

Andrea Tikvica, univ. bacc. inf.¹

doc. dr. sc. Darko Andročec¹

PRIMJENA INTEROPERABILNOSTI RAČUNALNOG OBLAKA U POHRANI PODATAKA

Stručni rad / professional paper

UDK 004.78

Zbog mogućnosti financijskih ušteda, sve veći broj poslovnih organizacija razmatra korištenje ili već koristi računarstvo u oblacima. Međutim, postoje i problemi koji otežavaju primjenu ove nove paradigme. Jedan od najznačajnijih problema je zaključavanje korisnika od strane pružatelja usluge i nedostatak interoperabilnosti. Od modela računarstva u oblacima, za krajnje je korisnike najzanimljiviji softver kao servis, jer isti pruža gotove aplikacije. Ovaj rad daje osvrt na interoperabilnost softvera kao servisa na primjeru servisa u oblacima za pohranu datoteka. Napravljena je i aplikacija koja omogućava pristup i rad sa tri poznate usluge u oblaku za pohranu datoteka (Dropbox, Google Drive i Box).

Ključne riječi: interoperabilnost, softver kao servis, računarstvo u oblacima, pohrana podataka u oblacima.

1. Uvod

U današnje vrijeme se značajno povećava broj dinamičnih web usluga, te postaje sve teže predvidjeti tijek njihova korištenja. Jedna od novih tehnologija je i računarstvo u oblaku (engl. *cloud computing*). Kao usluga računarstvo u oblaku nudi: infrastrukturu računalnog oblaka (engl. *infrastructure as a service – IaaS*), platformu (engl. *platform as a service – PaaS*) i softver (engl. *software as a service – SaaS*) kao servise koji će u nastavku rada biti detaljnije objašnjeni.

U prvom poglavlju se nastavlja opisivanje računarstva u oblaku, te njegova važna uloga u suvremenim informacijskim sustavima. U drugom je poglavlju detaljnije opisan softver kao servis, jer je na toj usluzi focus našeg rada. S obzirom da je glavna tema rada interoperabilnost softvera kao servisa bitno je naglasiti važnost uspostavljanja interoperabilnosti i njegove moguće probleme. Rad se nastavlja opisom praktičnog dijela odnosno programa kojim se dokazala mogućnost povezivanja različitih rješenja softvera kao servisa i njihova interoperabilnost, konkretno sustava za pohranu podataka poput DropBox-a, Google Drive-a, i Box-a.

¹ Fakultet organizacije i informatike Sveučilišta u Zagrebu, Varaždin

2. Računarstvo u oblaku

Postoji više definicija računarstva u oblaku (engl. *cloud computing*), a ovdje ćemo izdvojiti one koji se tiču programskog okruženja. „Oblak“ je vrlo česta riječ korištena kao metafora za Internet. Nastao je zbog želje za inovacijama i povećanjem kapaciteta bez dodatnih ulaganja, osposobljavanja novog osoblja, kupnje licenciranih programa, i sl. Računalni resursi se iznajmljuju prema potrebi korisnika, a oni im pristupaju bez obzira gdje su korišteni resursi smješteni. Osim toga korisnicima je usluga oblaka dostupna onoliko koliko plaćaju i koriste računalna sredstva (Stupar, 2012).

Programska rješenja u oblaku promoviraju dostupnost i sastoje se od pet ključnih karakteristika koje pokazuju odnos i razlike računarstva u oblaku i tradicionalnog pristupa informatici:

- pružanje usluga na zahtjev korisnika,
- brz mrežni pristup,
- udruživanje resursa,
- brza elastičnost,
- odmjerena usluga; od tri modela pružanja usluga koji se često zovu SPI modeli (program, platforma i infrastruktura):
- softver kao usluga (engl. *Software as a Service – SaaS*),
- platforma kao usluga (engl. *Platform as a Service – PaaS*),
- infrastruktura kao usluga (engl. *Infrastructure as a Service – IaaS*); te od četiri modela implementacije:
- javni oblak,
- privatni oblak,
- hibridni oblak (Laboratorij za sustave i signale, 2013).

Nacionalni institut za standarde i tehnologiju (engl. *The National Institute of Standards and Technology's*) definiciju računarstva u oblaku određuje kroz pet karakteristika: usluga na zahtjev (engl. *on-demand self-service*), brz mrežni pristup (engl. *broad network access*), grupiranje resursa (engl. *resource pooling*), brza elastičnost (engl. *rapid elasticity*) i odmjerena usluga (engl. *Measured service*) – Mell i Grance (2011).

2.1. Razine usluga dostupne putem računarstva u oblaku

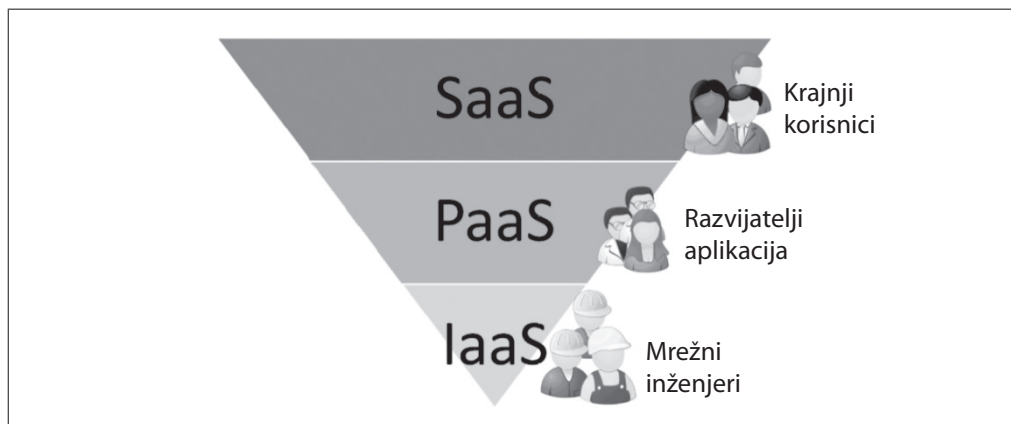
Računarstvo u oblaku podijeljeno je na tri arhitekturna modela koja ujedno predstavljaju SPI model (engl. *Software, Platform, Infrastructure*). Mogući modeli pružanja usluge su:

1. softver kao servis (engl. *software as a service – SaaS*),
2. platforma kao servis (engl. *platform as a service – PaaS*),
3. infrastruktura kao servis (engl. *infrastructure as a service – IaaS*).

Sva tri modela (slika 1) imaju slična značenja, ali različite primjene. SaaS omogućuje spremanje podataka, njihovu sigurnosnu kopiju i dijeljenje unutar oblaka. PaaS služi za kreiranje SaaS proizvoda i brine o obradi podataka, dok IaaS-om možemo testirati kreirane aplikacije,

te je njime korisniku omogućena implementacija i pokretanje proizvoljnog programa (Nacionalni CERT, 2010).

Slika 1: Korisnici i razine usluga dostupne putem računarstva u oblaku



Izvor: Stupar (2012)

2.1.1. Softver kao servis (SaaS)

Softver kao servis predstavlja tehnološku platformu koja omogućava uporabu dostupnih aplikacija putem interneta koji se nalaze u infrastrukturi oblaka. Uz pomoć web preglednika korisnik ima mogućnost korištenja aplikacija, no nema pristup pozadinskoj infrastrukturi (osim korisničkim konfiguracijama). Korisnik plaća korištenje usluga pružatelja softvera kao servisa (Grives, 2011).

SaaS je danas postao model zajedničke isporuke potrebnih podataka za mnoge poslovne programe kao što su programi za obradu podataka, za obračun plaće, DBMS sustava (engl. *Database Management System*) odnosno sustavi za upravljanje bazom podataka, računovodstva i mnogih drugih. Ova skupina pružanja usluga se sastoji od dvije vrste:

- vertikalni SaaS – programi koji odgovaraju djelatnostima određene industrije kao što su zdravstvo, nekretnine, financije, poljoprivreda, itd.
- horizontalni SaaS – programi koji se odnose na razvoj aplikacija (ili softvera općenito) kao što su prodaja, alati za razvojne programere, marketing, itd.

SaaS-ova bitna značajka je da njegove kopije usluga koristi više korisnika (kao npr. elektronička pošta), te ga tako razlikujemo od ostalih aplikacija tradicionalnih tehnologija. Sljedeća značajka se odnosi na centralizirani odnosno mrežni pristup s manje opterećenja kojeg SaaS nudi svojim korisnicima. Kao i svaka usluga SaaS ima svoje prednosti i mane, a kako bi poduzeće osigurala prednosti SaaS-a moraju predati kontrolu nad svojim podacima i imati povjerenje u pružatelja usluga da su ti podaci sigurni. Kako bi se steklo povjerenje, potrebno je stvoriti arhitekturu podataka SaaS-a koja je sigurna i s kojom bi zadovoljilo klijente koji su zabrinuti za svoje predane podatke. Osim toga, takva arhitektura se treba pokazati učinkovito za rad i biti ekonomična za administraciju i poslovanje te općenito održavanje (Micro-

soft, 2014). Kao bitna prednost SaaS aplikacija je njihovo često ažuriranje. Primjeri SaaS programa su: Google Apps, Microsoft Office 365, Dropbox, Salesforce.com, itd. Neke od mogućnosti spomenutih primjera SaaS aplikacija prikazane su u *Tablici 1*.

Tablica 1: Glavne mogućnosti primjera SaaS aplikacija

| Naziv | Mogućnost | Kratki opis |
|-----------------------------|---|--|
| Google Apps | Google Dokumenti | Pružaju mogućnost kreiranja dokumenata i njihovu obradu. Olakšava dijeljenje dokumenata i kolaboraciju u stvarnom vremenu. |
| | Google Kalendar | Nudi zakazivanje sastanaka, zapisivanje bilješki itd. Mogućnost objavljivanja javnog kalendara. |
| | Google Talk | Integrirana s Gmail-om, pokretanje <i>chat</i> sesije iz e-pošte. Također ima mogućnost prijenosa datoteka, glasovne komunikacije itd. |
| | Gmail | Nudi 25 GB prostora za e-pošte. |
| | Kreator stranica | Stvaranje web stranica koje su kompatibilne s Microsoft Internet Explorer-om i Mozilla Firefox-om. |
| Microsoft Office 365 | Kreiranje datoteka u stvarnom vremenu (Word, Excel, PowerPoint) | Kreiranje datoteka u stvarnom vremenu i njihovo spremanje na servisu kako bi drugi mogli vidjeti i mijenjati datoteku. |
| | Razgovor sa suradnicima | Integracija Skype aplikacija u Office sustavu. Mogućnost slanja poruka, dijeljenje zaslona i video/audio razgovora. |
| | Linkovi za pohranjene datoteke | Umjesto slanja privitaka, čime se zauzima memorija, postoji mogućnost umetanja veze na datoteku u oblaku. |
| | Miš kao laserski pokazivač za PowerPoint prezentacije | Pomoću tipkovničkog prečaca (Ctrl + P) uključuje se miš kao lažni laserski pokazivač crvene točke. |
| | Nastavak čitanja Wordovog dokumenta | Nakon zatvaranja Wordovog dokumenta, sustav pamti gdje ste zadnji put stali s čitanjem. |
| Dropbox | Vraćanje datoteka i starijih verzija | Jednostavno vraćanje izbrisanih datoteka i vraćanje prethodnih verzija datoteke. |
| | Dopuštenja za povezivanje | Linkovi sa zaštitom lozinke ili postavljanje isteka vremena za odobreni pristup. |
| | Pametno sinkroniziranje | Pristup svakoj datoteci izravno s njihove radne površine. |
| | Alat za prijenos računa | Prijenos računa s jednog korisnika na drugog kada se odgovornosti promijene. |
| | Nadzorna ploča administratora | Praćenje aktivnosti tima, pregled povezanih uređaja itd. |

| Naziv | Mogućnost | Kratki opis |
|-----------------------|--------------------------------------|--|
| Salesforce.com | Upravljanje računom i kontaktima | Kompletan prikaz kontakata, povijest aktivnosti, ključni kontakti, komunikacija s klijentima i rasprave. |
| | Upravljanje prilikama | Kompletan pregled poslovanja tima. Povezanost s ljudima i dobivanje informacija kojima se zatvaraju prodaje. |
| | Podaci o prodaji | Primanje pravih podataka u pravo vrijeme. |
| | Tijek rada i odobrenja | Brzo dizajniranje i automatiziranje bilo kojih poslovnih procesa jednostavnim povuci-i-ispusti (engl. <i>Drag-and-drop</i>) |
| | Sinkronizacija i dijeljenje datoteka | Brzo pronalaženje datoteka i njihovo dijeljenje. Mogućnost slanja upozorenja ako se nešto promijeni. |

Izvor: Babeni (2016), Dropbox (2017), Salesforce.com (2017)

2.1.2. Platforma kao servis (PaaS)

Platforma kao servis korisniku pruža mogućnost implementacije na infrastrukturu davatelja usluge koji su stvoreni pomoću programskih jezika, usluga i alata koje podržava davatelj usluga. Tu infrastrukturu, koja se nalazi u pozadini programa, korisnik ne može kontrolirati (operacijski sustav, mreža, itd.), ali ima kontrolu nad implementiranim programima i opcionalno konfiguracijama sustava. Aplikacije se korisniku poslužuju putem poslužiteljskog sučelja koji se mogu dohvatiti uz pomoć Interneta.

Točniji naziv za ovaj model bila bi arhitektura, zbog toga što plaćamo i koristimo tuđe rješenje koje se brine za prikupljanje, obradu, pohranjivanje i traženje podataka. Primjer PaaS programa su: Google App Engine, Force.com i Windows Azure. Kratki prikaz nekih primjera dan je u *tablici 2*.

Tablica 2: Glavne mogućnosti primjera PaaS platformi

| Naziv | Mogućnost | Kratki opis |
|--------------------------|---------------------------------------|--|
| Google App Engine | Identitet aplikacije | Okvir koji omogućuje pristup identitetu aplikacije uz pomoć OAuth. |
| | Sigurnosna kopija i vraćanje datoteka | Omogućuje izvoz/uvoz datoteka na/iz aplikacija pomoću „Google Cloud Platform Console“. |
| | Google Cloud Endpoints | Generira API-je za iOS, Android i web servise za lakši razvoj aplikacija. |
| | OAuth protokol | Autorizacija korisnika bez unosa korisničkog imena i lozinke. |
| | Radna točka REST API | Omogućuje upotrebu niza funkcionalnosti aplikacije putem REST-a. |

| Naziv | Mogućnost | Kratki opis |
|----------------------|---|---|
| Force.com | Platforma za sigurnost i model dijeljenja | Kontroliranje korisničkog pristupa različitim podacima. |
| | Pravila tijeka rada | Automatsko dodjeljivanje zadataka, ažuriranje podataka i slanje upozorenja putem e-pošte zbog poslovnih događaja. |
| | Postupci odobravanja | Postavljanje niza koraka potrebnih za odobravanje zapisa, uključujući one koji ih moraju odobravati na svakom koraku. |
| Windows Azure | Sigurnosna kopija i arhiviranje | Zaštita podataka i aplikacija bez obzira gdje se nalazili, te arhiviranje podataka u oblaku |
| | Skladište podataka | Rukovanje eksponencijalnim rastom podataka bez ostavljanja sigurnosti, skalabilnosti ili analitike iza sebe |
| | Migracija glavnog računala | Postignuće pouzdanosti, agilnosti i skalabilnosti |

Izvor: Google App Engine (2017), Force.com (2016), Microsoft Azure (2017)

2.1.3. Infrastruktura kao servis (IaaS)

Kao usluga pretplatnicima infrastruktura kao servis je najosnovniji model jer nudi računalnu infrastrukturu (virtualne uređaje i druge resurse), odnosno predstavlja temelj na kojem upravljamo cijelim sustavom. Ono pruža aplikacijsko programsko sučelje (API) visoke razine kako bi razlikovao različite detalje niske razine mrežne infrastrukture (lokacija, sigurnost, sigurnosna kopija, skaliranje, itd.). Korisnik nema pristup infrastrukturi, ali ima nadzor nad operacijskim sustavom i razvojem aplikacija. Neke odabrane komponente umrežavanja mogu biti ograničene za korisnika. Primjeri IaaS programa su: SQL Server, AT&T, Rackspace Cloud i Amazon Web Services (Nacionalni CERT, 2010).

Tablica 3: Glavne mogućnosti primjera IaaS infrastruktura

| Naziv | Mogućnost | Kratki opis |
|-------------------|--|--|
| SQL Server | Pohrana podataka | Baza podataka čija je zbirka tablica upisana u stupcima i podržava različite tipove podataka. |
| | Upravljanje međuspremnikom | SQL Server koristi radnu memoriju (RAM) kako bi se smanjio input/output diska. |
| | Istovremeno korištenje i zaključavanje | Mogućnost istovremenog korištenja baze podataka od strane više klijenata, a istovremeni pristup se može kontrolirati zaključavanjem. |
| AT&T | Ispitivanje opterećenja | Provjera da li je izvedba aplikacije u skladu s određenim ciljevima |
| | Nadgledanje web stranica i aplikacija | Primanje upozorenja ako se korisničko iskustvo ometa, te primanje izvješća o izvedbama |
| | Održavanje | Nema napora oko čuvanja aplikacije, ona uvijek ostaje pravilno podesiva |

| Naziv | Mogućnost | Kratki opis |
|------------------|--------------------------|---|
| Rackspace | Rackspace javni oblak | Nudi set alata i tehnologija koji pomažu poslovanju i održavanju rasta. |
| | Rackspace privatni oblak | Jedinstveno okruženje s fizički izoliranom mrežom. |
| | Hibridni oblak | Okruženje kombinirano s više oblaka. |

Izvor: Autori

3. Interoperabilnost

Interoperabilnost možemo opisati kao mjeru do koje različiti sustavi mogu međusobno komunicirati i uspješno surađivati. Međunarodne organizacije za standardizaciju IEEE i ISO definiraju interoperabilnost kao sposobnost dva ili više sustava ili aplikacija da razmjenjuju informacije i koriste ih.

U kontekstu računarstva u oblaku interoperabilnost bi se trebala smatrati kao sposobnost javnih, privatnih i ostalih vrsta oblaka unutar poduzeća da razumiju aplikacijsko okruženje, njegovo sučelje, konfiguraciju, itd. U tijeku razmjene informacija za računarstvo u oblaku su bitne one komponente koje se tiču oblaka. Misli se na korisnike koji koriste usluge oblaka, odnosno koriste usluge pružatelja u oblaku. Priroda takve interakcije postiže se mrežnom vezom pomoću propisanog sučelja ili API-ja. Oni su najčešće podijeljeni na više sučelja gdje svako sučelje ima svoju ulogu u oblaku kao na primjer: sučelje servisa za oblak, sučelje za upravljanje uslugama u oblaku, sučelje autorizacije, itd. (Cloud Standards Customer Council, 2012).

Ako razmatramo interoperabilnost između dva sustava korisno je imati model razmjene podataka s obzirom na to da se pokazalo da ima više mogućih aspekata razmjene koji se tiču povezivanja sustava. Postoji više modela interoperabilnosti, a izdvojiti ćemo model koji je usko povezan s korisničkim korištenjem usluga oblaka. Model je podijeljen na četiri razine i opisan je u radu provedenom za Europsku Uniju, a njegove razine su prikazane u *Tablici 4*.

Tehnička interoperabilnost se bavi uspostavljenim standardom kao što su TCP/IP i REST HTTP protokol odnosno protokolima za razmjenu informacija. Prilikom prijenosa podataka bitno je obratiti pažnju na njihov format koji na primjer može biti XML strukture podataka ili

Tablica 4: Razine interoperabilnosti

| Razina | Svrha | Objekt | Rješenje |
|-----------------------|--|----------------------|---|
| Tehnička | Siguran prijenos podataka | Signali | Protokol prijenosa podataka |
| Sintaktička | Obrada primljenih podataka | Podatak | Standardizirani formati prijenosa podataka |
| Semantička | Obrada i tumačenje primljenih podataka | Informacija | Zajednički direktoriji, podatkovni ključevi |
| Organizacijska | Automatsko povezivanje procesa između različitih sustava | Procesi (tijek rada) | Standardizirani procesni elementi |

Izvor: Autori

JSON podatkovni tok, a o tome se brine sintaktička interoperabilnost. Kod semantičke interoperabilnosti osim što obrađuje primljene podatke, ono im dodjeljuje značenje i strukturu. Organizacijska interoperabilnost se odnosi na odgovarajuće standarde prema kojima sustav slanja očekuje da će sustav primatelja razmijeniti podatke.

Idealan scenarij bio bi ako dva sustava uspješno uspostavljaju komunikaciju na temelju sve četiri razine. Na primjer, sustavi se mogu razlikovati na razini sintaktičke interoperabilnosti tako što jedan sustav koristi XML, a drugi sustav JSON datoteke (Kubicek i Cimander, 2009). Glavni problemi interoperabilnosti platforme kao servisa opisani su u radu (Andročec, Vrček, 2016). Tamo su problemi interoperabilnosti klasificirani korištenjem ontologije u sljedeće kategorije: problemi interoperabilnosti aplikacijskih programskih sučelja, problemi interoperabilnosti na aplikacijskoj razini, pravni problem interoperabilnosti, organizacijski problem interoperabilnosti i problemi interoperabilnosti spremišta podataka u oblacima.

4. SaaS rješenja i njihova API okruženja

U aplikaciji koju smo izradili, implementirana su tri SaaS rješenja pomoću njihovih API okruženja. Ideja programa je da se dokaže mogućnost interoperabilnosti više softvera kao servisa. Izrada ovog programa rezultiralo je jednostavnijim pristupom SaaS uslugama. Program je izrađen u Java programskom jeziku, korišteni alat je Eclipse, a integrirana su sljedeća tri SaaS rješenja: Dropbox, Google Drive i Box.

4.1. Dropbox

Dropbox je osobni servis za pohranu podataka unutar oblaka koji omogućuje prijenos i dijeljenje datoteka. Svim spremljenim datotekama (mapama, slikama, glazbi, itd.) imate pristup s bilo kojeg uređaja (računalo, mobitel, tablet i sl.) koji ima mogućnost spajanja na Internet. Dropbox je dostupan na operacijskim sustavima Windows, Linux i Macintosh. Mobilne aplikacije moguće je preuzeti na uređaje za Android, iPhone, iPad i BlackBerry.

Nakon otvaranja korisničkog računa, nudi se 2 GB besplatnog slobodnog prostora odnosno korisnik na samom početku koristi obični paket (*engl. Basic*). Ostala dva paketa koji se nadoplaćuju su: *Pro* koji nudi 1 TB prostora za pohranu i poslovni (*engl. Bussines*) koji nudi beskonačno slobodnog prostora, jer se dijeli na više osoba. Svi korisnički podaci zapisani su u Amazonovom jednostavnom servisu za pohranu (*engl. Amazon Simple Storage Service*) odnosno Amazon S3, a zaštićeni su 256-bitnim šifriranjem pomoću sigurnog sloja soketa (*engl. Secure Sockets Layer – SSL*) i naprednog sustava šifriranja (*engl. Advanced Encryption System – AES*).

Dropbox programerima nudi bogatu dokumentaciju za razvijanje aplikacija pomoću istoimenog programa. U ponudi nude službene SDK-ove za nekoliko programskih jezika i to: Java, JavaScript, Python, .NET, Swift, Objective-C i HTTP. Za potrebe izrade programa korišten je službeni Java SDK za API v2 koji pomaže integrirati Dropbox u svoje aplikacije pomoću Java programskog jezika (Cobbe, 2013).

Tablica 5: Popis korištenih funkcija API Dropbox-a

| Funkcija | Kratki opis |
|---|---|
| authDropbox(String dropBoxAccessToken) | Autorizacija korisnika putem unesenog korisničkog tokena. |
| uploadToDropbox(String fileName) | Služi za prijenos datoteke na Dropbox, ulazni parametar je naziv datoteke. |
| listDropboxFolders(String folderPath) | Ispis svih datoteka u Dropbox-u, unosi se putanja direktorija. |
| downloadFromDropbox(String fileName) | Skidanje željene datoteke s Dropbox-a na računalo, ulazni parametar je naziv datoteke. |
| deleteFile(String fileName) | Brisanje odabrane datoteke. |
| getDropboxSize() | Vraćanje informacije o zauzetosti prostora na Dropbox-u i podatak o veličini raspoloživog prostora. |

Izvor: Autori

4.2. Google Drive

Kao Dropbox-ov konkurent 2012. godine pojavljuje se Google Drive koji nudi usluge pohranjivanja i sinkronizaciju datoteka koju je razvio Google. Svojim korisnicima nudi mogućnost pohrane datoteka unutar oblaka, njihovo dijeljenje i sinkronizaciju na različitim uređajima. Google Drive svim svojim korisnicima na početku nudi 15 GB slobodnog prostora za pohranu. Uz nadoplatu korisnik može odabrati paket od 100 GB prostora, odnosno paket od 1 TB, 2 TB, itd. Google Drive ima mogućnost izvanmrežnog pristupa tako da u slučaju gubitka signala korisnik uvijek ima pristup svojim datotekama spremljenim na disku.

Google Drive platforma nudi mogućnost razvoja aplikacija koji se integriraju s Drive-om uz pomoć grupe API-ja s bibliotekama klijenata. Osim pohrane i preuzimanja datoteka, što predstavlja osnovnu funkciju Google Drive-a, njegova platforma pruža resurse za izgradnju klijentskih aplikacija. Njegove resurse mogu se implementirati pomoću SDK-ova sljedećih programskih jezika: .NET, Java, JavaScript, PHP, Ruby i Python, a nudi rješenja i za Android i iOS (GCFLearnFree, 2017).

Tablica 6: Popis korištenih funkcija API Google Drive-a

| Funkcija | Kratki opis |
|--|--|
| authorize(String clientId, String cSecret) | Služi za autorizaciju korisnika uz pomoć unesenih podataka klijent id i tajna klijenta |
| upload(String clientId, String clientSecret, String filePath) | Funkcija koja pokreće autorizaciju i spremanje datoteke |
| uploadFile(boolean useDirectUpload) | Sprema odabranu datoteku koju nalazi preko unesene putanje |

Izvor: Autori

4.3. Box

Box je još jedan u nizu servis za upravljanje spremljenim sadržajem odnosno njihovo pohranjivanje i dijeljenje na računalnim, mobilnim i sličnim uređajima. Kao dio oblaka korisnicima Box-a nudi usluge dijeljenja sadržaja s drugim korisnicima, odnosno mogu birati način dijeljenja. Korisnici mogu pozvati druge da pregledaju ili preuzmu podijeljen sadržaj, prenesu datoteke i fotografije u zajedničkoj mapi. Prilikom kreiranja korisničkog računa Box korisniku nudi 10 GB besplatnog slobodnog prostora za pohranu datoteka u oblaku. Korisnik može uz nadoplatu kupiti i osobni paket od 100 GB slobodnog prostora, dok s poslovnim paketima dobiva neograničeno pohranjivanje. Box uglavnom koriste tvrtke koje se bave određenim industrijama, uključujući zdravstvo i maloprodaju. Godine 2016. Box je stekao MedXT tako da klijentima medicine može bilježiti medicinske slike.

Za razvoj aplikacija Box nudi mnogo SDK-ove za popularne web i mobilne jezike kako bi se ubrzao razvojni proces. Box u svojoj ponudi ima sljedeće web SDK-ove: Java, .NET, Node.js, Python, Ruby, Chrome i Salesforce, te za mobilne operacijske sustave Android, Windows i iOS (Box, 2017).

Tablica 7: Popis korištenih funkcija API Box-a

| Funkcija | Kratki opis |
|--|---|
| BoxAuthorize(String token) | Autorizacija korisnika Box-a pomoću unešenog Developer tokena. |
| uploadFile(String fileName, String token) | Spremanje odabrane datoteke na Box. |
| downloadFile(Object object) | Preuzimanje odabrane datoteke s Box-a. |
| listFolder(BoxFolder folder) | Ispis svih datoteka koji se nalaze na Box-u. |
| usedSpace(BoxFolder folder) | Funkcija koja vraća informaciju o zauzeću prostora i o raspoloživosti prostora. |
| deleteFile(Object object) | Brisanje odabrane datoteke. |

Izvor: Autori

5. Opis funkcionalnosti razvijene aplikacije

Sav izvorni kod je napisan u klasama, a općenito je osnovni koncept u Javi klasa (*engl. class*). Svaka klasa u programu je deklarirana unutar datoteke i sufiksom .java. U *Tablici 8.* prikazane su sve korištene klase.

GlavnaForma.java – Ova klasa stvorila se automatski prilikom kreiranja aplikacijskog prozora u Swing proširenju. U dizajnerskom pogledu ima ponuđen izbornik svih komponenata koje je moguće unijeti na formu. Prilikom unosa svake komponente automatski se unosi i mali dio koda u pripadajućoj klasi kojom se osigurava postojanje odabrane komponente. Za tri gumba bilo je potrebno kreirati akcije (klik na gumb) koje su isto bile popraćene automatskim kreiranjem odgovarajućeg koda. Unutar novokreirane funkcije odnosno *ActionListener()*-a izvršavaju se određene akcije, kao na primjer otvaranje nove forme. Isto tako prilikom klika na radio gumbove prolazi provjera da je samo jedan radio gumb odabran.

Tablica 8: Popis klasa u programu

| Klase |
|--|
| GlavnaForma.java |
| FormaUnosPodataka.java |
| PodaciKorisnika.java |
| Dropbox.java |
| DriveSample.java |
| FileUploadProgressListener.java |
| App.java |

Izvor: Autori

FormaUnosPodataka.java – Kreirana je istim principom kao i *GlavnaForma.java*, no ovdje se radi o spremanju podataka u listu koja se nalazi u klasi *PodaciKorisnika.java*. Unesene podatke moguće je naknadno mijenjati.

PodaciKorisnika.java – Unos korisničkih podataka sprema se u prvi i jedini zapis u listi u klasi *PodaciKorisnika.java*. Lista se sastoji od sljedećih string-ova: *AccessToken*, *ClientId*, *ClientSecret* i *DeveloperToken*. Ako se korisnik odluči mijenjati podatke, program prepoznaje da se radi o mijenjaju ako postoji već upisan podatak u listi, briše ga i ponovno upisuje novo-uneseni podatak.

Dropbox.java – U klasi *Dropbox.java* napisan je kod koji je preuzet sa službene stranice *Dropbox*-a u rubrici za API spajanje, s manjim izmjenama. Prilikom pokretanja koda unutar klase prvo se uspostavlja autorizacija na temelju unesenog tokena. Nakon uspješnog spajanja prenosi se odabrana datoteka na spremište korisnika, te javlja poruku o uspješnosti prijena.

DriveSample.java – Klasa *DriveSample.java* služi sa spajanje na *Google Drive* račun uz pomoć unesenih klijentskih ključeva. Sav kod preuzet je sa službene *Google*-ove stranice za spajanje na *Google Drive* preko API-ja. Prvo se uspostavlja autorizacija, zatim kreće spremanje datoteke koja rezultira povratnom informacijom o uspješnosti prijena. Ova i sljedeća klasa su povezane.

FileUploadProgressListener.java – Ova klasa služi kao slušatelj progressa spremanja datoteke na *Google Drive*. U konzoli ispisuje poruku o procesu izvršavanja koje korisniku nisu vidljive.

App.java – Klasa *App.java* je pripadajuća klasa preuzetog projekta sa službene stranice *Box*-a. Predstavlja glavnu klasu koja izvršava spajanje na *Box*-ov korisnički račun i spremanje odabrane datoteke. Potrebno je osigurati odgovarajući token i autorizacija računa će se uspješno izvršiti.

Prilikom pokretanja programa otvara nam se glavna forma koja se sastoji od 2 dijela. Prvi dio se odnosi na spremanje datoteke i ono se sastoji od jednog *Textbox*-a za prikaz putanje odabrane datoteke, tri radio gumba (za odabir željenog servisa odnosno mjesto spremanja datoteke) i tri gumba: „Priloži“ (mogućnost odabira željene datoteke za prijenos), „Spremi datoteku“ (pokretanje spremanje datoteke na odabrani servis) i „Vaši podaci“ (otvara novu formu za unos korisničkih podataka).

Drugi dio služi za ispis svih datoteka s *Dropbox*-a i *Box*-a, preuzimanje pojedinih datoteka na računalo ili spremanje izravno na *Google Drive*, te brisanje odabrane datoteke. Sastoji se

Slika 11: Glavna forma programa

Izvor: Autori

od jednog listboxa (prikaz svih datoteka sa određenog servisa), jedne padajuće liste, jedne labele i četiri gumba: „Izlistaj“ (ispis svih datoteka), „Računalo“ (spremanje datoteke na računalo), „Google Drive“ (spremanje datoteke na Google Drive), i „Izbriši datoteku“ (brisanje odabrane datoteke). Svi dijelovi glavne forme prikazani su na *Slici 11*.

Prije samog spremanja datoteke nužno je unijeti korisničke podatke inače program ne može uspješno raditi. Klikom na gumb „Vaši podaci“ otvara se nova forma koja ima četiri Textbox-a i to: jedan za Dropbox (unos pristupnog tokena), dva za Google Drive (unos klijentskog ID-a i tajne klijenta) i jedan za Box (unos tokena za razvojne programere). Nije nužno unijeti svaki podatak već one za koje korisnik ima otvoren korisnički račun. Nakon uspješnog unosa korisnik klikom na gumb „Spremi“ zapisuje podatke u listu programa i zatvara formu klikom na gumb „Zatvori“. Korisnik se u svakom trenutku može vratiti na formu i mijenjati unesene podatke. Prilikom pritiska na gumb „Spremi“ korisnik dobiva povratnu informaciju o uspješnosti spremanja podataka. Pritiskom na gumb „Priloži“ odabiremo željenu datoteku za prijenos. Nakon odabira datoteke prikazuje se putanja datoteke na lokalnom disku korisnika. Isto tako, ispod prikazane putanje korisniku je vidljiv naziv odabrane datoteke i njena ekstenzija.

Nakon priložene datoteke i odabira mjesta spremanja potrebno je kliknuti na gumb „Spremi datoteku“ kako bi se pokrenulo spremanje. Ako korisnik nije unio korisničke podatke odmah mu se javlja poruka o grešci, inače javlja poruku o uspješnosti spremanja. U povratnoj informaciji korisnik vidi svoje korisničko ime spojenog servisa, naziv prenesene datoteke i njenu veličinu u KB.

Ako korisnik želi vidjeti sadržaj svog spremnika na Dropbox-u ili Box-u jednostavnim klikom na gumb „Izlistaj“, ovisno o odabranom servisu iz padajuće liste, ispisuje se lista svih njegovih datoteka. Korisnik sada može odabrati željenu datoteku i spremi je svoje računalo klikom na gumb „Računalo“. Druga opcija je spremi na Google Drive čime se onda odabrana datoteka sprema na računalo i onda izravno sprema na Google Drive s preduvjetom da su uneseni potrebni korisnički podaci. Poviše padajućeg izbornika prikazuje se informacija o zauzetosti prostora i raspoloživog prostora odabranog servisa. Korisnik odabranu datoteku isto tako može i izbrisati.

6. Zaključak

Sva tri servisa, odnosno njihova API okruženja, pokazala su se vrlo korisnim za korištenje i razvoj aplikacije. Implementacija kodova bila je puno jednostavnija za Dropbox i Box, dok je za Google Drive trebalo potrošiti više vremena. Isto tako, sva tri servisa se razlikuju po svojim pristupnim kodovima. S obzirom na to da API dozvoljava pristup korisničkim računima preko tokena i ključeva, kako bi se izbjeglo upisivanje korisničkih imena i lozinki, tako su navedeni servisi tu mogućnost koristili na različite načine. Google Drive ima dva fiksna podatka i oni se ne mijenjaju osim tajne klijenta koji se može promijeniti na zahtjev korisnika. Dropbox ima sličnu mogućnost kao i Google Drive, no potrebno je prilikom svakog spajanja unositi aktivacijski ključ što usporava postupak spremanja datoteke. Stoga se koristi pristupni token kojeg je potrebno generirati nakon svakog zatvaranja Dropbox-a. Na samom kraju nalazi se Box koji funkcionira kao i Dropbox, no njegov token traje 60 minuta, te ga je onda potrebno obnoviti odnosno ponovno zatražiti.

Kada se svi spoje u jednu aplikaciju odnosno kada se postigne interoperabilnost servisa ono rezultira funkcionalnim programom s mnogo mogućnosti. Problem međusobnog spajanja može nastati ako korisnik unese krive ili nepostojeće korisničke podatke pri čemu onda povezivanje nije omogućeno.

SaaS rješenja su se pokazala veoma korisnim kao zasebne aplikacije. Uglavnom su usvojene od strane malih i srednjih poduzeća. Postizanjem njihove međusobne interoperabilnosti dolazi do produktivnijeg pristupa uslugama. Svaka tvrtka trebala bi imati strategiju upravljanja podacima i shodno tome osigurati visoku sigurnost podataka. Prema tome treba pripaziti na moguće probleme prilikom interoperabilnosti kako se ne bi narušila sigurnost, a s tim povećao nadzor nad podacima. Svaki od SaaS rješenja nastoji izbjegavati takve probleme, no programeri koji žele koristiti njihove usluge moraju imati na umu moguće rizike.

LITERATURA

1. Andročec, D. i Vrčec, N. (2016.) *Ontologies for Platform as Service APIs Interoperability, Cybernetics and Information Technologies*, God. 16., Br. 4, str. 29-44.
2. Babeni, S. (2016.) 15 Cool Features You Should Be Using In Office 365, <http://www.sherweb.com/blog/15-cool-features-you-should-be-using-in-office-365/> (pristup: 5.9.2017.)
3. Box (2017.), About Box, <https://www.box.com/about-us> (pristup: 29.8.2017.)
4. Cloud Standards Customer Council (2014.) *Interoperability and Portability for Cloud Computing: A Guide*, <http://www.cloud-council.org/deliverables/CSCC-Interoperability-and-Portability-for-Cloud-Computing-A-Guide.pdf> (pristup: 20.8.2017.)
5. Cobbe S. (2013.) *Dropbox Core SDK for Java 6+*, <https://github.com/dropbox/dropbox-sdk-java> (pristup: 29.8.2017.)
6. Dropbox (2017.) *Dropbox Features*, <https://www.dropbox.com/business/tour> (pristup: 5.9.2017.)
7. Force.com (2016.) *The Benefits of a Force.com App*, https://resources.docs.salesforce.com/sfdc/pdf/salesforce_creating_on_demand_apps.pdf (pristup: 5.9.2017.)
8. GCFLearnFree (2017.) *Google Drive: All about Google Drive*, <https://www.gcflearnfree.org/googledriveanddocs/all-about-google-drive/1/> (pristup: 29.8.2017.)
9. Google App Engine (2017.) *The App Engine Standard Environment*, <https://cloud.google.com/appengine/docs/standard/> (pristup: 5.9.2017.)
10. Grieves, J. (2011.) *The Difference Between Software, Platform, and Infrastructure in SaaS, PaaS, and IaaS*, <http://bigswingdev.wpengine.com/2011/01/the-difference-between-software-platform-and-infrastructure-in-saas-paas-and-iaas.html> (pristup: 21.8.2017.)
11. Kubicek, H. i Cimander R. (2009.) *Three dimensions of organizational interoperability*, http://www.ifib.de/publikationsdateien/Kubicek_Cimander_ePractice_Journal_vol_6.pdf (pristup: 27.8.2017.)
12. Laboratorij za sustave i signale (2013.) *Cloud computing*, <http://security.lss.hr/arhiva-dokumenata/cloud-computing.html> (pristup: 20.8.2017.)
13. Mell, P. i Grance, T. (2011.) *The NIST Definition of Cloud Computing (tehnički izvještaj)*, National Institute of Standards and Technology: U.S. Department of Commerce
14. Microsoft Azure (2017.) *Azure solutions*, <https://azure.microsoft.com/en-us/solutions/> (pristup: 5.9.2017.)
15. Nacionalni CERT (2010.) *Cloud computing*, <http://www.cert.hr/sites/default/files/NCERT-PUBDOC-2010-03-293.pdf> (pristup: 20.8.2017.)
16. Salesforce.com (2017.) *Features*, <https://www.salesforce.com/products/sales-cloud/features/> (pristup: 5.9.2017.)
17. Stupar I. (2012.) *Ostvarivanje kvalitete usluge u računarstvu u oblaku putem skalabilnosti i sporazuma o razini usluge*, https://www.fer.unizg.hr/_download/repository/lvana_Stupar_KDI.pdf (pristup: 17.8.2017.)

Summary

APPLICATION OF CLOUD INTEROPERABILITY FOR DATA STORAGE

Cloud computing paradigm is accepted by an increasing number of organizations due to significant financial savings. On the other hand, there are some issues that hinder cloud adoption. One of the most important problems is the vendor lock-in and lack of interoperability as its outcome. The end users are most interested in software as a service model of cloud computing, as it provides ready-made applications. This paper gives an overview of the interoperability of the software as a service on the example of file storage services in cloud. An application that works with three well-known cloud storage services (Dropbox, Google Drive, and Box) has been created.

Keywords: *interoperability, software as a service, cloud computing, cloud data storage.*