prikaz kursa sa testovima za rešavanje

Klikom na drugu ikonicu (u sredini) pokreće se prethodno postavljeni test koji studenti rešavaju. Test ima svoje vremensko ograničenje, određeni broj pitanja koji studentu daje poene, ako odgovori tačno. Odgovori se daju na jedan od tri načina, u zavisnosti od pitanja: biranjem ponuđenog odgovora, kucanjem odgovora u za to predviđeno polje ili snimanjem audio zapisa studenta u situaciji kada je neophodno šire objašnjenje pitanja, komentarisanje ili ako je to testom predviđeno.



Slika 8.10 - Prikaz uslova testa



Slika 8.11 - Prikaz jednog od pitanja na testu

Sledeći kod kreira metodu koja prikazuje pitanja. Ukoliko pitanje ima više od jednog ponuđenog odgovora kreira se interfejs za odgovor uz pomoć *checkbox* sistema, koji omogućavaju izbor više stavki, uz iteraciju svih ponuđenih odgovora. Istovremeno, sakrivaju se interfejsi za „radio“ dugmad i upisivanje teksta.

Ukoliko postoji samo jedan tačan odgovor, interfejs se kreira korišćenjem „radio“ dugmića koja omogućavaju izbor samo jednog tačnog odgovora, uz uklanjanje *checkbox* načina upisa. Ukoliko nema ponuđenih odgovora onda se koristi polje sa tekstualnim opisom.

Listing 8.5. Kreiranje metode za prikaz pitanja

```
private void prikazi(int i) {
StructQuestion pitanje;
pitanje = all.get(i);
if (pitanje.ponudjeniOdgovor.size() > 1) {
        if (pitanje.tacanOdgovor.size() > 1) {
                // CheckBox
                txtAnswer1.setVisibility(View.GONE);
                radioWork1.setVisibility(View.GONE);
                workListOfTestQuestion.setVisibility(View.VISIBLE);
```

```
                int b = 1;
                for (String po : pitanje.ponudjeniOdgovor) {
                        addQuestion(String.valueOf(b) + ") " + po);
                        b++;
                }
        } else if (pitanje.tacanOdgovor.size() <= 1) {
                // RadioButton
                txtAnswer1.setVisibility(View.GONE);
                radioWork1.setVisibility(View.VISIBLE);
                workListOfTestQuestion.setVisibility(View.GONE);
                ArrayList<String> l = new ArrayList<String>();
                int b = 1;
                for (String po : pitanje.ponudjeniOdgovor) {
                        l.add(String.valueOf(b) + ") " + po);
                        b++;
                }
                String[] l1 = l.toArray(new String[0]);
                addRadioQuestion(l1);
        }
} else if (pitanje.ponudjeniOdgovor.size() <= 1) {
        // TextBox
        txtAnswer1.setVisibility(View.VISIBLE);
        radioWork1.setVisibility(View.GONE);
        workListOfTestQuestion.setVisibility(View.GONE);
}
}
```

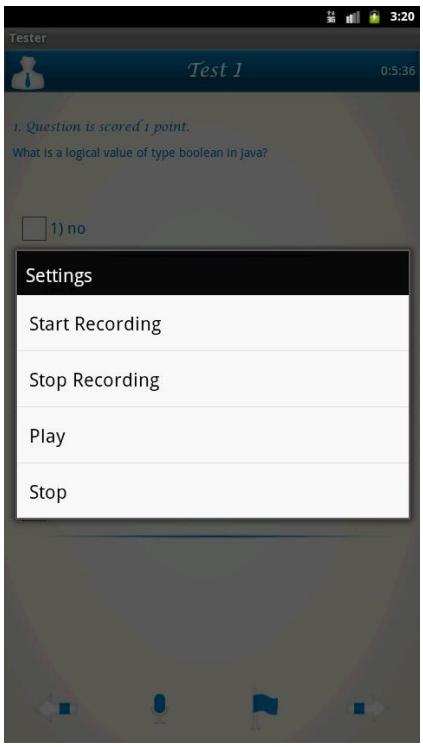
Klikom na treću ikonicu daje se pregled i uspešnost rešavanih testova.

Prilikom rešavanja testa, sa donje strane programa se nalaze 4 ikonice. Prva i poslednja ikonica omogućuju pomeranje na prethodno ili sledeće pitanje. Druga ikonica omogućuje snimanje audio snimka - glasa. Pritiskom na treću ikonicu student završava rešavanje testa.

Sledeći kod generiše datoteku sa audio zapisom, zatim inicijalizuje mikrofon na uređaju i počinje sa snimanjem.

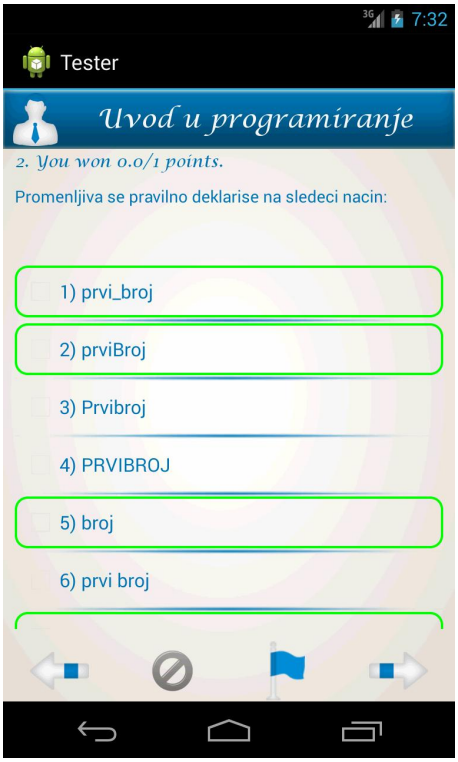
Listing 8.6. Generisanje audio datoteke

```
try {
  myMic = testNaziv + "-" + String.valueOf(brojac) + "-" + date;

  initMIC();

  myRecorder.prepare();
  myRecorder.start();
} catch (IllegalStateException e) {

} catch (IOException e) {

}
```



Slika 8.12 - Davanje audio odgovora na postavljeno pitanje

Radi smanjenja mogućnosti zloupotrebe prilikom rešavanja testova, svi testovi su vremenski ograničeni. Na taj način, student nema dovoljno vremena da pretražuje materijale vezane za test koji radi. Nakon završavanja testa, sistem ocenjuje uspešnost testa koji je student radio i omogućuje da student proveri uspešnost rešavanja svakog pitanja.



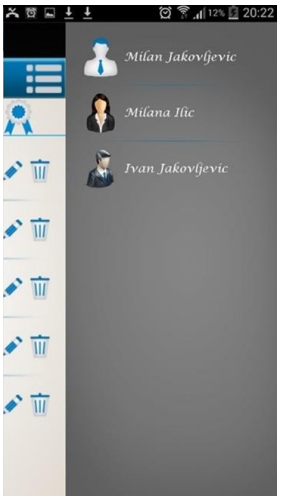
Slika 8.13 - Prikaz tačnih odgovora

Korisnički interfejs aplikacije je intuitivan i jednostavan za korišćenje. Prikaz je prilagođen i manjim i većim ekranima, a aplikacija je kompatibilna sa raznim vrstama *smartphone* i tablet uređaja.

8.5.2. Instruktorski deo platforme

Instruktor ima mogućnost definisanja i pregledanja raznih delova kurseva i testova. U okviru studijskog programa instruktor može dodati novi kurs, promeniti naziv prethodno kreiranog kursa ili obrisati kurs ukoliko prestaje potreba za njim (slika 8.14).

Slika 8.14 - Pregled kurseva

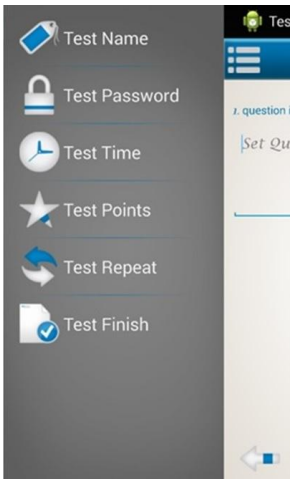


Slika 8.15 - Pregled studenata prijavljenih na kurs

Isto tako, instruktor može pregledati koji su se registrovani korisnici prijavili da slušaju određene kurseve, pregledati profil korisnika i slično (slika 8.15).

Deo predviđen za manipulaciju testovima predstavlja najobimniji instruktorski deo aplikacije. Osnovni deo predstavlja definisanje pitanja i tipova pitanja, kao i određivanja tačnih odgovora (slika 8.16).

Slika 8.16 - Dodavanje pitanja u test



Slika 8.17 - Definisanje postavki testa

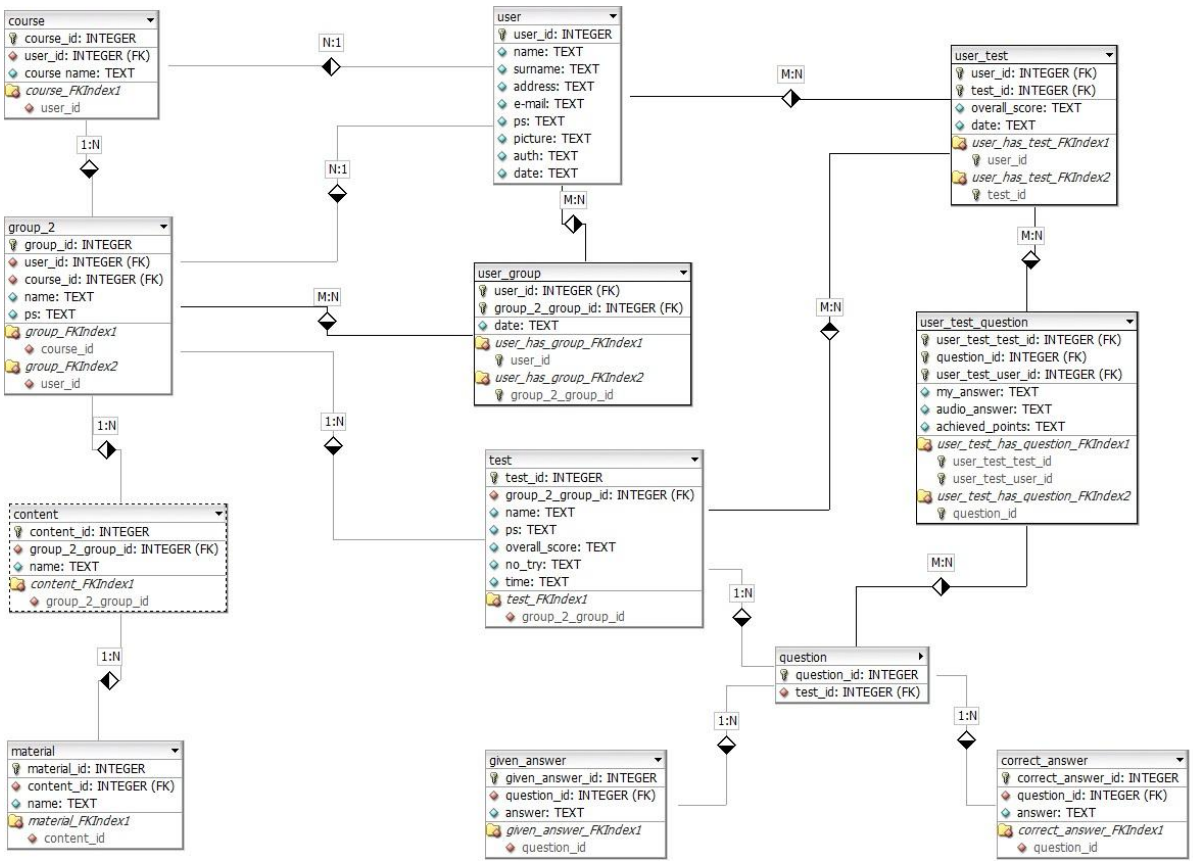
Instruktoru je pružena mogućnost imenovanja testa, postavljanje lozinke za pristup testu, određivanje dužine trajanja testa, broj poena koje student može ostvariti na testu i određivanje da li se test može rešavati više puta (slika 8.17). Opcijom Test Finish završava se definisanje svih opcija testa i objavljuje se u okviru kursa.

8.5.3. Sistemski deo platforme

Prilikom evaluacije elemenata potrebnih za izgradnju sistemskog dela platforme, moralo se voditi računa o više stvari. Osnovni uslov jeste mogućnost efikasnog povezivanja sa klijentskom stranom platforme koja je zasnovana na Android operativnom sistemu. Takođe, sistemski deo bi trebalo biti zasnovan na platformama koje su jeftine za postavljanje i održavanje. Iz tog razloga, izabrano je open-source rešenje u vidu Apache servera i MySQL baze podataka.

Sistem omogućuje pristup kroz tri nivoa korisnika – owner, admin i user. Svaki od ovih nivoa pristupa daje različite privilegije i/ili ograničenja. Owner ima apsolutno sva ovlašćenja u sistemu, može dodavati i uklanjati studijske, programe, kurseve, itd. Admin može dodati, menjati i brisati kurseve, materijale za učenje i testove. Ova vrsta naloga je predviđena za predavače, dok je korisnički nivo pristupa predviđen za krajnje korisnike sistema – studente. Oni mogu pratiti kurseve, pregledati materijale i rešavati testove.

Detaljni dijagram entiteta platforme je prikazan na sledećoj slici.



Slika 8.18 - Dijagram entiteta Tester LMS platforme

Aplikacija je pažljivo planirana, u skladu sa prethodno navedenim pedagoškim metodama, i sastoji se iz četiri jasno razdvojena dela:

Deo sa studijskim programima – predstavlja listu studijskih programa koje studenti mogu pohađati, Deo sa kursevima – omogućuje prikaz različitih kurseva koji su studentima raspoloživi u okviru studijskih programa, Deo sa lekcijama – sačinjen od lekcija u okviru svakog kursa koje postavlja predavač, i deo sa testovima – predstavlja deo svakog kursa pojedinačno, gde studenti mogu biti testirani nakon svake lekcije ili onako kako predavač kursa to odredi.

8.6. Evaluacija i rezultati

Tokom školskih godina 2007/2008. do 2014/2015. vršeno je praćenje napredovanja studenata upotrebom različitih metoda. Kako bi se izmerio uticaj mobilne platforme na učenje, bilo je neophodno formirati dve grupe – eksperimentalnu (E) i kontrolnu (K). Svaka od grupa je formirana na osnovu različitih faktora koji mogu uticati na upotrebu nove platforme u učenju: kvalitet studenata, tehnička opremljenost studenata, sklonost ka upotrebi mobilnih tehnologija, dosadašnje iskustvo upotrebe tehnologije u učenju, zainteresovanost upotrebe mobilnih uređaja u učenju i testiranju znanja. Navedeni faktori su utvrđeni anketiranjem

studenata, a kvalitet studenata je utvrđen na osnovu dosadašnjeg uspeha studenata. Uspeh studenata izražava se prosečnom ocenom koju je student postigao polaganjem ispita, i može biti od 6,00 (student je sve prethodne ispite položio sa minimalnom prolaznom ocenom 6) pa do 10,00 (student je sve prethodne ispite položio maksimalnom prolaznom ocenom 10).

Na osnovu prosečnih ocena studenata dobijenih u školskim godinama 2007/2008 – 2011/2012., svi studenti su podeljeni u četiri grupe sa prosekom studiranja od 6.00 – 6.99, 7.00 – 7.99, 8.00 – 8.99 i 9.00 – 10.00. Kao rezultat, dobijena je struktura sa određenim brojem studenata u svakoj grupi. Kako bi dobili validne rezultate, eksperimentalna i kontrolna grupa je formirana na način da svaka od njih odražava strukturu uspešnosti studenata.

Istraživanje se sastojalo iz tri etape. U prvoj etapi je urađeno anketiranje studenata radi prikupljanja podataka o stepenu opremljenosti mobilnim uređajima i njihovom korišćenju. U drugoj etapi obavljeni su testovi kojim se proveravalo znanje obe grupe studenata i analizirala upotreba mTester aplikacije. Treća etapa se sastojala iz anketiranja studenata i instruktora nakon polaganja ispita, kako bi se utvrdili utisci studenata i instruktora o korišćenju i upotrebnoj vrednosti mobilnog sistema.

U prvoj etapi obavljeno je istraživanje radi prikupljanja podataka o upotrebi mobilnih uređaja. Anketirano je 39 studenata studijskog programa Računarstvo i informatika, tokom zimskog semestra 2012. godine. Na osnovu upitnika, utvrđeno je da čak 76.92% studenata koristi smartphone uređaje sa Android operativnim sistemom, 7.69% koristi smartphone uređaje sa Windows Phone operativnim sistemom, dok 15.38% koristi starije mobilne telefone. Prilikom ispitivanja faktora koji mogu uticati na sklonost ka korišćenju mobilne aplikacije u svrhu učenja, 82.05% studenata izjavilo je da im se sviđa ideja upotrebe tehnologije u učenju, dok 79.49% koristi računare u svrhu spremanja ispita. Samo 48.72% studenata smatra da upotreba pametnih uređaja može biti od veće koristi nego upotreba računara, a 38.46% smatra da će bolje savladati kurseve uz pomoć pametnih uređaja.

Na osnovu anketiranja studenata, uočena je izvesna rezervisanost studenata prema novom načinu rada i učenja pomoću mobilnih uređaja, tako da upotreba aplikacije nije bila obavezna tokom prve dve školske godine.

Tokom poslednje tri godine sistem je upotrebljavan na nekoliko kurseva. Stepen prihvatanja je varirao, tako da su relevantni rezultati dobijeni na predmetima mobilno računarstvo, distribuirani sistemi i projekat.

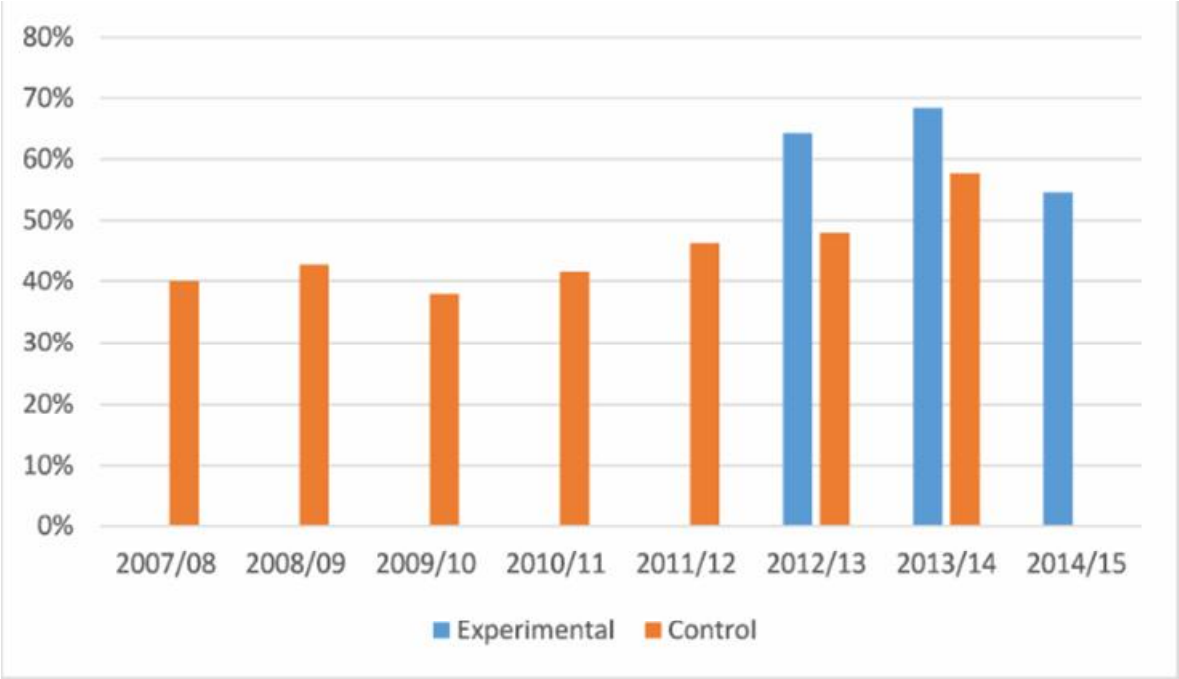
U drugoj etapi, vršena je analiza prisustva i prolaznosti eksperimentalne (M) i kontrolne (K) grupe studenata. Tabela 5 prikazuje ukupan broj studenata koji je prisustvovao kursu i položilo ispit. Tabela obuhvata i školsku 2011/2012. godinu kada aplikacija još uvek nije bila

predstavljena.

Tabela 8.5 - Prisustvo i prolaznost studenata (E-studenti koji su prihvatili sistem, K- studenti koji nisu prihvatili sistem)

Školska godina	Slušalo kurs		Položilo		%	
	E	K	E	K	E	K
2011/2012		41		19		46,34
2012/2013	14	25	9	12	64,29	48
2013/2014	19	26	13	15	68,42	57,69
2014/2015	42		23		54,76	

U drugom delu, dobijeni rezultati su upoređivani između dve grupe, kao i sa rezultatima ocenjivanja studenata u periodu pre predstavljanja sistema (Slika 8.19). Prikazano je i merenje uticaja na polaganje ispita u školskoj 2014/2015. godini, kada je upotreba aplikacije bila obavezna za sve polaznike kursa. Aplikacija se mogla koristiti bilo kada, a rešavanje testova nije bilo obavezno raditi u učionici.



Slika 8.19 - Uporedni prikaz prolaznosti studenata

Od 2008. do 2012. godine, studenti su učili na tradicionalan način, uz pomoć sistema za elektronsko učenje, ali bez upotrebe sistema mTester. Tokom tog perioda, procenat polaganja ispita je bio sličan (2008: 40%; 2009: 43%; 2010: 37%; 2011: 41%: 2012: 46%). Od 2013. na kursu je predstavljen mTester i broj studenata koji su položili ispit u prvom ispitnom roku je porastao na preko 50% (2013: 53.85%). U narednoj školskoj godini se primećuje nastavak trenda rasta prisustva i polaganja ispita (2014: 62.21%), dok istovremeno raste broj studenata koji želi da dobrovoljno koristi aplikaciju u učenju. Ovaj trend ukazuje na to da studenti imaju sve veće poverenje u nove metode učenja i da ova nova platforma daje pozitivne rezultate. U poslednjoj školskoj godini primećeno je smanjenje broja studenata koji su položili ispit u

prvom ispitnom roku. Može se zaključiti, na osnovu anketiranja i postignutih rezultata studenata, da razlog za ovo opadanje leži u nametanju obaveznosti upotrebe mTester aplikacije, jer je od poslednje školske godine upotreba aplikacije na kursu bila obavezna. Ovakav pristup je verovatno izazvao averziju kod studenata, što je na kraju uticalo na nešto lošije rezultate studenata. Ipak, može se primetiti da je u poslednjoj školskoj godini procenat prolaznosti veći u poređenju sa školskim godinama kada se mTester sistem nije upotrebljavao.
U trećoj etapi, koja se odvijala nakon testiranja, izvršeno je anketiranje studenata iz eksperimentalne grupe i angažovanih instruktora nizom izjava, kako bi se utvrdili utisci o korišćenju i upotrebnoj vrednosti mobilnog sistema. Od ukupnog broja anketiranih studenata, 92.86% se slaže sa izjavom da tehnologija može pomoću u poboljšanju učenja, a 85.71% misli da bi spremanje ispita bilo lakše uz upotrebu mobilnih uređaja. Na izjavu "Test na mobilnom uređaju je bio mnogo zanimljiviji nego na papiru ili računaru", 78.57% je odgovorilo potvrdno, dok bi 85.71% anketiranih studenata preporučilo mTester drugim studentima. Studenti su tokom upotrebe aplikacije ukazivali na nedostatke sistema, jer nisu bili u potpunosti zadovoljni dizajnom korisničkog interfejsa, obaveznošću upotrebe aplikacije u poslednjoj školskoj godini i nemogućnošću povezivanja sa društvenim mrežama.
Pored anketiranja studenata, izvršeno je i anketiranje instruktora kako bi se ispitali pozitivni i negativni efekti aplikacije. 50% anketiranih instruktora se slaže sa izjavom da su im predavanja izvođena uz pomoć mobilne aplikacije zanimljivija. 66.66% ispitanih smatra da aplikacija smanjuje vremem potrebno za pripremu testova. Primećeno je da prihvatanje mobilnog sistema može biti važan izazov i za instruktore. Instruktor A je prokomentarisao da nije imao adekvatnu tehničku podršku prilikom problema sa infrastrukturom institucije, tako da je povezivanje bilo moguće samo preko 3G mobilne mreže. Drugi instruktor je izjavio da mu dosta vremena oduzima preslušavanje audio odgovora studenata, dok je treći prokomentarisao da je mobilni internet skup u uslovima kada nema WiFi pristupa.
Zahvaljujući pozitivnim iskustvima autora i povećanoj prolaznosti studenata na ispitu, 75% anketiranih instruktora izrazilo je spremnost da u narednom periodu uvrsti aplikaciju na svoj kurs. Takođe, iz prethodnih rezultata se može uvideti da uvođenje inovativnih metoda učenja, gde studenti mogu pristupiti sadržaju kursa i rešavati testove van učionice, dovodi do povećanja uspešnosti studenata. Student može biti bilo gde i nesmetano pratiti predavanja, pod uslovom da mobilni uređaj (*smartphone* ili tablet) studenta ima 3G ili WiFi vezu sa Internetom. Jedan od najvećih problema kod platforme za mobilno učenje jeste relativno mali kapacitet memorije korisničkih uređaja, a posebno kratka autonomija baterija uređaja krajnjih korisnika. Kako bi se autonomija produžila, aplikacija omogućuje da se prilikom pregleda

video materijala sadržaj direktno strimuje korisniku. Na taj način se izbegava dodatni utrošak energije, jer se preskače korak preuzimanja video materijala na memoriju uređaja korisnika, pa naknadnog pokretanja u okviru mTester aplikacije.

Primenom najsavremenijih programskih rešenja, uspešno je povezan proces modernog načina obrazovanja sa *cloud* sistemima. Uz to, korišćenjem mobilnih uređaja koji su poznati kao mali potrošači, napravljena je značajna ušteda električne energije, čime je primenjen koncept energetske efikasnosti. Prema tome, ova nova metoda mobilnog učenja se pokazala boljom od tradicionalnih metoda učenja koje se izvode isključivo u učionici.

8.7. Doprinos

Na osnovu dosadašnjih naučno istraživačkih iskustava iz ove oblasti, kao i istraživanja u ovoj doktorskoj disertaciji, predložene su nove metode u procesu učenja i ocenjivanja studenata, čijom se upotrebom povećava učinak u edukaciji.

Ostvareni doprinosi su:

- Pregled i analiza dosadašnjih istraživanja iz oblasti *cloud computing* tehnologije,
- Pregled i analiza postojećih energetski efikasnih polisa i algoritama,
- Pregled *cloud computing* platformi,
- Pregled energetski efikasnih hipervizora,
- Pregled i evaluacija energetski efikasnih klijentskih rešenja koja se mogu koristiti u učenju
- Pregled i analiza bezbednosnih problema i rešenja koja se mogu primeniti na *cloud computing*,
- Urađena je studija i analiza teorije mobilnog učenja,
- Analizirane su ključne karakteristike,
- Analizirani su faktori koji stvaraju stimulanse za mobilno učenje,
- Analizirane su trenutne perspektive i bitni elementi,
- Definisani su argumenti za i protiv mobilnog učenja.

Takođe, studija o mobilnom učenju u ovoj tezi može u pozitivnom smislu da utiče na percepciju roditelja o mobilnoj generaciji koja se sastoji od učenika svih uzrasta. Ona takođe ukazuje na potrebu angažovanja studenata u mobilnom učenju tokom radnih i neradnih dana čime ova tehnologija i socijalno umrežavanje postaju deo školske svakodnevice.

- Analizirani su principi i šabloni dizajna interfejsa. Date su smernice i uputstva, što može olakšati posao novim programerima u kreiranju dobrih aplikacija. Takođe, ova

analiza može pomoći projektantima da shvate i uzmu u obzir nekoliko važnih stvari kada žele da projektuju novu mobilnu aplikaciju ili prenesu desktop aplikaciju u mobilno okruženje.

- Urađena je analiza potencijalnih mobilnih platformi koje se mogu koristiti za izvršavanje mobilne aplikacije.
- Predočena je taktika za rešavanje zajedničkih problema kod razvoja Android aplikacija koja može poslužiti kao vodič novim programerima.
- Kreirana je potpuno funkcionalna Android aplikacija koja može da podrži učenje na daljinu.

Glavne prednosti novostvorene aplikacije su:

- Projektovana je da podstakne studente da budu aktivniji i motivisaniji tokom nastavnog perioda.
- Trenutno, nema mnogo mobilnih aplikacija koje podržavaju učenje na daljinu ili koje se mogu koristiti tokom nastavnog perioda.
- Pošto se radi o Android aplikaciji, može se instalirati direktnim preuzimanjem sa veb sajta obrazovne ustanove na mobilni uređaj zasnovanom na Android operativnom sistemu.
- Izvorni kod mogu koristiti i drugi programeri da prošire postojeće funkcije ili dodaju nove kako bi se poboljšao obrazovni proces.
- Urađena je evaluacija praktične upotrebe aplikacije. Procena je zasnovana na testiranju funkcionalnih zahteva i korisničkog interfejsa.

9. ZAKLJUČAK

U poslednjih nekoliko godina, *cloud computing* predstavlja evoluciju internet servisa koji ubrzava inovacije u kompjuterskoj industriji. *Cloud computing* je računarski model zasnovan na mrežama, naročito na Internetu, čiji je zadatak da korisnici mogu jednostavno da koriste računarske resurse na zahtev i plate u skladu sa njihovim korišćenjem.

U integraciji elektronskog učenja i interneta, naglasak je stavljen na izgradnju hardverskih i softverskih platformi za elektronski sistem učenja, funkcionalnu strukturu, upravljanje bezbednošću računarskih mreža i obuku, integraciju informacionih tehnologija u nastavi, onlajn obrazovanje i implementaciju rešenja za elektronsko i mobilno učenje u savremenoj elektronskoj učionici.

U ovom radu definisane su tehnologije za elektronsko učenje, zatim, potencijalna integracija tehnologije elektronskog učenja u *cloud* platformu. Razmatrane su metode virtuelizacije, delimična, para i potpuna virtuelizacija, njihova povezanost, implementacija i međusobni uticaj.

Kao poseban deo, opisan je projekat unapređenja energetske efikasnosti *cloud computing* infrastrukture u svrhu obrazovanja.

Kako bi bilo detaljno prikazano koje je najefikasnije rešenje koje se može praktično primeniti u jednoj elektronskoj učionici, vršena je analiza komponenti *cloud* sistema i njihov uticaj na potrošnju energije. Obavljena je analiza energetski efikasnih klijentskih rešenja namenjenih učenju. Upoređivana je potrošnja električne energije i ostalih troškova PC računara, tankih klijenata i *smartphone*/tablet uređaja. Utvrđeno je da je najefikasnije rešenje upotreba *smartphone*/tablet uređaja.

Mobilno učenje predstavlja jedno od najperspektivnijih trendova u sferi obrazovanja. U radu su izučavane i mobilne aplikacije namenjene učenju. Analizom raspoloživih sistema utvrđeno je da svaki od sistema koristi samo neke od mogućnosti i da nijedan od njih ne pruža sve mogućnosti koje bi bile poželjne. Stoga su na osnovu kritičke analize raspoloživih sistema i sopstvenih iskustava u primeni elektronskog učenja, najpre definisane željene karakteristike jednog ovakvog sistema, zatim razvijen mobilni sistem za učenje mTester i na kraju je izvršena evaluacija efekata primene ovog sistema. Iz navedenih rezultata se može videti da uvođenje inovativnih metoda učenja, gde studenti mogu pristupiti sadržaju kursa i rešavati testove van učionice, dovodi do povećanja uspešnosti studenata.

Dakle, uz uporednu analizu energetske efikasnosti različitih potencijalnih sistema koji se mogu koristiti u edukaciji, kao i uz praktičnu softversku implementaciju namenjenu učenju u

mobilnom okruženju, prikazano je jedno kompletno rešenje koje se može koristiti za unapređenje savremene nastave.

U ovom trenutku mogućnost prepoznavanja lokacije korisnika nije umanjila efikasnost mTester sistema. Ipak, nesumnjivo je da bi se na taj način proširile mogućnosti sistema u učenju, a to je jedna od opcija koja je planirana za budući razvoj. Prihvatanje sistema je od velike važnosti, tako da će u budućem periodu biti neophodno omogućavanje dostupnosti na svim mobilnim platformama. Pored navedenog, predviđena je modifikacija dizajna mobilne aplikacije i uvođenje mogućnosti povezivanja sa društvenim mrežama, zatim, smanjenje mogućnosti zloupotrebe, optimizacija transfera podataka, kao i povećanje energetske efikasnosti celokupnog sistema.

LITERATURA

[1] R. Buyya, J. Broberg, and A. M. Goscinski, *Cloud Computing: Principles and Paradigms*, John Wiley & Sons, Hoboken, NJ, 2010.

[2] T. L. Wentling, C. Waight, J. Gallaher, J. Fleur, C. Wang, and A. Kanfer, E-learning - A Review of Literature, Knowledge and Learning Systems Group. NCSA. Urbana, IL, USA: Univ. Illinois at Urbana-Champaign, 2000.

[3] O'Malley, C., Vavoula, G., Glew, J. P., Taylor, J., Sharples, M., Lefrere, P., Lonsdale, P., Naismith, L., and Waycott, J. (2005) WP4—Guidelines for Learning/Teaching/Tutoring in a Mobile Environment. MOBILearn.http://www.mobilearn.org/download/results/guidelines.pdf

[4] Velev, D. G. Challenges and Opportunities of Cloud-Based Mobile Learning. *International Journal of Information & Education Technology*, *4*(1), 2014

[5] Y. M. Huang and P. S. Chiu, "The effectiveness of a meaningful learning-based evaluation model for context-aware mobile learning," Brit. J. Educ. Techn, doi: 10.1111/bjet.12147, 2014

[6] Daryl C. Plummer, Thomas J. Bittman, Tom Austin, David W. Cearley, and David Mitchell Smith, "Cloud Computing: Defining and Describing an Emerging Phenomenon," Gartner, June 17, 2008.

[7] Frank E. Gillett with Eric G. Brown, James Staten, and Christina Lee, "Future View: The New Tech Ecosystems Of Cloud, Cloud Services, and Cloud Computing: Understanding, Segmenting, and Competing in the Next Computer Revolution," Forrester Research, August 28, 2008.

[8] Badger, L, Grance, T, Patt-Corner, R and Voas, J. *Cloud Computing Synopsis and Recommendations*, National Institute of Standards and Technology, USA, 2012

[9] Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., & Leaf, D. (2011). NIST cloud computing reference architecture. *NIST special publication*, *500*(2011), pp. 292.

[9] Zhou Peng, Li Mei, Wang Fei, and Yin Fei, "The analisis of Gis software engineering pattern under the cloud computing environment," IEEE-ICEIT 2010, V2, pp. 450-452.

[10] Amazon, www.amazon,com, posećeno 02.02.2016.

[11] Lydia Leong, Douglas Toombs, Bob Gill, *Magic Quadrant for Cloud Infrastructure as a Service*, Worldwide, Gartner, 2015

[12] Trend Micro, *Top 10 AWS Security Tips*, dostupno na http://blog.trendmicro.com/top-10-aws-security-tips-5-create-restrictive-firewall-policies/, 2013, posećeno 02.02.2016.

[13] https://cloud.google.com/appengine/docs

[14] https://azure.microsoft.com/en-us/

[15] Welicki, Leon. "Announcing Azure Portal general availability", *https://azure.microsoft.com/en-us/blog/announcing-azure-portal-general-availability/ Microsoft.* posećeno 20.01. 2016.

[16] http://www.vasanth.in/content/binary/Windows%20Azure.jpg

[17] http://vcloud.vmware.com/explore-vcloud-air/what-is-vcloud-air

[18] Projekat Eucalyptus, http://www.eucalyptus.com/, posećeno 02.02.2016.

[19] "Eucalyptus Components" by Source (WP:NFCC#4). Licensed under Fair use via Wikipedia - https://en.wikipedia.org/wiki/File:Eucalyptus_Components.png#/media/File: Eucalyptus_Components.png

[20] Open Nebula, http://www.opennebula.org, posećeno 02.02.2016.

[21] D. Milojicic, I.M. Llorente, and R.S. Montero, "OpenNebula: A Cloud Management Tool,"IEEE Internet Computing,vol. 15, no. 2,pp. 11-14, Mar./Apr. 2011.

[22] Nimbus, http://www.nimbusproject.org, posećeno 01.03.2016.

[23] OpenStack, https://www.openstack.org/, posećeno 01.03.2016.

[24] Ken pepple july 2011. Deploying OpenStack. Oreilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472

[25] G. von Laszewski, J. Diaz, F. Wang, and G.C. Fox, "Comparison of Multiple Cloud frameworks,"Proc. IEEE Conf. Cloud Computing, pp. 734-741, 2012.

[26] http://en.wikipedia.org/wiki/Virtualization

[27] http://en.wikipedia.org/wiki/Hypervisor

[28] Wang, C.; Wang, Q.; Ren, K.; Lou, W. Towards secureand dependable storage services in cloud computing. IEEE Trans. Serv. Comput. 2012, 5, 220–232.

[29] Morin, J.; Aubert, J.; Gateau, B. Towards cloud computing SLA risk management: Issues and challenges. In Proceedings of the 2012 45th Hawaii International Conference on System Science (HICSS), Maui, HI, USA, 4–7 January 2012; pp. 5509–5514.

[30] Khalil, I.M. ELMO: Energy aware localmonitoring in sensor networks. IEEE Trans. Dependable Secur. Comput. 2011, 8, 523–536.

[31] Popovic, O.; Jovanovic, Z.; Jovanovic, N.; Popovic, R. A comparison and security analysis of the cloud computing software platforms. In Proceedingsof the 2011 10th International Conference on Telecommunication in Modern Satellite Cable and Broadcasting Services (TELSIKS), Nis, Serbia, 5–8 October 2011; Volume 2, pp. 632–634.

[32] Sengupta, S.; Kaulgud, V.; Sharma, V.S. Cloud computing security—Trends and research directions. In Proceedings of the 2011 IEEE World Congress on Services (SERVICES), Washington, DC, USA, 4–9 July 2011; pp. 524–531.

[33] Gul, I.; ur Rehman, A.; Islam, M.H. Cloud computing security auditing. In Proceedings of the 2011 The 2nd International Conference on Next Generation Information Technology (ICNIT), Gyeongju, Korea, 21–23 June 2011; pp. 143–148.

[34] Khalil, I. M., Khreishah, A., & Azeem, M. Cloud computing security: a survey. *Computers*, *3*(1), 2014, pp.1-35.

[35] Fangfei, Z.; Goel, M.; Desnoyers, P.; Sundaram, R. Scheduler vulnerabilities and coordinated attacks in cloud computing. In Proceedings ofthe 2011 10th IEEE International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 25–27 August 2011; pp. 123–130.

[36] Somani, G., Gaur, M. S., Sanghi, D., Conti, M., & Buyya, R. (2015). DDoS Attacks in Cloud Computing: Issues, Taxonomy, and Future Directions. *arXiv preprint arXiv:1512.08187*.

[37] Dimitrios Zissis and Dimitrios Lekkas. 2012. Addressing cloud computing security issues. Future Generation Computer Systems 28, 3 (2012), 583 – 592. DOI:http://dx.doi.org/10.1016/j.future.2010.12.006

[38] Karnwal, T.; Sivakumar, T.; Aghila, G. A comberapproach to protect cloud computing against XML DDoS and HTTP DDoS attack. In Proceedings of the 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 1–2 March 2012; pp.1–5.

[39] Chris Burt. *Large Volume DDoS Attacks See Exceptional Growth in First Half of 2014*: Arbor Networks. 2014. dostupno na: http://www.thewhir.com/web-hosting-news/large-volume-ddos-attacks-see-exceptional-growth-first-half-2014-arbor-networks. posećeno 04.02.2016.

[40] Kaspersky Labs. 2014. GLOBAL IT SECURITY RISKS SURVEY 2014 DISTRIBUTED DENIAL OF SERVICE (DDoS) ATTACKS. http://media.kaspersky.com/en/B2B-International-2014-Survey-DDoS-Summary-Report.pdf.

posećeno 04.02.2016.

[41] Mahalingam, Sankara M. "Cloud Based Security Center: To Protect Networking Attack by Forensic Scrutiny." *International Journal of Computer Science and Network Security (IJCSNS)* vol.4, no.3, 2015, pp. 280-284..

[42] Zeus Bot Found Using Amazon's EC2 asC&C Server. Web adresa:

http://www.theregister.co.uk/2009/12/09/amazon ec2bot control channel/, posećeno: 23.02.2016.

[43] Google Cloud Platform Used for Botnet Control. Web adresa: http://www.infosecuritymagazine.com/view/5115/google-cloud-platform-used-for-botnet-control/, posećeno: 23.02.2016.

[44] Raytheon UK Targeted in Cloud-Based Attack. Web adresa:

http://www.zdnet.co.uk/news/security-threats/2011/10/12/raytheon-uk-targeted-in-cloud-basedattack-40094173/, posećeno 23.02.2016.

[45] Lin, W.; Lee, D. Traceback attacks in cloud—Pebbletrace botnet. In Proceedings of the 2012 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW), Macau, China, 18–21 June 2012; pp. 417–426

[46] Karnwal, T.; Sivakumar, T.; Aghila, G. A comberapproach to protect cloud computing against XML DDoS and HTTP DDoS attack. In Proceedings of the 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 1–2 March 2012; pp.1–5.

[47] Riquet, D.; Grimaud, G.; Hauspie, M. Large-scale coordinated attacks: Impact on the cloud security. In Proceedings of the 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Palermo, Italy, 4–6 July 2012; pp. 558–563

[48] Snort, http://www.snort.org, posećeno 20.02.2016.

[49] J. Lindemann. 2015. Towards Abuse Detection and Prevention in IaaS Cloud Computing. In Availability, Reliability and Security (ARES), 2015 10th International Conference on. 211–217. DOI:http://dx.doi.org/10.1109/ARES.2015.72

[50] Gruschka, N.; Jensen, M. Attack surfaces: A taxonomy for attacks on cloud services. In Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), Miami, FL, USA, 5–10 July 2010; pp. 276–279

[51] Liu, S.-T.; Chen, Y.-M. Retrospective detection of malware attacks by cloud computing. In Proceedings of the 2010 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Huangshan, China, 10–12 October 2010; pp. 510–517

[52] http://hadoop.apache.org/, posećeno 01.03.2016.

[53] Singh, A., & Shrivastava, M. (2012). Overview of attacks on cloud computing. *International Journal of Engineering and Innovative Technology (IJEIT)*, *1*(4).

[54] Oberheide, J.; Cooke, E.; Jahanian, F. CloudAV: N-version antivirus in the network cloud. In Proceedings of the 17th Conference on SecuritySymposium (SS '08); USENIX Association: Berkeley, CA, USA, 2008; pp. 91–106.

[55] Aviram, A.; Hu, S.; Ford, B.; Gummadi, R. Determinating timing channels in compute clouds. In Proceedings of the 2010 ACM Workshop on Cloud Computing Security Workshop (CCSW '10);

ACM: New York, NY, USA, 2010; pp. 103–108.

[56] Qiasi Luo and Yunsi Fei, Algorithmic Collision Analysis for Evaluating Cryptographic System and Side- Channel Attacks, International Symposium on Hardware-Oriented Security and Trust, 2011.

[57] Hlavacs, H.; Treutner, T.; Gelas, J.; Lefevre, L.; Orgerie, A. Energy consumption side-channel attack at virtual machines in a cloud. In Proceedings of the 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC), Sydney, NSW, Australia, 12–14 December 2011; pp. 605–612.

[58] Hlavacs, H.; Treutner, T.; Gelas, J.; Lefevre, L.; Orgerie, A. Energy consumption side-channel attack at virtual machines in a cloud. In Proceedings of the 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC), Sydney, NSW, Australia, 12–14 December 2011; pp. 605–612

[59] Gunasekhar, T., et al. "Mitigation of Insider Attacks through Multi-Cloud." *International Journal of Electrical and Computer Engineering* 5.1 (2015): pp. 136-141.

[60] www.cert.com/insider-threat, posećeno 01.03.2016.

[61] Rocha, F.; Correia, M. Lucy in the sky without diamonds: Stealing confidential data in the cloud. In Proceedings of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSNW '11), Hong Kong, China, 27–30 June 2011; IEEE Computer Society: Washington, DC, USA, 2011; pp. 129–134

[62] *Top Threats to Cloud Computing V1.0*; Cloud Security Alliance: March 2010.

[63] Khalil, I. M., Khreishah, A., & Azeem, M. Cloud computing security: a survey. *Computers*, *3*(1), 2014, pp.1-35.

[64] Szefer, J.; Lee, R.B. Architectural support for hypervisor-secure virtualization. SIGARCH Comput. Arch. News 2012, 40, 437–450

[65] Te-Shun, C. Security threats on Cloud Computing vulnerabilities, International Journal of Computer Science & Information Technology (IJCSIT) Vol 5, No 3, June 2013

[66] Tupakula, U.; Varadharajan, V.; Akku, N. Intrusion detection techniques for infrastructure as a service cloud. In Proceedings of the 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC), Sydney, Australia, 12–14 Dec. 2011; pp. 744–751.

[67] Vieira, K.; Schulter, A.; Westphall, C.B.; Westphall,C.M. Intrusion detection for grid and cloud computing. IT Prof. 2010, 12, 38–43

[68] Wang, D.; Zhou, Z. Application of cloud model in intrusion detection. In Proceedings of the 2010 2nd International Conference on e-Business and Information SystemSecurity (EBISS), Wuhan, China, 22–23 May 2010; pp. 1–4.

[69] https://snort.org/downloads, posećeno 20.02.2016.

[70] http://manual.snort.org/node3.html, posećeno 20.02.2016.

[71] https://www.wireshark.org/, posećeno 20.02.2016.

[72] Manesh, T., El-atty, S. M. A., Sha, M. M., Brijith, B., & Vivekanandan, K. Forensic investigation framework for VoIP protocol. IEEE *Anti-Cybercrime (ICACC), 2015*. pp. 1-7.

[73] K. Vieira, A. Schulter, and C. Westphall, "Intrusion Detection for Grid and Cloud Computing," IT Professional, vol. 12, pp. 38-43, 2010

[74] Zargar, S.T.; Takabi, H.; Joshi, J.B.D. DCDIDP: A distributed, collaborative, and data-driven intrusion detection and prevention framework for cloud computing environments. In Proceedings of the 2011 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), Orlando, FL, USA, 15–18 October 2011; pp. 332–341.

[75] Kannadiga, P.; Zulkernine, M. DIDMA: A distributed intrusion detection system using mobile agents. In Proceedings of the Sixth International Conference on SoftwareEngineering, Artificial Intelligence, Networking and Parallel/DistributedComputing, and First ACIS International Workshop on Self-Assembling Wireless Networks, 23–25 May 2005; pp. 238–245

[76] El Mourabit, Y.; Toumanari, A.; Zougagh, H. A mobile agent approach for IDS in Mobile Ad Hoc Network. International Journal of Computer Science Issues, Vol. 11, Issue 1, No 1, January 2014; pp. 148-152

[77] Doelitzscher, F.; Reich, C.; Knahl, M.; Clarke, N. An autonomous agent based incident

detection system for cloud environments. In Proceedings of the 2011 IEEE Third International

Conference on Cloud Computing Technology and Science (CloudCom), Athens, Greece,

29 November–1 December 2011; pp. 197–204.

[78] Balen, D.; Westphall, C.; Westphall, C. Experimental assessment of routing for grid and cloud. In Proceedings of the Tenth International Conference on Networks (ICN 2011); St. Maarten, The Netherlands Antilles, January 23-28, 2011, pp. 341–346

[79] Sodhi, B.; Prabhakar, T.V. A cloud architecture using smart nodes. In Proceedings of the 2011 IEEE Asia-Pacific Services Computing Conference (APSCC), Jeju Island, Korea, 12–15 December 2011; pp. 116–123

[80] Yazir, Y.O.; Matthews, C.; Farahbod, R.; Neville, S.; Guitouni, A.; Ganti, S.; Coady, Y. Dynamic resource allocation in computing clouds using distributed multiple criteria decision

analysis. In Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), Miami, FL, USA, 5–10 July 2010; pp. 91–98.

[81] Mareschal, B. Aide a la Decision Multicritere: Developpements Recents des Methodes PROMETHEE; Cahiers du Centre d'Etudes en Recherche Operationelle: Bruxelles, Belgium, 1987; pp. 175–241

[82] Bishop, M. Computer Security: Art and Science; Addison-Wesley Professional: Reading, MA, USA, 2002

[83] Leandro, M.A.P.; Nascimento, T.J.; dos Santos, D.R.; Westphall, C.M.; Westphall, C.B. Multitenancy authorization system with federated identity for cloud-based environments using shibboleth. In Proceedings of the Eleventh International Conference on Networks, 2012; pp. 88–93.

[84] Sanchez, R.; Almenares, F.; Arias, P.; Diaz-Sanchez, D.; Marin, A. Enhancing privacy and dynamic federation in IdM for consumer cloud computing. IEEE Trans. Consum. Electron. 2012, 58, 95–103.

[85] Leandro, M.A.P.; Nascimento, T.J.; dos Santos, D.R.; Westphall, C.M.; Westphall, C.B. Multitenancy authorization system with federated identity for cloud-based environments using shibboleth. In Proceedings of the Eleventh International Conference on Networks, 2012; pp. 88–93.

[86] Jansen, Ryan, and Paul R. Brenner. "Energy efficient virtual machine allocation in the cloud." *Green Computing Conference and Workshops (IGCC), 2011 International*. IEEE, 2011

[87] R. Miller, "Strong Growth for Amazon EC2 in Ireland". Data Center Knowledge, http://www.datacenterknowledge.com, 2010.

[88] M. Eisen Marcum Technology, Introduction to Virtualization,"The Long Island", Chapter of the IEEE Circuits and Systems (CAS) Soc.,Apr. 2011.

[89] Garg, Saurabh Kumar, et al. "Energy-efficient scheduling of HPC applications in cloud computing environments." *arXiv preprint arXiv:0909.1146* (2009).

[90] K. Kim, A. Beloglazov, R. Buyya, "Power-aware Provisioning of Cloud Resources for Real-time Services," Middleware for Grids, Clouds, and e-Science (MGC), Urbana-Champaign, Illinois, 2009

[91] GRAUBNER, Pablo; SCHMIDT, Matthias; FREISLEBEN, Bernd. Energy-efficient virtual machine consolidation. *IT Professional*, 2013, 15.2: 0028-34

[92] DONG, Jiankang, et al. Virtual Machine Placement for Improving Energy Efficiency and Network Performance in IaaS Cloud. In: *Distributed Computing Systems Workshops (ICDCSW), 2013 IEEE 33rd International Conference on*. IEEE, 2013. p. 238-243

[93] HWANG, Inkwon; PEDRAM, Massoud. Hierarchical Virtual Machine Consolidation in a Cloud Computing System. In: *Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on*. IEEE, 2013. p. 196-203

[94] DONG, Jiankang, et al. Energy-Saving Virtual Machine Placement in Cloud Data Centers. In: *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*. IEEE, 2013. p. 618-624

[95] Malik, S., S. Khan, and S. Srinivasan, "Modeling and Analysis of State-of-the-art VM-based Cloud Management Platforms", IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 1, NO. 1, 2013

[96] S. Srikantaiah, A. Kansal, F. Zhao, “Energy Aware Consolidation for Cloud Computing”. HotPower'08 Proc. of the 2008 Workshop on Power Aware Computing and Systems, San Diego, California, 2008.

[97] Jansen, Ryan, and Paul R. Brenner. "Energy efficient virtual machine allocation in the cloud." *Green Computing Conference and Workshops (IGCC), 2011 International*. IEEE, 2011.

[98] J. Koomey, “Growth in data center electricity use 2005 to 2010,” Analytics Press, Oakland, CA, Tech. Rep., 2011

[99] Linux Foundation, “The Xen Project, the powerful open source industry standard for virtualization,” Dostupno na: http://www.xenproject.org/

[100] Scott Lowe, Mastering VMware Vsphere 5, Wiley India/Goels Computer Hut, (2012)

[101] Guzek, Mateusz, et al. "A Holistic Model of the Performance and the Energy-Efficiency of Hypervisors in an HPC Environment." *Energy Efficiency in Large Scale Distributed Systems*. Springer Berlin Heidelberg, 2013. 133-152.

[102] Guzek, Mateusz, et al. "A Holistic Model of the Performance and the Energy-Efficiency of Hypervisors in an HPC Environment." *Energy Efficiency in Large Scale Distributed Systems*. Springer Berlin Heidelberg, 2013. 133-152.

[103] Jin, Yichao, Yonggang Wen, and Qinghua Chen. "Energy efficiency and server virtualization in data centers: An empirical investigation." *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*. IEEE, 2012.

[104] J. Che, et al., „A synthetical performance evaluation of OpenVZ, Xen and KVM“. IEEE Services Computing Conf., China, 2010.

[105] Jin, Yichao, Yonggang Wen, and Qinghua Chen. "Energy efficiency and server virtualization in data centers: An empirical investigation." *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*. IEEE, 2012

[106] Fayyad-Kazan, H., Perneel, L. and Timmerman, M., 2013. Benchmarking the Performance of Microsoft Hyper-V server, VMware ESXi and Xen Hypervisors. *Journal of Emerging Trends in Computing and Information Sciences*, *4*(12), pp.922-933.

[107] Yenké, B.O., Ari, A.A., Mbeuyo, C.D. and Voundi, D.A., 2015. Virtual Machine Performance upon Intensive Computations. *GSTF Journal on Computing (JoC)*, *4*(3), p.98.

[108] L. Caviglione, A. Merlo, and M. Migliardi. What is green security? InThe 7th International Conference on Information Assurance and Security (IAS), pages 366–371. IEEE, 2011.

[109] L. Caviglione, A. Merlo, and M. Migliardi. Green-aware security: Towards a new research field. The International Journal of Information Assurance and Security (JIAS), 7:338–346, 2012.

[110] M. Migliardi and A. Merlo. Modeling the energy consumption of distributed ids: A step towards green security. In MIPRO, 2011 Proceedings of the 34th International Convention, pages 1452–1457. IEEE, 2011.

[111] S. Chhabra and Y. Solihin. Transactions on computational science X. chapter Green secure processors: towards powerefficient secure processor design, pages 329–351. SpringerVerlag, Berlin, Heidelberg, 2010

[112] McAfee, ICF. The carbon footprint of email spam report. 2008. http://resources.mcafee.com/content/NACarbonFootprintSpam.

[113] http://www.esecurityplanet.com/network-security/almost-100-billion-spam-e-mails-sent-daily-in-q1-2013.html, posećeno 04.03.2015.

[114] http://www.symantec.com/security_response/landing/spam/, posećeno 17.01.2016.

[115] http://www.antivirusware.com/testing/performance/, posećeno 17.01.2016.

[116] L. Caviglione and A. Merlo. The energy impact of security mechanisms in modern mobile devices. Network Security,11(2), 2012.

[117] LI, Xun; CHONG, Frederic T. A Case for Energy-Aware Security Mechanisms. In: *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*. IEEE, 2013. p. 1541-1546.

[118] S. Chhabra and Y. Solihin. Transactions on computational science X. chapter Green secure processors: towards powerefficient secure processor design, pages 329–351. SpringerVerlag, Berlin, Heidelberg, 2010.

[119] N. R. Potlapally, S. Ravi, A. Raghunathan, and N. K. Jha. A study of the energy consumption characteristics of cryptographic algorithms and security protocols.IEEE Transactions on Mobile Computing, 5(2):128–143, 2006.

[120] S. J. Yang, J. Nieh, and N. Novik, “Measuring thin-client performance using slow-motion benchmark,” ACM Trans. Computer Systems, vol. 21, pp.87-115, no. 1,Febrary, 2003

[121] Agrawal, Shalabh; Biswas, Rana; Nath, Asoke, "Virtual Desktop Infrastructure in Higher Education Institution: Energy Efficiency as an Application of Green Computing," *Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference on* , vol., no., pp.601-605, 2014

[122] Youming Lin; Di Francesco, Mario, "Energy consumption of remote desktop access on mobile devices: An experimental study," *Cloud Networking (CLOUDNET), 2012 IEEE 1st International Conference on* , vol., no., pp.105,110, 2012

[123] Cimen, Caghan, Yusuf Kavurucu, and Halit Aydin. "Usage of thin-client/server architecture in computer aided education." *TOJET* 13.2,2014

[124] https://www.apc.com/resource/include/techspec_index.cfm?base_sku=BR1000G-JP, posećeno 18.01.2016.

[125] http://research.microsoft.com/en-us/projects/joulemeter/, posećeno 19.01.2016.

[126] http://ziyang.eecs.umich.edu/projects/powertutor/, posećeno 19.01.2016.

[127] Shine, Keith P., et al. "Alternatives to the global warming potential for comparing climate impacts of emissions of greenhouse gases." *Climatic Change* 68.3 (2005): 281-302.

[128] Santos, Ieda M. "Use of students’ personal mobile devices in the classroom: Overview of key challenges." *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*. No. 1. 2013.

[129] Graesser, Arthur C., Mark W. Conley, and Andrew Olney. "Intelligent tutoring systems." (2012).

[130] Pipatsarun Phobun, Jiracha Vicheanpanya, Adaptive intelligent tutoring systems for e-learning systems, Procedia Social and Behavioral Sciences 2 (2010), pp.4064–4069

[131] Liang Bing, "E-learning and modern education reform", Education Information, 2001, pp.21-25

[132] Feng Jian, “Cloud computing based distance education outlook”,China electronic education, 2009, Totally 273, pp.39-42

[133] Moodle. (2016). [Online], Available: https:// moodle.org/

[134] Xu Chuanling, Lu Hongjie, “E-learning”, Software World, 2001.08, pp. 139-141

[135] Vouk, M., Averitt, S., Bugaev, M., Kurth, A., Peeler, A., Shaffer, H., Sills, E., Stein, S., Thompson, J.: Powered by VCL - using virtual computing laboratory (VCL) technology to power cloud computing. In: 2nd Intl. Conference on the Virtual Computing Initiative(ICVCI), Research Triangle Park, North Carolina, USA (2008)

[136] Dong, B., Zheng, Q., Qiao, M., Shu, J., Yang, J.: BlueSky Cloud Framework: An E-Learning Framework Embracing Cloud Computing. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) Cloud Computing. LNCS, vol. 5931, pp. 577–582. Springer, Heidelberg (2009)

[137] Sulistio,A.,Reich,C.,Doelitzscher,F.:CloudInfrastructure&Applications–CloudIA. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) Cloud Computing. LNCS, vol. 5931, pp.583–588. Springer, Heidelberg (2009)

[138] Liang, P.-H., Yang, J.-M.: Virtual Personalized Learning Environment (VPLE) on the Cloud. In: Gong, Z., Luo, X., Chen, J., Lei, J., Wang, F.L. (eds.) WISM 2011, Part II.LNCS, vol. 6988, pp. 403–411. Springer, Heidelberg (2011)

[139] Blackboard. (2016). [Online], Available:http://www.blackboard.com

[140] Moodle. (2016). [Online], Available: https:// moodle.org/

[141] Claroline. (2016). [Online], Available:http://www.claroline.net/

[142] Canvas. (2016). [Online], Available:http://www.instructure.com/

[143] Hu,Z.,Zhang,S., ”Blended/hybrid course design in active learning cloud at south dakota state university“. In: 2nd ICETC, vol. 1, pp. V1-63–V1-67, 2010

[144] Vaquero, L.M., Educloud: Paas versus IaaS cloud usage for an advanced computer science course. IEEE Transactions on Education 54(4), 590–598, 2011

[145] Mustaffa, Mas Rina, and Evi Indriasari Mansor. "Facilitating Students’ Learning through M-LMS for Tablet Device." *International Journal of Computer Theory and Engineering* 7.3 (2015): 236.

[146] Hwa-Young Jeong & Bong-Hwa Hong, A practical use of learning system using user preference in ubiquitous computing environment, Multimed Tools Appl, 2013

[147] Velev, D. G. (2014). Challenges and Opportunities of Cloud-Based Mobile Learning. *International Journal of Information & Education Technology*, *4*(1

[148] Wang, M., Chen, Y., & Khan, M. J. (2014). Mobile cloud learning for higher education: A case study of Moodle in the cloud. *The International Review of Research in Open and Distance Learning*, *15*(2).

[149] S. Ahmed and D. Parsons, “Abductive science inquiry using mobile devices in the classroom,”Comput. Edu., vol. 63, pp. 62–72, 2013

[150] A. C. Jones, E. Scanlon, and G. Clough, “Mobile learning: Two case studies of supporting inquiry learning in informal and semiformal settings,”Comput. Edu., vol. 61, pp. 21–32, 2013

[151] Butoi, A., Tomai, N., &Mocean, L. (2013). Cloud-Based Mobile Learning. *Informática Economicâ*, *17*(2), 27-40.

[152] Ian F. Alexander, Neil Maiden, *Scenarios, Stories, Use Cases: Through the Systems Development Life Cycle*, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, England, 2004.

[153] Giemza, A., Verheyen, P., & Hoppe, H. U. (2012). Challenges in Scaling Mobile Learning Applications: The Example of Quizzer. In *WMUTE* (pp. 287-291)

[154] Chia, Y., Tsai, F. S., Tiong, A. W., & Kanagasabai, R. (2011, March). Context-aware mobile learning with a semantic service-oriented infrastructure. In *Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference on* (pp. 896-901). IEEE

[155] Marwan, M. E., A. R. Madar, and N. Fuad. "Development of Mobile EEF Learning System (MEEFLS) for Mobile Learning Implementation in Kolej Poly-Tech MARA (KPTM)." *Development* 1 (2014): 6183

[156] Boticki, I., Barisic, A., Martin, S., & Drljevic, N. (2012, March). Sortko: Learning Sorting Algorithms with Mobile Devices. In *Wireless, Mobile and Ubiquitous Technology in Education (WMUTE), 2012 IEEE Seventh International Conference on* (pp. 49-56). IEEE.

[157] de-Marcos, Luis, et al. "An experiment for improving students performance in secondary and tertiary education by means of m-learning auto-assessment." *Computers & Education* 55.3 (2010): 1069-1079.

[158] Blackboard Mobile Learn, Northern Illinois University, web: http://www.niu.edu/blackboard/mobile/, visited 18.08.2015.

[159] Malandrino, D., Manno, I., Palmieri, G., Scarano, V., Tateo, L., Casola, D., Ferrante I., Foresta, F. A Tailorable Infrastructure to enhance Mobile Seamless Learning. *IEEE Transactions on Learning Technologies,* vol. 8 , no. 1, 2015, pp. 18-30.

[160] Popovic, O., Markovic, S.D., Popovic, R., mTester – Mobile Learning System, *Computer Application in Engineering Education*, USA, 2016, DOI: 10.1002/cae.21719

[161] Flora, Harleen K., Xiaofeng Wang, and Swati V. Chande. "An Investigation on the Characteristics of Mobile Applications: A Survey Study." *International Journal of Information Technology and Computer Science (IJITCS)* 6.11 (2014): 21.