

UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

ZAKLJUČNA STROKOVNA NALOGA VISOKE POSLOVNE ŠOLE

POSLOVNI POMEN PLATFORM V OBLAKU (PaaS)

Ljubljana, februar 2017

MATIC KOSMAČ

IZJAVA O AVTORSTVU

Podpisani Matic Kosmač, študent Ekonomske fakultete Univerze v Ljubljani, avtor predloženega dela z naslovom »Poslovni pomen platform v oblaku (PaaS)«, pripravljenega v sodelovanju s svetovalcem red. prof. dr. Tomažem Turkom

IZJAVLJAM

1. da sem predloženo delo pripravil samostojno;
2. da je tiskana oblika predloženega dela istovetna njegovi elektronski obliki;
3. da je besedilo predloženega dela jezikovno korektno in tehnično pripravljeno v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani, kar pomeni, da sem poskrbel, da so dela in mnenja drugih avtorjev oziroma avtoric, ki jih uporabljam oziroma navajam v besedilu, citirana oziroma povzeta v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani;
4. da se zavedam, da je plagiatorstvo – predstavljanje tujih del (v pisni ali grafični obliki) kot mojih lastnih – kaznivo po Kazenskem zakoniku Republike Slovenije;
5. da se zavedam posledic, ki bi jih na osnovi predloženega dela dokazano plagiatorstvo lahko predstavljalo za moj status na Ekonomski fakulteti Univerze v Ljubljani v skladu z relevantnim pravilnikom;
6. da sem pridobil vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v predloženem delu in jih v njem jasno označil;
7. da sem pri pripravi predloženega dela ravnal v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil soglasje etične komisije;
8. da soglašam, da se elektronska oblika predloženega dela uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;
9. da na Univerzo v Ljubljani neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve predloženega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja predloženega dela na voljo javnosti na svetovnem spletu preko Repozitorija Univerze v Ljubljani;
10. da hkrati z objavo predloženega dela dovoljujem objavo svojih osebnih podatkov, ki so navedeni v njem in v tej izjavi.

V Ljubljani, dne _____

Podpis študenta: _____

KAZALO

UVOD	1
1 RAČUNALNIŠTVO V OBLAKU	3
1.1 Opredelitev računalništva v oblaku.....	3
1.2 Arhitektura računalništva v oblaku	4
1.2.1 Infrastruktura kot storitev	5
1.2.2 Platforma kot storitev	5
1.2.3 Programska oprema kot storitev.....	6
1.3 Modeli vzpostavitve oblaka	8
1.3.1 Javni oblak.....	8
1.3.2 Zasebni oblak	8
1.3.3 Hibridni oblak	9
1.3.4 Skupnostni oblak.....	9
1.4 Poslovna in ekonomska vrednost računalništva v oblaku.....	10
1.4.1 Skupni stroški lastništva oblaka	11
1.5 SWOT analiza računalništva v oblaku	11
1.6 Varnostni in pravni vidik računalništva v oblaku	14
1.7 Prihodnost računalništva v oblaku	15
2 PLATFORMA KOT STORITEV	16
2.1 Model tradicionalne platforme	16
2.2 Model platforme kot storitve.....	16
2.2.1 Model aplikacijske platforme kot storitve.....	18
2.2.2 Model povezovalne platforme kot storitve.....	19
2.3 Razlike med tradicionalno platformo in PaaS.....	19
2.4 Tipi platform	20
3 ANALIZA PLATFORM	20
3.1 Google Epp Engine platforma	20
3.1.1 Podatkovna shramba v Google App Engine	22
3.2 Microsoft Windows Azure platforma	23
3.2.1 Windows Azure.....	23
3.2.2 SQL Azure.....	24
3.2.3 Windows Azure AppFabric.....	24
3.2.4 Windows Azure Marketplace.....	25

3.2.5	Varnost Microsoft Windows Azure platforme	25
3.3	Amazon Web Services platforma	26
3.3.1	Amazon Elastic Compute Cloud	26
3.3.2	Amazon Elastic Beanstalk	27
3.3.3	Amazon Simple Storage Service	28
3.3.4	Amazon Simple Queue Service	29
3.3.5	Amazon SimpleDB	29
3.3.6	Amazon Relational Database Service	30
3.3.7	Amazon Elastic MapReduce	30
3.3.8	Amazon CloudFront in CloudWatch	30
4	PRIMERJAVA ANALIZIRANIH PLATFORM.....	31
4.1	Google App Engine	31
4.2	Microsoft Windows Azure	31
4.3	Amazon Web Service	32
	SKLEP	33
	LITERATURA IN VIRI	35
	PRILOGE	
	KAZALO SLIK	
	Slika 1: Storitveni modeli računalništva v oblaku	5
	Slika 2: Upravljanje računalništva v oblaku	7
	Slika 3: SWOT analiza računalništva v oblaku	13
	Slika 4: Vloge v oblačni platformi	17
	Slika 5: Arhitektura oblačne platforme Google App Engine	21
	Slika 6: Arhitektura oblačne platforme Microsoft Windows Azure	23
	Slika 7: Arhitektura oblačne platforme Amazon Web Services	27

UVOD

Računalništvo v oblaku (angl. *Cloud Computing*) je v današnji terminologiji informacijske tehnologije (angl. *Information Technology*, v nadaljevanju IT) vse bolj priljubljena beseda. V zadnjih dveh desetletjih se je računalništvo v oblaku razvilo iz zgolj futurističnih konceptov, ki so bili omenjeni že na začetku druge polovice dvajsetega stoletja, in je danes postalo vseprisoten koncept na področju IT. Cilj koncepta je zagotavljanje avtomatiziranih storitev in nabora tehnologij ter poslovnih modelov za prikaz IT virov kot storitev na zahtevo (Gouda & Acharjee, 2013, str. 1403).

Vlade, podjetja in posamezniki se selijo v oblak zaradi privlačnih lastnosti, ki jih ponuja. Zaradi vse večje potrebe po količini podatkov in računalniških virih predstavlja računalništvo v oblaku privlačne rešitve. Prehod v oblak za organizacije predstavlja veliko spremembo, saj se mora podjetje za uresničitev vrednosti oblaka soočiti z velikimi spremembami notranje IT in filozofije razmišljanja zaposlenih. Ker obstajajo različne vrste oblakov za različne organizacije, se v družbi ustvarja tudi konkurenca. Pri delovanju organizacije oz. spremembah, ki se danes odvijajo vse bolj proaktivno, se kljub vsem tehnološkim spremembam še vedno soočamo z odporom znotraj organizacij do računalništva v oblaku (Ojala, 2016, str. 409; Gupta, Seetharaman, & Raj, 2013, str. 861).

Razvijalec, ki želi razviti in zagnati aplikacije v infrastrukturi oblaka, ne rabi vnaprejšnje kapitalske naložbe v infrastrukturo, saj s sredstvi razpolaga kot s storitvijo (angl. *as-a-service*). Podjetja z vstopom v računalniški oblak bolj obvladujejo stroške infrastrukture in programske opreme. Uporabniku ni treba investirati v nakup strojne opreme, temveč plača le električno energijo in zakup storitve v oblaku ter plača le za tisto, kar zares potrebuje. Ko sistemskih virov več ne potrebuje, je njihova sprostitev enostavna. Zaradi prednosti ekonomije obsega računalništvo v oblaku za mnoge organizacije predstavlja skupek internetnih storitev in aplikacij, ki zadostijo vsem informacijskim potrebam, s katerimi se soočajo, brez stroškov uvajanja, implementacije in vzdrževanja sistemov (Gupta et al., 2013, str. 861).

Izoblikovalo se je več slojev storitev v oblaku. Uporabniki so se najprej srečali s programjem kot storitvijo (angl. *Software as a Service*, v nadaljevanju SaaS), kjer so lahko uporabljali že zgrajene aplikacije preko spletnega brskalnika, ki je omogočal dostop na daljavo. Na nižjih slojih pa se je izoblikovala podlaga storitve kot platforme (angl. *Platform as a Service*, v nadaljevanju PaaS), ki ponuja pomoč celotnemu življenjskemu ciklu programov od razvoja, uporabe in vzdrževanja ter tudi infrastrukture kot storitve (angl. *Infrastructure as a Service*, v nadaljevanju IaaS), predvsem ob pomoči računalniške virtualizacije, ki omogoča dinamično stopnjevanje zmogljivosti z zakupom navideznih strojev glede na trenutne potrebe organizacije oz. posameznika (Geministyle – Storitveni modeli računalništva v oblaku, 2016).

V tipičnem razvojnem okolju računalništva v oblaku, kot je Amazon Web Services (v nadaljevanju AWS), Google App Engine (v nadaljevanju GAE) ali Microsoft Windows Azure (v nadaljevanju MWA), so sredstva hitro priskrbljena in se sprostijo preko vmesnika s funkcijo samopostrežnosti in minimalnim upravljanjem. To omogoča aplikacijam hitro skalabilnost za dodajanje ali odvzem virov in skoraj stoddostno prilagodljivost. Razširljivost se nanaša na sposobnost računalniškega sistema za stalno ohranjanje razpoložljivosti, zanesljivosti in učinkovitosti z večjim številom zahtev od več uporabnikov sistema. Z razširljivostjo razvijalcem in lastnikom aplikacij ni treba več načrtovati in predvidevati končnih zahtev, ki jih bo imela rešitev (Sharma & Garg, 2015, str. 232).

Namen zaključne strokovne naloge je analizirati tri največje ponudnike PaaS na primerih MWA, GAE ter AWS ter analizirati naslednje ekonomske dejavnike: elastičnost, mobilnost, varnost ter ekonomičnost. Kot glavna cilja strokovne zaključne naloge pa smo si zastavili raziskovalni vprašanja, na kateri bomo odgovorili skozi sam potek raziskave:

R1: Ali PaaS res omogoča znižanje stroškov poslovanja in informatike ter pripomore k večji učinkovitosti poslovanja organizacije?

R2: Ali si z najemom platforme v oblaku res zagotovimo hitrejšo vzpostavitev in enostavnejše poslovanje ter s tem omogočimo večjo elastičnost, zanesljivost in razpoložljivost virov?

Pri delu bomo uporabljali opisno metodo. Za zbiranje ustrezne literature in pisanje vsebine bodo uporabljene strokovne in znanstvene literature tujih in domačih avtorjev, članki objavljeni v strokovnih revijah, vendar ne starejši od desetih let, ter strokovne publikacije in uradne spletne strani organizacij in posameznikov.

Zaključno strokovno nalogo bomo razdelili na tri poglavja. V prvem bomo predstavili osnovne koncepte računalništva v oblaku s storitvenimi in izvedbenimi modeli, poslovno vrednost računalništva v oblaku ter pravni in varnostni vidik oblačnega računalništva. V drugem poglavju bomo podrobneje opisali model PaaS in ga primerjali s tradicionalnim modelom, ki se izvaja na lokalnem strežniku razvijalca ter opisali razlike in funkcionalnosti med njima. V zadnjem poglavju pa bomo analizirali tri največje ponudnike spletnih platform v oblaku na trgu: GAE, MWA in AWS.

1 RAČUNALNIŠTVO V OBLAKU

1.1 Opredelitev računalništva v oblaku

Koncept računalništva v oblaku izvira iz leta 1960, ko je računalniški znanstvenik John McCarthy predlagal zamisel o dostavi oz. najemu računalniških storitev za gospodarske in javne organizacije v konceptu javne dobrine (Dukarić & Jurič, 2013, str. 1208).

Računalništvo v oblaku je razvoj porazdeljene računalniške paradigme, ki zagotavlja izredno prožnost, združevanje virov, skladiščenje in velja za tehnologijo, ki bo nadomestila klasično računalništvo v naslednjih nekaj letih (Srinivasan & Raja, 2014, str. 1). Slovenski informacijski slovar termin oblačnega računalništva pojasnjuje kot uporabo oblaka in z njim povezane tehnologije (Računalništvo v oblaku, b.l.).

Nacionalni inštitut za standarde in tehnologijo (angl. *National Institute of Standards and Technology*, v nadaljevanju NIST) računalništvo v oblaku definira kot model za zagotavljanje omrežnega dostopa do deljenega nabora računalniških virov (kot so omrežja, strežniki, diski, aplikacije in storitve), ki jih je mogoče hitro pripraviti za uporabo in hitro sprostiti, oboje z minimalnim trudom in z minimalno interakcijo ponudnika storitve, ter velja za uradno definicijo računalništva v oblaku (Mell & Grance, 2011, str. 2; Sedlar, Bešter, & Kos, 2011).

Storitveni modeli v oblaku so danes trend modela zunanjega izvajanja (angl. *outsourcing*) za uvajanje informacijsko-komunikacijske tehnologije (v nadaljevanju IKT). Velik porast zajema storitev v oblaku smo zaznali v pojavu zadnje gospodarske krize, ki se je začela leta 2008, ko so podjetja zniževala stroške na vsakem koraku. Vse več organizacij, predvsem pa mala in srednje velika podjetja, so iskala cenejše storitve in se odločala za izvajanje informatike v oblaku, ki se je izkazala za stroškovno zelo učinkovit in tehnološko dovršen model v zadnjih dveh desetletjih od pojava računalništva v oblaku (Marian, 2012, str. 135). Potencial računalništva v oblaku ni samo v infrastrukturi, temveč tudi v programski opremi, skladiščenju, varnosti, zanesljivosti in testnih oz. razvojnih okoljih (Rao, 2015, str. 8).

Oblak omogoča tako podjetjem kot posameznikom, da izkoristijo storitve in informacije, ki so na voljo na vsakem mestu, v vsakem trenutku, vendar pod pogojem, da sistem deluje v internetnem omrežju. Vse več podjetnikov in tudi multinacionalk izkorišča priložnosti, ki jih ponuja računalništvo v oblaku. S pojavom modela oblačnega računalništva lahko govorimo o informacijsko-tehnološki revoluciji v poslovnem svetu, saj je v procesu distribucije izdelkov in storitev pripomogel k časovnemu in stroškovnemu prihranku ter povečanju produktivnosti (Bhadauria, Chaki, Chaki, & Sanyal, 2014, str. 162). Modeli storitve predstavljajo velik prihranek stroškov predvsem za mala podjetja. Tako lahko uporabljajo poslovno analitiko, ki potrebuje veliko računalniške zmogljivosti in zmanjšuje

stroške vzdrževanja informacijskih sredstev, kar je posledično za nova podjetja povzročilo znižanje stroškov vstopnih ovir (Gupta et al., 2013, str. 863).

V poslovnem svetu zagotavlja visoke in fleksibilne storitve, do katerih lahko dostopamo in jih prilagajamo preko spletne nadzorne plošče ali neposredno v okolju oblaka. Ena od največjih prednosti uporabe računalništva v oblaku je sprostitev oportunitetnih stroškov upravljanja z IT, ki jih lahko usmerimo na primarne poslovne vidike, kot je rast osnovne dejavnosti (Nath, Aleya, Santra, & Maji, 2016, str. 315).

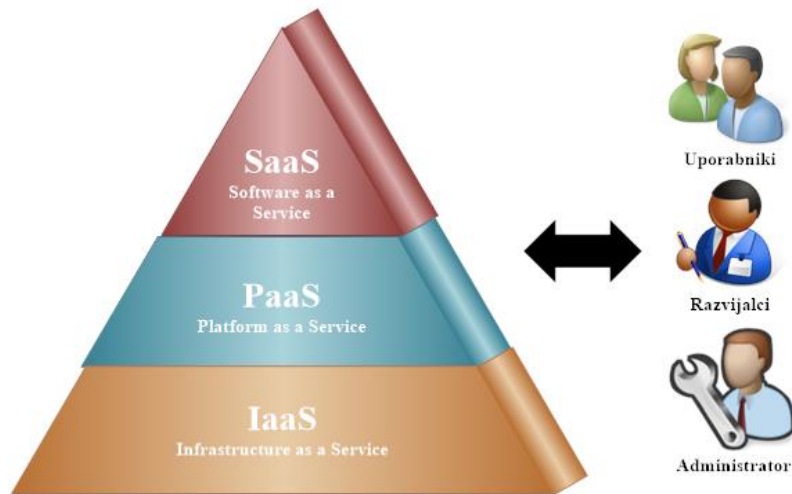
Storitev računalništva v oblaku omogoča hitrejši dostop do aplikacij in zmanjša stroške infrastrukture (Vouk, 2008, str. 243). Analitično-informacijska hiša Gartner je v svoji študiji objavila, da je bila vrednost trga računalništva leta 2013 v oblaku vredna 138 milijard ameriških dolarjev, leta 2015 je dosegla 150 milijard ameriških dolarjev ter bo v naslednjih letih vrednost le še naraščala. Po teh ocenah lahko govorimo, da računalništvo v oblaku predstavlja velik potencial za inovacije in razvoj IT v oblaku (Srinivasan & Raja, 2014, str. 1).

1.2 Arhitektura računalništva v oblaku

Arhitektura računalništva v oblaku je sestavljena iz mnogih komponent, ki sestavljajo oblak in so povezani v ohlapno celoto. Arhitekturo lahko delimo na sprednji in zadnji konec, vsak izmed koncev pa je povezan preko internetne povezave. Sprednji konec se nanaša na stranko oz. na del porabnikovega sistema računalništva v oblaku ter je sestavljen iz uporabniškega vmesnika in aplikacije, ki sta osnovno potrebni komponenti za odjem oblačne storitve. Zadnji del se nanaša na sam oblak in je na strani ponudnika storitev. Sestavljen je iz vseh sredstev, ki so potrebna za zagotovitev storitev v oblaku. Obsega velike in zmogljive podatkovne centre, virtualne stroje, varnostne mehanizme, modele uvajanj strežnikov ter ostalo tehnologijo (Nath et al., 2016, str. 316).

Storitvene modele računalništva v oblaku lahko porazdelimo na tri plasti (Slika 1). Najnižja plast računalniško oblačne piramide je IaaS, v sredini je model PaaS in na najvišji ravni najdemo model SaaS. Pri uporabi storitvenih modelov kaj hitro opazimo skorajšnjo premo sorazmernost med fleksibilnostjo in izgubo nadzora nad sistemom. Bolj, ko se pomikamo proti vrhu piramide v storitev SaaS, manjši je nadzor nad sistemom, medtem ko se fleksibilnost povečuje v primerjavi z IaaS modelom (Bhadoria et al., 2014, str. 161).

Slika 1: Storitveni modeli računalništva v oblaku



Vir: R. Dukarić, R. Povše, & M.B. Jurič, *Računalništvo v oblaku*, 2011, str. 7.

1.2.1 Infrastruktura kot storitev

IaaS je ena od najpomembnejših plasti računalništva v oblaku (Dukarić & Jurič, 2013, str. 1197). Omogoča poseben virtualni strežnik z edinstvenim IP-naslovom in bloki za shranjevanje, ki so prilagodljivi glede na povpraševanje. Tako kupci plačajo le znesek storitve, ki ga dejansko porabijo (Nath et al., 2016, str. 316). Doprinaša k možnosti oskrbovanja procesnih, podatkovnih in omrežnih kapacitet ter drugih računalniških virov. Uporabnik lahko postavi in zaganja poljubno programsko opremo, ki vključuje operacijski sistem in aplikacije (Mell & Grance, 2011, str. 3). Tipična prednost modela IaaS je, da se organizacije zanjo odločijo, ker ne želijo imeti lastnih infrastruktur in potrebnih resursov za vzdrževanje (Prasanth, 2012, str. 27). Strošek infrastrukture tako predstavljajo le porabljeni viri (Gouda, Kar, & Patra, 2013, str. 2454), ki so na voljo v obliki virtualizacije, katera predstavlja osnoven gradnik IaaS modela (Sedlar et al., 2011).

Najbolj prepoznavni primer storitev IaaS nudi ponudnik Amazon z modeli Amazon Simple Storage Solution (v nadaljevanju Amazon S3) in Amazon Elastic Compute Cloud (v nadaljevanju Amazon EC2). Med ostalimi ponudniki najdemo še IBM Computing on Demand, Rackspace in Microsoft Live Mesh ter ostale (Chakravarthy, Kannan, & Kannan, 2014, str. 1225).

1.2.2 Platforma kot storitev

PaaS je sklop programske opreme in razvojnih orodij, ki gostujejo na strežnikih izvajalca, in predstavlja vmesno programsko opremo (angl. *middleware*) (Prasanth, 2012, str. 27). Je

storitev za razvoj lastnih aplikacij in njihovo uporabo preko spleta za razvijalce. PaaS omogoča hitrejši in stroškovno učinkovitejši model za razvoj aplikacij in dostavo, saj zagotavlja celotno infrastrukturo, potrebno za izvajanje aplikacij preko interneta (Gouda et al., 2013, str. 2454).

PaaS vsebuje vse potrebne mehanizme za podporo in izvedbo celotnega življenjskega cikla aplikacije in zagotavlja dostop izključno preko internetne povezave brez nameščanja oz. investiranja v drago programsko opremo za končne uporabnike (Mell & Grance, 2011, str. 3). Za razliko od modela IaaS, kjer razvijalci ustvarijo lasten primer z lastnim operacijskim sistemom na lokalnem strežniku, se PaaS razvijalci ukvarjajo samo z razvojem in jih ne zanima, kateri operacijski sistem se uporablja. Aplikacije so razvite s pomočjo programskih jezikov, ki jih podpira ponudnik oblačne storitve (Karunakaran, Krishnaswamy, & Rangaraja, 2015, str. 583).

Primeri programskih jezikov za platformo Google App Engine so Python, Java, GO, in .NET pri Microsoftovi platformi. Vsaka platforma podpira določene programske jezike, zato je migriranje na ponudnike konkurenčnih platform dokaj otežena naloga (Pardeshi, 2014, str. 592).

Mell in Grance (2011) pa ne izključujeta možnosti uporabe združljivosti programskih jezikov, orodij in storitev, ki jih podpira več ponudnikov PaaS. Spletna platforma svojim uporabnikom skriva infrastrukturo, tako fizično kot virtualno strojno opremo (Sedlar et al., 2011). PaaS je sposobna vključiti vse vrste tekočih poslov, navadno kot aplikacijski strežnik z dostopom do poslovno inteligenčne sposobnosti in poslovne odprte platforme (Karunakaran et al., 2015, str. 583). Izračunava zmogljivost infrastrukture glede na poslovne zahteve in poziva vire strojne opreme, ki temeljijo na API-jih (angl. *Application Program Interface* v nadaljevanju API) preko modela IaaS (Gouda et al., 2013, str. 2454). Primeri ponudnikov platform so Google App Engine, Microsoft Azure, Apprenda in ostali (Cloud360.com – The Top 20 Platform as a Service Vendors, 2016).

1.2.3 Programska oprema kot storitev

Model SaaS je najširša ponudba storitev v oblaku na trgu. Njen osnovni namen je najem programske opreme za različne odjemalce (Nath et al., 2016, str. 316). Pri tem modelu ponudnik stranki ponuja in omogoča uporabo le svojih aplikacij, komunikacija z uporabnikom pa poteka preko uporabniškega vmesnika (Gouda et al., 2013, str. 2454).

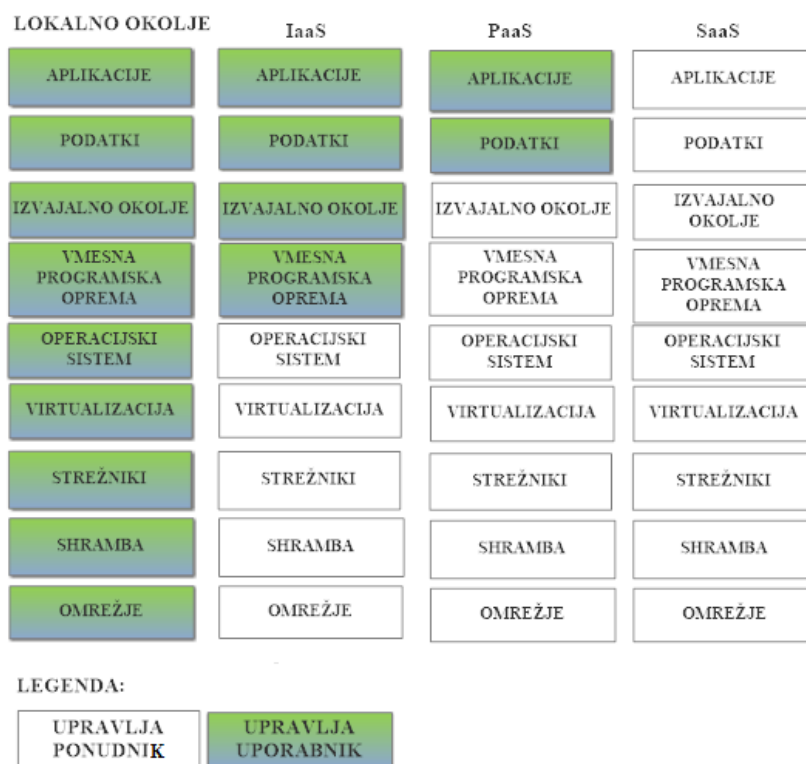
Aplikacije v oblaku so oblačno usmerjene storitve in nameščene v sistemih, do katerih lahko posameznik dostopa od kjer koli kadar koli preko svojega uporabniškega računa z geslom, upravljanje lokacij pa poteka iz centralne lokacije (Rao, 2015, str. 8). Prednost modela SaaS je, da uporabnikom ni treba nadgrajevati programske opreme in skrbeti za posodobitve (Prasanth, 2012, str. 26). Dostava aplikacij poteka na dva načina, in sicer po

modelu zaračunavanja po porabi (angl. *Pay-As-You-Go*) ali v obliki naročnine (Vouk, 2008, str. 245).

Primer najbolj uporabljenih SaaS storitev so Twitter, Facebook in ostala socialna omrežja. Novonastala in mala podjetja si lahko zdaj privoščijo aplikacije, kot so celovite programske rešitve (angl. *Enterprise Resource Planning*, v nadaljevanju ERP), programska orodja za upravljanje odnosov s strankami (angl. *Customer Relationship Management*, v nadaljevanju CRM), programska oprema za optimizacijo prodaje podjetja Sales Force (angl. *Sales Force Automation – SFA*) in programska orodja managementa oskrbovalne verige (angl. *Supply Chain Management – SCM*) zaradi ekonomičnosti tovrstnih storitev programske opreme v oblaku (Tomić & Jovanović, 2016).

Takojsen dostop do virov strojne opreme je na voljo brez vnaprejšnja kapitalске naložbe, posledično pa je hitrejši odzivni čas na trgu z lažje dostopno IT (Tiwari & Mishra, 2012, str. 309).

Slika 2: Upravljanje računalništva v oblaku



Vir: R. Dukarić, R. Povše, & M.B. Jurič, *Računalništvo v oblaku*, 2011, str. 15.

1.3 Modeli vzpostavitve oblaka

1.3.1 Javni oblak

Znan je tudi kot zunanji oblak (angl. *public cloud*), ki je na voljo vsakemu in je odprt za splošno uporabo. Lahko je v državni ali vladni lasti oz. v kombinaciji z javno organizacijo, ki ga upravlja. Model predstavlja prosto dostopna okolja za širšo javnost z namenom znižanja stroškov, saj izkorišča zunanje vire. Vsak je lahko stranka in dostopa do njega z nizkimi stroški ali celo brezplačno. Javni oblak lahko gosti posamezno storitev ali knjižnico storitev (Rani & Ranjan, 2014, str. 459).

Gre za skupek sistemov in storitev za enostavno dostopnost širši javnosti, vendar se moramo zavedati, da je zaradi svoje narave manj varen od zasebnega oblaka (Nath et al. 2016, str. 316). Na takšen način podjetja ponujajo svoje storitve zunanjim uporabnikom z dostopom do njihove infrastrukture neposredno preko interneta. Obstaja kar nekaj prednosti javnega oblaka. Zagotavlja relativno zanesljivo in varno podatkovno skladiščenje v primerjavi z drugimi metodami shranjevanja glede na ceno storitve. Javni oblak uporablja veliko organizacij in posameznikov, saj so lokalni računalniki velikokrat tarča fizičnih ali hekerskih napadov. Če so podatki shranjeni v skladišču javnega oblaka, se storilec ne more okoristiti zaradi nepooblaščenega dostopa do njih (Winkler, 2011, str. 40).

1.3.2 Zasebni oblak

Zasebni oblak (angl. *private cloud*) je zgrajen za posamezne osebe ali organizacije, ki lahko učinkovito nadzorujejo podatke pod visoko varnostjo in kakovostjo storitev. Podjetje ima infrastrukturo in manipulira z uvajanjem aplikacij preko zasebnega oblaka, katerega temeljna vrednota so zasebna sredstva. Zasebni oblak lahko ustanovi ponudnik storitev ali podjetje, za katerega ponudnik oblačnih storitev upravlja in namešča konfiguracijo in vzdržuje infrastrukturo zasebnega oblaka v svojem poslovnem podatkovnem centru (Mell & Grance, 2011).

Prednosti zasebnega oblaka so več kot očitne. Podatki so relativno bolj varni od drugih metod uporabe vzpostavitvenih modelov, saj omogoča sisteme in storitve za potrebe, kjer potrebujejo večjo varnost in zasebnost za varovanje poslovnih skrivnosti. Ta je navadno zgrajen za požarnim zidom namesto na oddaljenem podatkovnem centru na neznani lokaciji. Prednosti zasebnega oblaka so tudi v integraciji s starejšimi modeli za njegovo izgradnjo, saj so sposobni izkoristiti obstoječo strojno opremo in aplikacije v primerjavi z javnim oblakom, kljub višjim investicijam in usposobljenemu kadru s področja informatike, ki jih terjajo. Vsaka organizacija ima za svoj zasebni oblak svoje procese upravljanja, kar je predvsem privlačno za podjetja, ki ponujajo edinstvene proizvode ali storitve (Nath et al. 2016, str. 316; Rani & Ranjan, 2014, str. 459).

1.3.3 Hibridni oblak

Hibridni oblak (angl. *hybrid cloud*) je mešanica javnega in zasebnega oblaka ali več različnih infrastruktur v oblaku, medsebojno povezanih s standardno tehnologijo, ki omogoča prenos podatkov, in aplikacij z izvajanjem kritičnih aktivnosti z uporabo zasebnega oblaka, medtem ko nekritične aktivnosti izvajamo s pomočjo javnega oblaka (Rani & Ranjan, 2014, str. 459). Kombinacija javnega in hibridnega oblaka je nepogrešljivi model, saj prenos podatkov nima skupnega medsebojnega vpliva (Nath et al., 2016, str. 316).

Analitska hiša Gartner napoveduje, da bo imela več kot polovica velikih podjetij do leta 2017 vzpostavljene hibridne oblake, če seveda upoštevamo infrastrukturno obravnavo. Razlog za širjenje oblačne hibridnosti je velikokrat v tem, da so se podjetja pri uvajanju zasebnega oblaka osredotočala predvsem na določeno tehnologijo, namesto da bi reševala procesne izzive in izvedla preobrazbo IT funkcije. Prav zato je z vidika vsebine in poslovnih funkcij razširjenost hibridnih okolij danes mnogo večja kot z vidika infrastrukture. To zagotovo potrjuje tudi elektronsko poslovanje, kjer se skoraj ves promet z e-računi izvaja prek storitev v oblaku. Dodajanje virov iz javnih oblakov je cenovno ugodno, ne obremenjuje kapitala, predvsem pa se plača le za tisto, kar se v resnici potrebuje. Čeprav gre pri računalništvu v oblaku v resnici še vedno predvsem za storitve gostovanja, predvsem modela IaaS, pa se pri obravnavi hibridnih oblakov hitro pojavita tudi druga modela, PaaS in SaaS (IKT Informator – Finance, 2016).

1.3.4 Skupnostni oblak

Skupnostni oblak (angl. *community cloud*) se nanaša na okolje za posebne namene v skupni rabi več organizacij, ki sodelujejo v skupni domeni. Model uvajanja deli sredstva z mnogimi organizacijami v skupnosti, ki si med seboj delijo skupne interese, kot so varnost, upravljanje in skladnost, ter prispevajo svoj del infrastrukture (Rani & Ranjan, 2014, str. 459; Mell & Grance, 2011).

Model predstavlja velik potencial za podjetja in organizacije, ki so predmet enakih regulativ, skladnosti ali skupne pravne omejitve (Winkler, 2011, str. 41). Zgrajeni so na mestu, kjer imajo uporabniki podobne zahteve z enotnimi storitvami. Za primer lahko vzamemo Univerze, katerih uporabniki so profesorji, študentje in zaposleni na univerzi in svojih matičnih fakultetah ter raziskovalnih ustanovah. Storitve vključuje oblak strežnikov, oblak skladiščenja in podatkovni center v oblaku (Goyal, 2014, str. 25).

1.4 Poslovna in ekonomska vrednost računalništva v oblaku

Na področju IT je zelo težko oceniti in izmeriti koristi. Na primer rešitve za varnost, kot sta požarni zid in protivirusna programska oprema, ne morejo ustvarjati koristi, ki se lahko merijo (Gheorghe, 2012, str. 90).

Ključne kategorije stroškov, ki so najpogosteje povezane z investicijami v IKT in njihovo spremljanje v oblaku v primerjavi s tradicionalnim računalništvom, so stroški storitev varnostnega spremljanja za vse požarne zidove in naprave za odkrivanje vdorov v strežnike in usmerjevalnike ter vzdrževanje notranje arhitekture skupaj s stroški planiranja in razvijanja, ki so za odjemalca storitev v oblaku skoraj nični (Rittinghouse & Ransome, 2009, str. 47).

Ko prestopimo ali se odločimo za poslovanje v oblaku, moramo na strateški ravni jasno definirati, kdo je odgovoren za upravljanje oblaka, kako ga bomo upravljali ter kako bo oblak spreminjal naše poslovanje. Računalništvo v oblaku ustvarja poslovno vrednost le, če se zavemo, kaj je dejansko najbolj pomembno za organizacijo in imamo realna pričakovanja. Ponuja kreativni način poslovanja, boljšo izkoriščenost IT, možnost sproščanja virov z osredotočenjem na primarnih kompetencah ter omogoča zmanjšanje operativnih stroškov in takojšen dostop do novih in cenejših načinov za doseganje poslovnih ciljev (Komšo, b. l.).

Najbolj pomembni dejavniki, ki zagotavljajo ključne poslovne prednosti za organizacije z uporabo storitev računalništva v oblaku, so naslednje (Reed & Bennett, 2011; Mell & Grance, 2011):

- **agilnost** (angl. *Agility*): prinaša velike časovne prihranke, saj omogočajo visoko razpoložljivost ter hitro dostopnost do potrebnih storitev, posledično pa je organizacija zaradi prihranka tudi bolj produktivna in odzivnejša ter osredotočena na konkurenco, kupce ter dana sredstva;
- **osredotočanje na primarno raven organizacije** (angl. *Business focus*), saj organizacija uporablja in gostuje na storitvah, ki jih potrebuje in ji prinašajo zahtevane informacije, ideje in povratne informacije s strani kupcev oz. tretjih oseb;
- **nadzor nad stroški in razpoložljivimi sredstvi** (angl. *Cost and Budget control*): storitev se zaračunava na podlagi naročnine ali na podlagi porabljenih in zakupljenih količin sredstev. Tako organizaciji ni več treba skrbeti za sistemski vprašanja in varnost informatike, saj je to delo ponudnika storitev v oblaku;
- **optimizirana struktura** (angl. *Optimized infrastructure*): govorimo o več najemniškem modelu (angl. *multitenancy model*). Model omogoča, da so storitve zmogljivosti računalništva v oblaku dostopne več uporabnikom;
- **skalabilnost in upravljanje kapacitet** (angl. *Scalability and capacity management*): storitve računalništva v oblaku so zgrajene tako, da se prilagajajo številnim zahtevam

in tako imamo zagotovljeno normalno delovanje tudi, ko je povečan obseg obremenitve. Računalništvo v oblaku omogoča takojšno povečanje zmogljivosti virov in tudi njihovo zmanjšanje, če je to potrebno, zato lahko govorimo o veliki elastičnosti oblačnih storitev;

- **vzdrževanje** (angl. *Maintenance*): uporabnik storitev praktično nima več opravka z vzdrževanjem in nameščanjem posodobitev, saj je to naloga ponudnika storitev, uveljavljanje sprememb na relaciji ponudnik–uporabnik pa je takojšnje;
- **mobilitet** (angl. *Mobility*): dostop do storitev računalništva v oblaku lahko opravimo ne glede na lokacijo, edina zahteva je dostop do interneta iz katere koli IKT naprave.

Ekonomsko pa lahko investicijo v oblak ovrednotimo po naslednjem kazalniku (ISACA, 2012):

1.4.1 Skupni stroški lastništva oblaka

Pred vložkom v vsako investicijo morajo podjetja razumeti vrednost naložbe, ki jo ta prinaša v organizaciji. Z izračunom skupnih stroškov lastništva (angl. *Total Cost of Ownership – TCO*), ki je merilno orodje za ocenjevanje neposrednih in posrednih stroškov, ki sodelujejo pri izvedbi investicije, se pokaže stroške za celotno življenjsko dobo infrastrukture. Cilj seštevek vseh spremenljivk v formuli je izračun življenjskih stroškov za pridobivanje, upravljanje in vzdrževanje storitev oz. sredstev, ki so koristni za določanje razlike med kupnimi in dolgoročnimi stroški investicije (Aggarwal & McCabe, 2009, str. 4–5). Izračun skupnih stroškov lastništva izračunamo po enačbi (1):

$$\begin{aligned} TCO = & \textit{nakup} + \textit{financiranje} + \textit{vzdrževanje} + \textit{nadgradnja} + \\ & + \textit{izboljšave} + \textit{namestitev} + \textit{varnost} + \textit{amortizacija} + \textit{razgradnja} \\ & + \textit{odstranitev} + \textit{stroški}_n \end{aligned} \quad (1)$$

Obdobje, ki ga vzamemo za izračun skupnih stroškov lastništva, je odvisno od standardov posamezne organizacije, ki določi, kdaj se lastništvo začne in kdaj konča. Življenjsko obdobje razdelimo na splošno v tri skupine (ISACA, 2012, str. 14):

- obdobje amortizacije;
- ekonomsko obdobje;
- servisno obdobje.

1.5 SWOT analiza računalništva v oblaku

Iz predhodno narejene analize prednosti, slabosti, priložnosti in nevarnosti (angl. *Strengths, Weaknesses, Opportunities, Threats – SWOT*) (Slika 3) vidimo, da ima računalništvo v oblaku veliko ekonomično prednost pred tradicionalnim računalništvom, saj je ključni

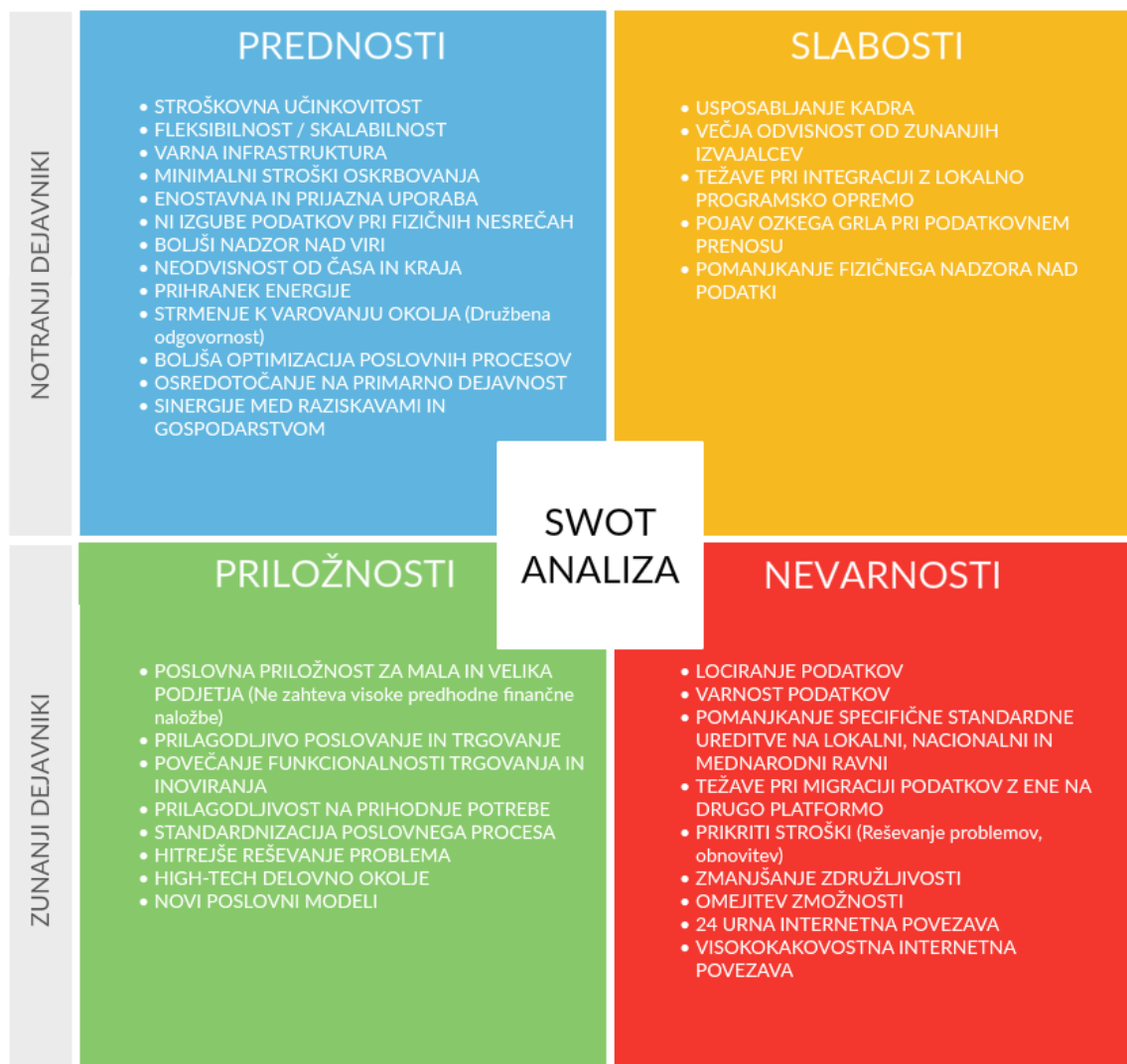
faktor sprememba vloge v interesnih skupinah uporabe. Pri modelu tradicionalnega računalništva je potrošnik odgovoren za vzdrževanje in nadgradnjo tako programske kot strojne opreme, pri oblacnem računalništvu pa ponudnik storitev pristopa in opravlja vzdrževanje in nadgradnjo sistema ter je odgovoren za varovanje podatkov uporabnika. Naročnik oz. uporabnik v tem primeru le proti plačilu zakupi sistem ter ga nato uporablja. Uporaba in zajem storitev v oblaku privede do nižjih kapitalskih naložb v informatiko v podjetju ter nižjih ostalih stroškov, kot so vzdrževanje in ostala dela, ki jih navadno zahteva lastna infrastruktura. Oblak omogoča dostop do storitev kadar koli in od kjer koli, tako da nismo več odvisni od časa in kraja dostopa do podatkov, omogoča večjo prilagoditev in specializacijo aplikacij lastnim potrebam, interaktivno sodelovanje med uporabniki, veliko energetske učinkovitost ter s tem strateško zasledovanje družbene odgovornosti podjetij in pridobitev standardov, ki so v današnjem poslovnem svetu zelo pomembni in se odražajo tudi v nefinančnih dejavnikih podjetja (Ghaffari, Delgosha, & Abdolvand, 2014, str. 16–18).

Analize kažejo, da razen velikim multinacionalkam, ki v večji meri uporabljajo računalništvo v oblaku, predstavlja poslovni trend tudi za mala in srednja podjetja skozi vidik zmanjšanja stroškov v infrastrukturo in strojno opremo, hitrejšo in stroškovno učinkovitejšo nadgradnjo programske opreme, lažji in hitrejši dostop do aplikacij ter učinkovitejšo uporabo računalniških virov, ki jih omogoča računalništvo v oblaku za organizacije. Kljub temu da obstajajo tveganja, ga v današnjem poslovnem svetu uporablja vse več organizacij, od malih podjetij do velikih multinacionalk. To se predvsem odraža v hitrejšem odzivnem času pri trženju s pomočjo socialnih omrežij, kot sta Facebook in Twitter, ki igrajo pomembno vlogo pri spodbujanju inovativnosti, ter spremljanju nakupnih navad s CRM orodji, ki predstavljajo nov pogled na kupca ter filozofijo podjetja s pomočjo IT (Marešová & Kuča, 2015, str. 61–68).

Dodatni oz. prikriti stroški (angl. *cover cost*) se pojavijo v primeru okvar, nesreč in ob izgubi podatkov, za katere moramo imeti vnaprej pripravljene načrte za obnovo ter odpravo napak s čim manjšimi motnjami. Poslovne aplikacije danes zahtevajo in bodo v prihodnje zahtevale veliko zanesljivost ter visoko zmogljivo operacijsko podporo. Poleg visoke zanesljivosti potrebujemo tudi neprestano štiriindvajseturno zanesljivost povezave, saj je delovanje storitev v oblaku odvisno od velike hitrosti, velika hitrost pa od internetne povezave. Za to potrebujemo dostopnost do električne energije, v nasprotnem primeru privede do izpada in s tem se povezava izgubi. Z najemom storitev v oblaku in predvsem z najemom ERP rešitev v podjetju dosežemo večjo standardizacijo poslovanja, ta pa zahteva razširljivo infrastrukturo, ki izhaja iz čim večje integracije podatkovnih baz. Računalništvo v oblaku je v veliki meri odvisno od političnih interesov, ki jih določajo z zakoni na lokalni ravni. Zakoni se lahko močno razlikujejo med seboj in ustvarjajo vrzel do globalnega sprejemanja računalništva v oblaku, s tem pa je povezana tudi odprtost, povezljivost in dovolj velika pasovna širina ter dodeljevanje sredstev za IKT infrastrukturo, katerih lastnik je v večji meri država, zato lahko govorimo, da je delovanje in prihodnost oblacnega

računalništva močno odvisna od globalne in lokalne politike (Ghaffari et al., 2014, str. 16–19).

Slika 3: SWOT analiza računalništva v oblaku



Povzeto in prirejeno po K. Ghaffari, S.M. Delgosha, & N. Abdolvand, Towards Cloud Computing: A SWOT Analysis On Its Adoption In Smes, 2014, str. 15; P. Marešová & K. Kuča, Tehnological environment and SWOT Analysis of Cloud Computing in Europe, 2015, str. 62–68; M. Masrom & A. Rahimli, Cloud Computing Adoption in the Healthcare Sector: A SWOT analysis, 2015, str. 14–17.

Kragelj in Rajkovič (2012) sta za oceno ponudnikov storitev v oblaku ter za izbiro najprimernejšega uporabila tri okvire meril odločanja. Prvo merilo se nanaša na tehnični sklop, ki zajema predvsem varnost poslovanja z združljivostjo sedanje infrastrukture ter mobilnostjo poslovanja. Z vidika drugega merila, organizacije in poslovanja podjetja sta vključila finančne kategorije ter način in potek samega poslovnega procesa v podjetju. V praksi ključno težavo opažata pri izračunu vseh stroškov, povezanih s postavitvijo in vzdrževanjem strežniške infrastrukture ter doseženih koristi. Podjetja razdelimo v tri

osnovne dejavnosti, in sicer trgovsko, proizvodno in storitveno, za vsako od teh pa moramo upoštevati, da potrebujejo različen vpliv IT in da je v nekaterih primerih migriranje v oblak težje izvedljivo. Za tretje merilo, ki sta ga opredelila, pa obravnavata predvsem pravni vidik in intelektualno lastnino, predvsem preučevanje pogodbe o uporabi storitev v oblaku ter definirane avtorske pravice. Eden od ključnih faktorjev za pravilno izbiro ponudnika storitev v oblaku je strateški poslovni načrt, ki je opredeljen za naslednjih nekaj let vnaprej. Za dokončno odločitev izbire ponudnika pa navajata, da je odločilen človeški faktor (Kragelj & Rajkovič, 2012, str. 245–248).

1.6 Varnostni in pravni vidik računalništva v oblaku

Varnost v oblaku predstavlja velik nabor politik, IT, nadzora in metod za organiziranje zaščite podatkov, aplikacij in vso infrastrukturo, ki jo računalništvo v oblaku predstavlja (Srinivasan & Raja, 2014, str. 1). Varstveni vidik je pomemben in velikokrat tudi kritičen, predvsem zaradi nezaupanja pri migriranju podatkov v oblak. Odpira se vrsta vprašanj in z njimi povezanih težav, zato mora ponudnik storitev v oblaku potrošniku zagotoviti, da je oblak dovolj varen in odporen na zunanje grožnje s področja kibernetkega kriminala (Nath et al., 2016, str. 315).

Varnostni vidik je treba opravljati na dveh ravneh. Prvi je na ravni izvajalca, drugi pa je na ravni uporabnika. Predvsem je treba dobro zavarovati tako fizični kot tehnološki dostop do strežnikov, da najemnik storitev ne naleti na težave, kot so izguba ali kraja podatkov (Kaur, Sood, Garg, & Palta, 2016, str. 38). Prvo vprašanje varnosti se nanaša na občutljivost podatkov v oblaku, saj ob prenosu podatkov v oblak te damo na razpolago širši javnosti, nahajajo pa se na strežniku, katerega lokacija nam je neznana (Radut, Popa, & Codreanu, 2012, str. 173). Tako obstaja potreba po integriteti podatkov v oblaku. Kraja podatkov je resno kaznivo dejanje in v tovrstnem položaju se znajdemo kaj hitro, saj je strogost zakonodaje na tem področju na najvišji ravni. V Sloveniji tovrstno krajo obravnavamo po Kazenskem zakoniku (Ur. l. RS, št. 55/2008-KZ-1, 66/2008-KZ-1, 39/2009-KZ-1A, 91/2011-KZ-1B, 50/2012-UPB2). Namen zakona je preprečevanje vdora v informacijski sistem ter zlorabljanje informacij (143. člen Kazenskega zakonika). Obravnava tudi varovanje osebnih podatkov in njihovo občutljivost ter preprečevanje ponarejanja listin (242. in 256. člen Kazenskega zakonika).

Zaupnost podatkov je pomembno tudi v oblaku. Ker se vse obravnava s strani tretje osebe, mora ponudnik storitev biti zelo previden, da so strankini podatki dobro zavarovani pred drugimi ponudniki, kupci, uporabniki in ostalimi uporabniki. Večino oblaka sestavljajo virtualni stroji, zato je zaupnost še toliko težje vzpostaviti in se še danes veliko dela na zaščiti, integriteti ter etičnem vidiku uporabe in varovanju oblaka (Sridevi & Rahinipriyadharshini, 2016, str. 1748–1753).

Veliko tveganj, ki so pogosto povezana z oblakom, ni novih in jih lahko najdemo v večini podjetij tudi danes (Delak & Terčelj, 2011). Pred tovrstnimi dejanji nas v Sloveniji pravno ščiti Zakon o varstvu osebnih podatkov (Ur. l. RS, št. 86/2004, 133/2005-ZInfP, 51/2007-ZUstS-A, 67/2007, 94/2007-UPB1), namen katerega je predvsem preprečevanje nezakonitega in neupravičenega posega v zasebnost posameznika, ko govorimo o transakciji ter obdelavi osebnih podatkov (npr. spletno nakupovanje in spletno bančništvo) in njihovem zbiranju ter uporabi.

Preden se odločimo za poslovanje v oblaku, moramo preveriti, katere varnostne IT uporablja ponudnik za zaščito, preverimo poslovno prakso ponudnika, pridobljene standarde in splošno zadovoljstvo uporabnikov ponudnika. Ponudnik mora imeti dobro infrastrukturo, varne strojne elemente in komponente, kot so požarni zidovi, usmerjevalniki, strežniki, operacijski sistem, programska opremo in druge potrebne tehnologije (Chakravathy et al., 2014, str. 1225–1227).

Ponudnik mora zagotavljati tudi varnost z vidika šifriranja podatkov pri razvoju aplikacij. IT managerji ponudnika morajo imeti dobro opredeljen načrt in strategijo za zagotavljanje ključne varnosti, ki je osnovna zahteva uporabnikov. Na trgu obstaja veliko nove tehnologije, ki je v vzponu in ima namen človeku olajšati življenje, vendar moramo biti previdni z varnostnega vidika in se zavedati, da je bistvo dobre infrastrukture ščitenje pred kibernetскими napadi (Nath et al., 2016, str. 317).

1.7 Prihodnost računalništva v oblaku

Z računalništvom v oblaku in tehnologijami, ki jih ponuja, obstaja veliko potencialnih možnosti in sposobnosti za ustvarjanje popolnoma novega sveta delovnih mest, platform, aplikacij in ostalih storitev. Glavni razlog za močan razvoj prihodnosti oblaka obstaja zaradi učinkovite izmenjave podatkov na mednarodni ravni. Po ocenah svetovnega trga naj bi v naslednjih treh letih računalništvo v oblaku prehitelo uporabo krajevnih rešitev. Tako naj bi namesto na lokalnih strežnikih večina podjetij svoje poslovne aplikacije poganjala na oddaljenih strežnikih (Apprenda – Future of Cloud Computing, 2016).

Gartner napoveduje, da bo do leta 2020 več kot polovica razvitih aplikacij temeljila na internetu stvari (angl. *internet of things*, v nadaljevanju IoT), ki temelji na brezžični telekomunikacijski tehnologiji. Tovrstna tehnologija bo spodbujala dodatno uporabo PaaS za razvijanje poslovnih aplikacij, ki bodo temeljili na predelavi velikih količin podatkov in bodo transformirale dosedanje prakse oblikovanja aplikacij (Gartner IoT – Says IoT Adoption Is Driving the Use of Platform as a Service, 2016). Predvsem manjša podjetja se ne želijo in tudi ne morejo ukvarjati s tehničnimi vprašanji IT, temveč le s samo storitvijo oz. izdelkom, ki jih ponujajo. Tako lahko računalništvo v oblaku pomaga v prihodnje, predvsem manjšim podjetjem premagati pomanjkljivosti nepoznavanja poslovne informatike (Bojanc & Šušmak, 2013, str. 43–47). Analitska hiša IDC pa napoveduje, da

bo po njihovih ocenah tretjina vseh informacij na internetu domovala v računalniških oblakih oz. bodo v nekem obdobju z njimi povezane (Geministyle – Prihodnost računalništva v oblaku, 2016).

2 PLATFORMA KOT STORITEV

Računalništvo v oblaku pomeni prelomnico tako s tehnološkega kot poslovnega vidika. Pri modelu PaaS uporabnik najame storitev za razvoj in produkcijsko okolje, v katerem razvija poljubne aplikacijske rešitve brez investiranja v infrastrukturo za izgradnjo in uvajanje rešitev, tako kot pri tradicionalnem razvijanju rešitev na lokalnem strežniku (Kragelj & Rajkovič, 2012, str. 244).

2.1 Model tradicionalne platforme

Izgradnja in izvajanje aplikacije ter nadaljnji razvoj ter vzdrževanje je bilo vedno zapleteno ter drago opravilo, velikokrat pa tudi tvegano. Vsaka aplikacija je potrebovala svoje vire in terjala svojo vlogo s strani razvijalcev ter programske opreme, strojne opreme, podatkovne baze, vmesne programske opreme, spletnih strežnikov, operacijskih sistemov in druge IT. Ko je bil sistem vzpostavljen, je morala skupina razvijalcev za sam razvoj aplikacije uporabljati zahtevna in kompleksna programska orodja, kot sta J2EE in .NET. Vsak projekt je zahteval ekipo za sistemsko administracijo, ki je nadzirala omrežje, podatkovne zbirke in strokovnjake, da je delo potekalo gladko in tekoče. Vsaka bolj zahtevna poslovna aplikacija je potrebovala tudi dodeljevanje poslovnih vlog ter velikokrat tudi dolgo in zamudno testiranje in uvajanje ter vzpostavitveni cikel. Poleg drage in občutljive IT podjetje potrebuje tudi primerno infrastrukturo tehnoloških centrov, kjer je oprema fizično locirana (Kulkarni, Khatawkar, & Gambhir, 2011, str. 117; Rittinghouse & Ransome, 2009, str. 49).

Celotna arhitektura tradicionalnega modela skupaj terja tudi ogromno porabo električne energije za pogon vseh strežnikov kot tudi sistemov ter njihovo ohlajanje. Na koncu pa je bilo treba vse skupaj še zavarovati in zagotoviti dvojnik oz. t.i. povezovanje dveh ali več trdih diskov (angl. *Redundant array of independent disks*, v nadaljevanju REST), katerega namen ni bil samo povečati in vzpostaviti hitrejšo ter zanesljivejšo logično enoto, temveč tudi pridobiti na varnosti podatkov in zagotoviti obnovo v primeru nesreče (Kulkarni et al., 2011, str. 117).

2.2 Model platforme kot storitve

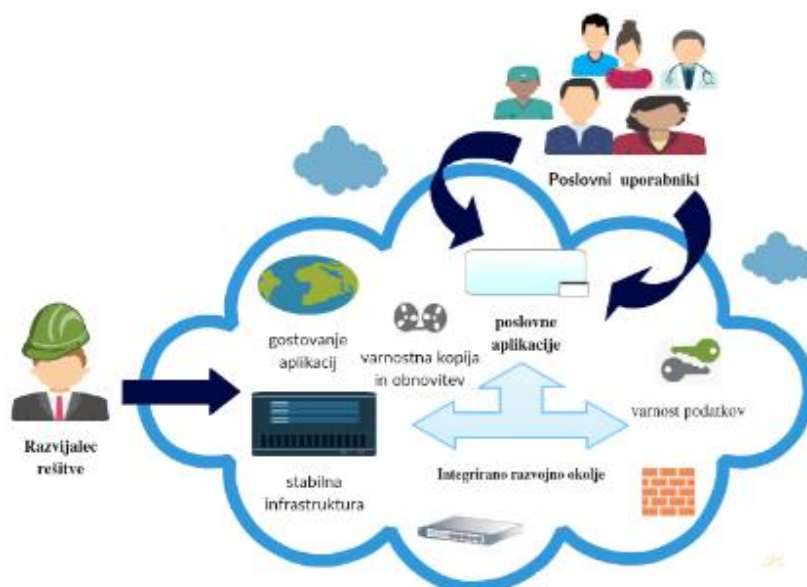
Platforma kot storitev je koncept storitve računalništva v oblaku, kjer sredstev, potrebnih za gradnjo aplikacij, ni treba nameščati in prenašati na lokalne strežnike, ampak so dostopna preko interneta (Sharma & Garg, 2015, str. 232).

Najbolj poznana definicija PaaS po NIST, ki skrbi za razvoj standardov, je, da ponuja zmožnost postavitve različnih tipov aplikacij, ki so razvite z uporabo programskih jezikov in orodij, ponujenih s strani ponudnika. Uporabnik pri tem ne upravlja in ne nadzira spodaj ležeče infrastrukture (omrežja, operacijskega sistema oz. strežnika ter shrambe), temveč ima nadzor nad postavljeno aplikacijo in pogosto tudi nastavitvami gostujočega okolja (Dukarić et al., 2011, str. 100).

PaaS omogoča hitrejši, učinkovitejši in stroškovno dostopnejši model za razvoj aplikacij in njihovo dostavo. Zagotavlja celotno infrastrukturo, potrebno za izvajanje aplikacij preko interneta. Temelji na odmeri oz. po naročniškem modelu, tako da uporabniki plačajo le za tisto, kar uporabljajo. Z modelom PaaS se lahko tako razvijalci kot vodstvo informatike osredotočajo na inovacije, namesto na kompleksnost razvoja aplikacij in infrastrukture (Jaiswar, 2015, str. 11).

Vpliv PaaS na organizacije je viden predvsem, da lahko sredstva, ki bi jih sicer namenili za potrebno infrastrukturo in IT, namenijo za ustvarjanje aplikacij, ki zagotavljajo resnično poslovno vrednost ter konkurenčno prednost organizaciji. Tako lahko razvijalci (Slika 4) po vsem svetu dostopajo do neomejene računalniške moči, lahko gradijo zmogljive aplikacije s preprostim uvajanjem in jih ponudijo uporabnikom ne glede na geografsko lokacijo ter ne nazadnje tako tudi spodbujajo konkurenčnost v okviru razvijalcev in si izmenjujejo znanja ter se nevede spodbujajo k vseživljenjskemu učenju (Pastore, 2013, str. 30–32).

Slika 4: Vloge v oblaki platformi



Vir: B. Ovčjak, G. Jošt, G. Polančič, & M. Heričko, Primerjava učinkovitosti delovanja aplikacij v oblaku in lokalnem okolju, 2011, str. 195.

Prasanth (2012) navaja, da glavna prednost PaaS modela predstavlja, da uporabniku oz. razvijalcu ni treba prenašati programske opreme na lokalni strežnik, zato se te kombinacije preprostosti in stroškovne učinkovitosti poslužuje veliko malih in srednje velikih podjetij ter posameznih razvijalcev, da razvijajo in zgradijo svoje aplikacije brez velikih začetnih investicij. Praktiki in akademski svet so si enotni v mnenju, da je model PaaS pripeljal poslovni svet v obdobje množične inovativnosti in konkurenčnosti v segmentu razvoja aplikacij. V modelu PaaS je mnogo podjetji videlo priložnost zaradi hitrega razvoja aplikacij (angl. *Rapid Application Development – RAD*).

PaaS je primeren tako za poslovne in strokovne razvijalce. Poslovni razvijalci, ki ne pišejo programskih kod, lahko platformo uporabijo kot t.i. Wave Makerji, da razvijejo svoje aplikacije na način s komponentami »povleci in spusti«. Strokovni razvijalci pa največkrat uporabijo platformo GAE za izgradnjo in uporabo spletnih aplikacij. PaaS zagotavlja možnost za podporo testiranja »od začetka do konca« (angl. *End to End Testing*). Najbolj znana ponudnika PaaS sta GAE za Javo in Python, MWA za .NET in Javo ter AWS s svojo pestro ponudbo storitev (Kulkarni et al., 2011, str. 117).

2.2.1 Model aplikacijske platforme kot storitve

Gartner – Application Platform as a Service (2016) aplikacijski model platforme kot storitve (angl. *Application Platform as a Service*, v nadaljevanju aPaaS) opredeljuje kot storitev v oblaku, ki omogoča razvoj in uporabo okolja za aplikacijske storitve. aPaaS ponuja možnost, da razvijemo aplikacijo z že obstoječimi vmesniki, ki so na voljo kot oprema v razvojnem okolju. S pomočjo aPaaS zagotovimo takojšnjo integracijo programske aplikacije z drugimi storitvami. Prednosti uporabe aPaaS je storitev, ki neposredno vpliva na stroške in časovni načrt razvoja poslovnih aplikacij.

Tako dosežemo velike časovne prihranke, prihranke pri programiranju in kodiranju aplikacije ter prihranke pri vzdrževanju aplikacije v primerjavi s tradicionalnimi platformami za razvoj aplikacij (Natis, Pezzini, Iijima, Thomas, & Dunie, 2015).

Forresterjeva analitika Clay Richardson in John Rymer sta leta 2014 skovala izraz »*Low-Code*« in v objavi omenila novo vrsto platforme za razvoj aplikacij. Takšna platforma je namenjena abstraktnim razvijalcem z osnovnim programerskim znanjem, saj zagotavlja že vnaprej vgrajene vmesnike t.i. nizko kodne platforme, uporablja vizualno kompozicijsko okolje, ki omogoča uporabnikom programersko moč, čeprav niso programerji in se brez večjega programiranja lahko lotijo gradnje aplikacije s komponentami, ki uporabljajo preprost jezik za opredelitev značilnosti posamezne komponente aplikacije. Govorimo o uporabi »povleci in spusti«. Tako lahko razvijalec dela po korakih in enostavno določi poslovno logiko na vseh komponentah. Rezultat modela je deklarativni pristop k izgradnji programske opreme oz. aplikacij. Nizko kodne platforme oz. model aPaaS so hit v današnjih storitvah, ki jih ponujajo ponudniki platform v oblaknih storitvah, saj ne

zahtevajo naprednega programerskega znanja in razvijalcu ponujajo razvoj in licenco za sredstva programske opreme do tretjih oseb. Model aPaaS zagotavlja svoj portfelj aplikacij, ki so prilagojene posebnim potrebam vsake stranke (Agilepoint – What is Low Code Application as a Service, 2016).

2.2.2 Model povezovalne platforme kot storitve

Povezovalna ali integracijska platforma kot storitev (angl. *Integration Platform as a Service*, v nadaljevanju iPaaS) predstavlja nabor storitev v oblaku, ki omogočajo razvoj, izvajanje in upravljanje integracijskih tokov v poljubne kombinacije v okolju oblaka za storitve, aplikacije in podatke v posameznih ali več organizacijah, ki uporabljajo isto tehnologijo. Področja povezovanja, ki jih definira Gartner, so povezovanje s pomočjo elektronskega poslovanja B2B (angl. *Business to business* – B2B) za povezavo partnerskih aplikacij, povezovanje oblakov in uporaba dveh arhitekturnih modelov programske opreme, ki naj bi bila dolgoročno usmerjena (Gartner – Integration Platform as a Service, 2016).

2.3 Razlike med tradicionalno platformo in PaaS

Ustvarjanje vrednosti v programskih platformah kot storitev je na področju današnjega načina dobave storitev in načina uporabe storitev okrepila na inovativen način razmišljanja o naravi, ki jo ponuja tovrstna tehnologija. Glavna značilnost storitvenih platform v primerjavi s tradicionalno programsko opremo, ki jo dobimo na trgu, je ravno v oblikovanju razvojnega okolja storitev tehnologije, lociranem na oddaljenem strežniku, ki ga uporabijo razvijalci, da razvijejo funkcionalno aplikacijo za končne uporabnike in jo na isti platformi lahko tudi testirajo ter tako ustvarijo časovni prihranek. V današnjem poslovnem okolju ni dovolj samo najem storitev, ki jih sodobne tehnologije omogočajo, pomembno je predvsem prepoznati dodano vrednost, ki jo z razvojem aplikacije dosežemo, kot je primerjava funkcij s standardno programsko opremo, ki je na voljo na trgih. Da lahko govorimo o ustvarjanju vrednosti, je potrebno vzpostaviti integracijo in analizirati vrednost, ki jo ta prinese. Uporabniška izkušnja je zelo izboljšana z modelom PaaS, saj vsebuje kanale za povratne informacije uporabnikov v vseh fazah projektiranja, razvoja, testiranja, proizvodnje in razvoja rešitve. V sami strategiji za ponudnike platform ni pomembno le, da se osredotočajo na izgradnjo velikosti in številčnosti aplikacij v mreži, temveč tudi za ohranitev privlačnih donosov za razvijalce, da lahko govorimo o win-win učinku ter ustvarjanju vrednosti v programskih platformah kot storitev za razvijalce in uporabnike (Haile & Altmann, 2016, str. 496).

Oblačne platforme imajo tudi nekaj funkcionalnih razlik v primerjavi s tradicionalno razvojno platformo. Najbolj vidna je razlika v več najemniškem razvojnem orodju, ki je pri tradicionalnem razvoju bil na voljo le posamezniku oz. razvijalcu, ki je uporabljal lokalno platformo, medtem ko storitev PaaS podpira več uporabnikov, vsakega z več aktivnimi

projekti. Razlike je mogoče opaziti še v integriranem upravljanju oblačne platforme, ki ga tradicionalno lokalna razvojna platforma ni bila deležna. PaaS uporablja tudi model integriranega obračuna z mehanizmi za zaračunavanje, ki temeljijo na uporabi, podobno kot pri modelu SaaS. Zadnja ključna razlika je v virtualizaciji, saj uporabniki najetih storitev uporabljajo navidezno napravo, na kateri se odvija ločena aplikacija s svojim operacijskim sistemom in programsko opremo (Kulkarni et al., 2011, str. 118).

2.4 Tipi platform

Večina obstoječih PaaS cilja na javni oblak. Aplikacije, ki so na oblačni platformi, se imenujejo na zahtevo, ki temelji na Web ali SaaS rešitvi. PaaS je poslovni model, ki zagotavlja platformo strežnika ali razvojno okolje za programerje. Da bi v celoti izkoristili vse zmogljivosti, ki nam jih ponuja PaaS, bi razvijalci morali prilagoditi svoje vloge, da se lahko odzovejo na spreminjajoče se razmere, ki jih zahtevajo ustrezne IaaS in PaaS storitve preko API klicev znotraj svojih aplikacij. PaaS storitev je še posebno privlačna za nova start-up podjetja, projekte z ambicioznimi roki in organizacije z omejenim proračunom za investicijske odhodke (Kulkarni et al., 2011, str. 120).

Glede storitev, ki nam jih ponujajo platforme, poznamo več tipov PaaS. Platforma za nadgrajevanje že razvitih aplikacij, ki jih uporabljamo za nadgradnjo in spreminjanje že obstoječih aplikacij v povezavi z modelom SaaS. Drugi tip so samostojne platforme, ki so najbolj razširjene in ponujajo razvojno okolje in nimajo povezave na SaaS aplikacije. Tretji tip so platforme za vzpostavitev aplikacij nudijo samo storitev gostovanja aplikacij s poudarkom na zagotavljanju elastičnosti in varnosti na zahtevo in ne ponujajo razvojnega okolja. Zadnji hit med tipi platform so v zadnjem desetletju trendov IT odprte platforme, ki razvijalcu ponujajo prosto izbiro programskih jezikov, operacijskih sistemov in podatkovne baze za izgradnjo aplikacije (Sridevi & Rahinipriyadharshini, 2016, str. 1750; Giessmann & Slabeva, 2012, str. 33–26; Eurich, Giessmann, Mettler, & Slabeva, 2011).

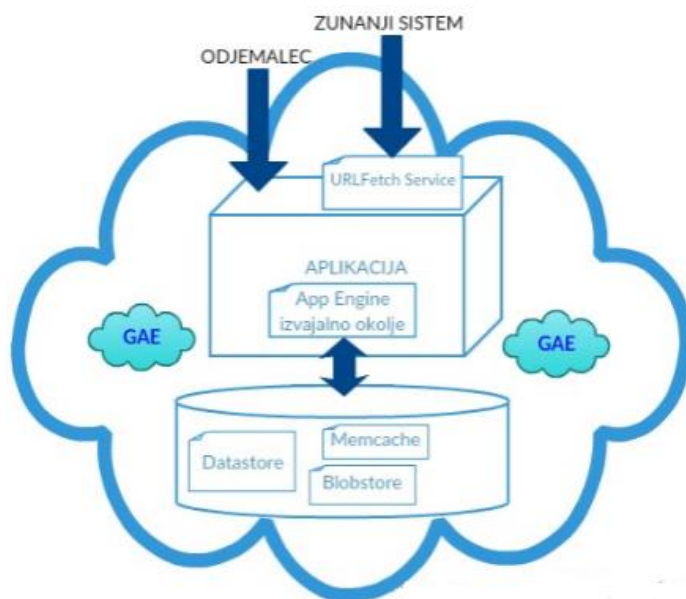
3 ANALIZA PLATFORM

3.1 Google Epp Engine platforma

GAE je t.i. izvajalna (angl. *runtime*) platforma, ki razvijalcem omogoča, da gostujejo in zaganjajo spletne aplikacije ter podatke shranjujejo na infrastrukturi, nastanjeni v številnih Googlovih podatkovnih centrih, ki so razporejeni po vsem svetu. Na platformi razvijalci razvijejo svoje aplikacije na njihovih lokalnih računalnikih pred nalaganjem aplikacije na GAE infrastrukturo. Platforma skrbi za oskrbovanje zahtevane infrastrukture, uvajanje naloženih aplikacij na infrastrukturi in samodejno uravnavanje naloženih aplikacij. Razvijalci v tem primeru ne potrebujejo vzdrževanj in administracije strežnikov, saj to delo prevzame platforma, eden njenih glavnih ciljev pa je, da bi bila enostavna za uporabo za

razvijalce po vsem svetu, brez vlaganja v strojno in programsko opremo. GAE podpira programske jezike, kot so Python, Java, Go ter jezike, ki temeljijo na Java Virtual Machine. Med te jezike sodijo Groovy, Jruby, Scala, Clour in Jython. GAE obračunava svoje storitve po modelu pay-as-you-go, za katerega je značilno, da uporabniki plačajo le vire, ki jih dejansko uporabijo, ter obljublja 99,9-odstotno razpoložljivost. Aplikacije pa tečejo v standardnem okolju, ki je zanesljivo in varno pod imenom Personal Home Page (PHP) (Ovčjak, Jošt, Polančič, & Heričko, 2011, str. 196).

Slika 5: Arhitektura oblačne platforme Google App Engine



Vir: R. Dukarić, R. Povše, & M.B. Jurič, Računalništvo v oblaku, 2011, str. 106.

Za uspešno uporabo GAE platforme se mora uporabnik prijaviti z brezplačnim uporabniškim računom, ustvarjenim pri Googlu. Platforma omogoča spletno konzolo za upravljanje in razporejanje aplikacij, vključno z novo razvitimi. Za razvoj aplikacije je treba namestiti GAE programsko opremo (Software Development Kit, v nadaljevanju SDK) za izbor programskih jezikov na lokalni enoti. SDK simulira GAE okolje vključno z operacijami baz podatkov in komuniciranjem na lokalnem strežniku razvijalca. Naloga SDK je vključevanje orodij za nalaganje aplikacije na GAE, ko je aplikacija pripravljena za uvajanje z ustreznim programskim jezikom (Computer Business Review – AWS vs Google Cloud vs Microsoft Azure: How do they compare and which one is the right for your business, 2016).

Gostovanje in zaganjanje aplikacij na GAE je v določenih kvotah brezplačno. Trenutno je v brezplačno uporabo všteti 1 GB pomnilnika in dovolj pasovne širine za približno 5 milijonov ogledov strani na mesec. Ko uporabnik preseže določene brezplačne kvote, lahko enostavno nadgradi svoj račun in plača dodatne vire, odmerjene na osnovi. Platforma

je namenjena za velike povečave zmogljivosti spletnih aplikacij in je najboljša izbira za povečavo prilagodljivosti oz. elastičnosti že obstoječih aplikacij, saj se povečujejo samodejno, brez da bi razvijalci morali nastavljati mehanizme avtomatskega skaliranja (Hada & Agrawal, 2013, str. 107).

Google je z namenom večjega povezovanja aplikacij, razvitih na GAE, pred časom vstopil v partnerstvo z VMware. Tako se aplikacije, ki so razvite z razvojnimi orodij Google in VMware programsko opremo, lahko prenesejo na GAE zasebni oblak ali drugo infrastrukturo, kot je Amazon EC2 (Hada & Agrawal, 2013, str. 108).

3.1.1 Podatkovna shramba v Google App Engine

GAE platforma podpira tradicionalni sistem podatkovnih baz, ki so ga uvedli pod imenom Google Cloud SQL. Google Cloud SQL je spletna storitev, ki razvijalcem omogoča, da vključijo v svoje na GAE zgrajene aplikacije tudi podatkovno bazo MySQL. Storitev omogoča enostavno premikanje aplikacij in podatkov v in iz GAE oblačne platforme. Pred uvedbo Google Cloud SQL je platforma podpirala samo svoj lastniški sistem podatkovne zbirke pod imenom BigTable, ki predstavlja stolpčno podatkovno bazo. Tehnologija se je pokazala kot uspešna predvsem v uporabi Googlovih storitev, Google Earth in Google Finance (Google Cloud SQL – Google Cloud Platform, 2016).

BigTable storitev ima številne strategije za izvajanje relacijskih podatkovnih baz. Tako kot pri tradicionalnih relacijskih bazah se v BigTable zbirke podatkov izvršujejo na način večdimenzionalnega sortiranja map, vsaka vrednost pa je indeksirana s ključem v vrstici in stolpcu ter časovnim žigom. BigTable se razlikuje tudi v načinu dostopa, saj uporablja vmesnik za shrambo podatkov v obliki API-jev, ki ponujajo funkcijo za ustvarjanje, spreminjanje in brisanje podatkovnih modelov (Severance, 2009, str. 5).

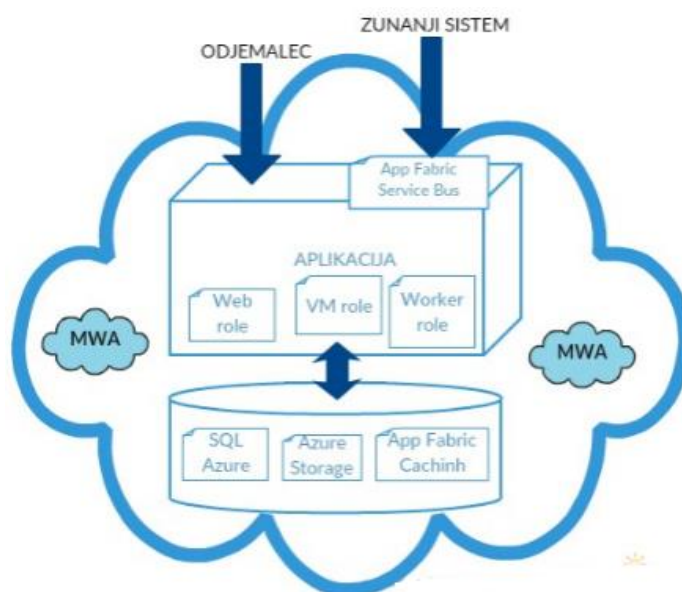
API ponuja funkcije za shranjevanje, spreminjanje in brisanje podatkovnih modelov. Ravno tako omogoča spreminjanje stolpcev metapodatkov, kot je nadzor pravic za dostop. GAE ima dve možnosti za shrambo podatkov: Master in Slave vlogo. Master/Slave uporablja asinhrono replikacijo in je privzeta možnost. Slabost tega je, da lahko v primeru odpovedi ali izpada v podatkovnem centru izgubimo zapis podatkov. Pri drugi možnosti imamo visoko replicirano in bolj odporno shrambo podatkov na okvare in druge izpade iz omrežja, saj se podatki zapisujejo na različne lokacije (Severance, 2009, str. 5–7).

3.2 Microsoft Windows Azure platforma

3.2.1 Windows Azure

MWA platforma se nahaja na številnih zmogljivih Microsoftovih podatkovnih centrih in so za uporabnike dostopni preko spleta. Platformo sestavljajo tri komponente. Operacijski sistem Windows Azure, SQL Azure in Windows Azure AppFabric (Ovčjak et al., 2011, str. 196). Razvijalci lahko gradijo aplikacije, ki uporabljajo .NET okvir programske opreme, ki jo je razvil Microsoft. Aplikacije, zgrajene na MWA, so navadno napisane v programskih jezikih, kot so C#, Visual Basic, C++ in Java z uporabo razvojnega okolja Visual Studio. Razvijalci lahko zgradijo spletne aplikacije, ki uporabljajo tehnologijo ASP.NET, ki je ravno tako plod razvoja podjetja Microsoft in je namenjena za razvoj spletnih strani (Chappell, 2010, str. 4).

Slika 6: Arhitektura oblačne platforme Microsoft Windows Azure



Vir: R. Dukarić, R. Povše, & M.B. Jurič, *Računalništvo v oblaku*, 2011, str. 107.

Windows Azure je operacijski sistem, ki opravlja delo gostovanja, izvajanja aplikacij, abstrakcijo strojne opreme in zagotavljanje vmesnika med uporabniki aplikacij. Aplikacije, gostujoče na Windows Azure, se ne izvajajo vedno na enem strežniku, zaradi česar za dodeljevanje virov ni zadolžen operacijski sistem platforme, temveč to delo opravlja virtualni stroj, ki skrbi za porazdelitev storitev na različne strežnike. Azure deluje na načelu virtualnega trdega diska (angl. *Virtual Hard Disk*), ki je lahko vnaprej določen s strani Microsofta ali s strani tretjih oseb, oz. je lahko uporabniško definiran, s čimer določimo število jeder in velikost pomnilnika (Microsoft Azure, 2016).

Gartnerjevi analitiki Natis in drugi (2015) v svojem poročilu navajajo, da je MWA ena izmed bolj priljubljenih PaaS za razvoj podjetniških spletnih ter mobilnih aplikacij. Microsoft beleži, da je Azure v zadnjem letu več kot podvojila število razvitih aplikacij. Azure platforma podpira in nudi sisteme, ki so najpogosteje uporabljeni ter so namenjeni upravljanju podatkovnih baz, podatkovno analitiko ter storitve za IoT.

Analitska hiša Gartner ocenjuje in priznava Azure kot vodilnega v svojem magičnem kvadratu za Enterprise Application PaaS že tretje leto zapored na podlagi vizije in sposobnosti za izvršitev. Storitve Azure App omogoča enostaven in hiter razvoj spletnih in mobilnih aplikacij, medtem ko Azure Service Fabric zagotavlja arhitekturo, temelječo na mikroservisu za ustvarjanje aplikacij, ki temeljijo na uporabi v oblaku ter so vedno dostopne in skalabilne. Ena zadnjih večjih novosti Azure je Microsoft PowerApps, ki ponuja rešitve za poslovne analitike (Herskowitz, 2016).

3.2.2 SQL Azure

SQL Azure ponuja storitve v oblaku na podlagi relacijske podatkovne baze. Od klasičnega SQL strežnika se razlikuje v tem, da ne potrebujemo fizične povezave na SQL strežnik, ampak se povežemo s simulacijo strežnika. Gre za oblačno storitev, ki zagotavlja značilnosti podatkovne baze SQL strežnika in deluje tako, da ovija (angl. *wrapper*) fizično infrastrukturo (Ovčjak et al., 2011, str. 197).

SQL Azure ponuja tri komponente brez predhodne namestitve oz. vzdrževanja programske opreme. Azure DB nudi sistem za upravljanje baz podatkov. Ta tehnologija omogoča shranjevanje podatkov v Microsoftove strežnike v obliki tabel v Microsoftovih podatkovnih centrih. Tako kot pri drugih storitvah v oblaku, tudi tukaj uporabnik plača le tisto, kar uporablja. Uporaba baz podatkov v oblaku omogoča tudi transformacijo stroškov v poslovne odhodke, kot so naložba iz diskov v sisteme za upravljanje baz podatkov. Uporabniki lahko tako za delo s podatki v oblaku uporabijo tudi lokalno programsko opremo, kot je SQL Server Reporting Service. Druga komponenta je izdelava poročil v storitvi SQL Azure Reporting. Uporablja se skupaj s SQL Azure podatkovno bazo in na podlagi informacij oz. podatkov v okolju oblaka omogoča, da uporabnik ustvari in objavlja poročila. Tretja komponenta SQL Azurja je sinhronizacija podatkov med podatkovnimi bazami v oblaku in tistimi na lokalnem strežniku, ki jih nudi SQL Azure Data Synce. S tem modelom prihranimo veliko časa, saj nam ni treba pisati programa, ampak moramo le nastaviti določene nastavitve in sinhronizacija se izvede sama (Chappell, 2010, str. 13–16).

3.2.3 Windows Azure AppFabric

Azure AppFabric zagotavlja infrastrukturo za aplikacije. Razvijalci, ki ustvarjajo aplikacije, lahko uporabijo različne vrste infrastrukture. Ravno tako kot SQL Azure, tudi

AppFabric ponuja tri komponente. Prva je storitveno vodilo v MWA platformi (angl. *Service Bus*) ter nudi izpostavljanje storitev aplikacije do drugih aplikacij, vzpostavljenih v oblaku. Druga, nadzor dostopa (angl. *Access Control*), omogoča definiranje pravil za nadzor dostopa vsakega uporabnika preko centralne nadzorne plošče na platformi. Nadzori omogočajo razvijalcu aplikacije podporo za implementacijo mehanizmov (Chappell, 2010, str. 21). Je knjižnica storitev, ki pomagajo pri poganjanju storitev v oblaku ter, kot že omenjeno, povezovanju z zunanjimi programskimi okolji. Predstavlja varen prehod med obstoječimi aplikacijami in storitvami v oblaku. Prednost te tehnologije je, da lahko povezujemo aplikacije brez poznavanja povezovalne tehnologije. Zadnja komponenta pa služi pospeševanju dostopa do istih podatkov z načinom t.i. predpomnjenja (angl. *Caching*). S tem zmanjšamo število dostopov med aplikacijo in podatkovno bazo, ki jo uporabljamo (Ovčjak et al., 2011, str. 198; Chappell, 2010, str. 21).

3.2.4 Windows Azure Marketplace

S storitvijo Azure tržnice (angl. *marketplace*) razvijalcu omogoča dostop do storitev iskanja in nakupa aplikacij ter podatkov v okolju oblaka. Storitve tržnice delimo na dva dela oz. komponenti (Chappell, 2010, str. 17). Prva, storitev tržnice aplikacij, razvijalcem omogoča, da po izgradnji in uvedbi aplikacije izpostavijo in ponudijo rešitev potencialnim uporabnikom, bodisi tretjim osebam bodisi znotraj organizacije. Druga komponenta, tržnica podatkov (angl. *DataMarket*), ponuja storitev za iskanje in nakup podatkov za aplikacije in njihove uporabnike. Storitve nam ponudi podatke, ki so na voljo, uporabnik pa se odloči, katere bo zakupil. Dostop do aplikacije pa lahko izvedemo na dva načina: preko protokolov REST (angl. *Representational State Transfer*, v nadaljevanju REST) ali OData (angl. *Open Data Protocol* – Odata), ki sta poznana po tem, da izvajata spletno poizvedovanje in posodabljanje podatkov. Storitve deluje v povezavi s SQL Azure DB, saj so podatki največkrat shranjeni na sami platformi ali pa so v podatkovnih centrih zunaj okolja oblaka (Ovčjak et al., 2011, str. 198; Chappell, 2010, str. 21).

V shrambi podatkov je od leta 2012 razvijalcem na voljo tudi široka paleta storitev dodatkov (angl. *add-on*) v trgovini Microsoft Azure, ki nudi razvijalcem aplikacijske storitve tretjih oseb in podatkovne podatke, ki izboljšujejo in omogočajo prilagoditvena okolja storitev MWA (Microsoft Azure Marketplace, 2016).

3.2.5 Varnost Microsoft Windows Azure platforme

MWA je zasnovan po protokolu, da zmanjša odgovornost upravljanja infrastrukture in omogoča razvijalcem, da se osredotočijo na sam razvoj rešitve. Microsoft uporablja visoke standarde ameriške civilne obveščevalne službe (angl. *Central Intelligence Agency* – CIA) in jih poskuša zagotavljati v največji meri z varnostnimi mehanizmi, ki se izvajajo na MWA platformi. Zaupnost podatkov je opremljena z mehanizmom za varno preverjanje

prisotnosti na portalu in v edinstvenih naključno generiranih ključih za shranjevanje računov. Podatki se nahajajo na več lokacijah Microsoftovih strežnikov, s čimer se zavarujejo v primeru, da je eden od strežnikov ogrožen. Razvijalec se lahko odloči za šifriranje svojih podatkov z uporabo močnih šifrirnih algoritmov. Glavni mehanizem za ohranitev celovitosti podatkov pa predstavlja način repliciranja podatkov na treh virtualnih diskih. Podatkov, ki so shranjeni na dveh od treh virtualnih diskov, ni mogoče izbrisati ali prepisati, medtem ko so podatki na primarnem disku shranjeni in zavarovani preko konfiguracije datotek, ki jih razvijalec naloži pri uvajanju uporabe. Na ravni PaaS in IaaS MWA samodejno odpravlja več varnostnih groženj, brez poseganja razvijalca z varnostjo požarnega zida, ki blokira vsa komunikacijska vrata, ki niso posebej odprta in omogočena s strani virtualnega stroja (Microsoft Trust Center, 2016).

3.3 Amazon Web Services platforma

Amazonove spletne storitve zagotavljajo prilagodljivo in stroškovno učinkovito ter uporabniku prijazno oblako platformo, saj se prilagajajo vsem njihovim zahtevam (Hada & Agrawal, 2013, str. 107). Amazon je nastopil na trgu leta 2006 z nizom ponujanja računalniških virov ter infrastrukture kot storitve. Uporabnikom omogoča, da imajo sredstva IT v najemu proti ustreznemu plačilu (Sharma, Sah, & Pani, 2015, str. 20). Ponujajo več različnih storitev znotraj portfelja, pogledali pa si bomo samo nekatere, ki so najpogosteje uporabljene (Amazon Web Services, 2016).

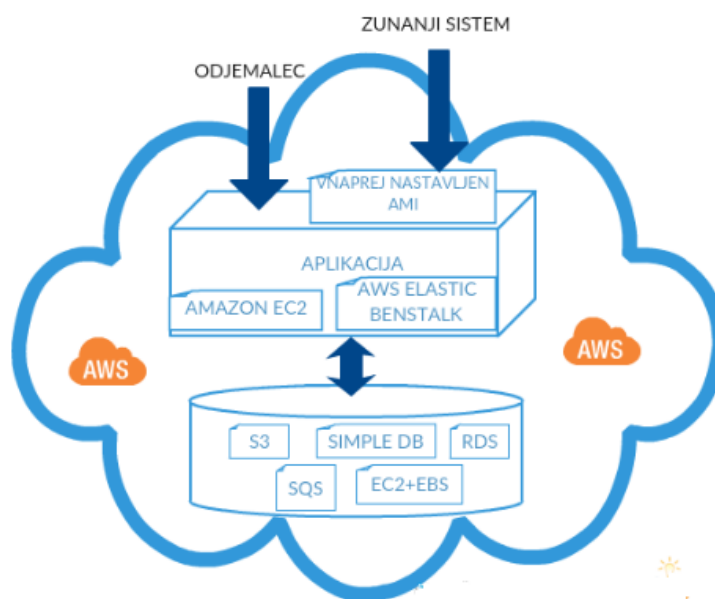
3.3.1 Amazon Elastic Compute Cloud

Amazon Elastic Compute Cloud (v nadaljevnaju Amazon EC2) je storitev infrastrukture, IaaS na zahtevo za upravljanje virtualnih strojev. Nastavimo lahko virtualno IT infrastrukturo v oblaku s kreiranjem virtualnih naprav, na katerih se izvajajo razvijalčeve aplikacije. Ta storitev deluje s pomočjo Amazonovih Machine Images (v nadaljevnaju AMI) strojev, ki predstavljajo programski sklop operacijskih sistemov z dodatnimi programskimi paketi. S pomočjo te storitve lahko upravljamo celoten življenjski cikel AMI-ja. AMI vsebuje operacijski sistem, ki ga dobimo v paketu z dodano programsko opremo. Uporabniki lahko uporabijo vse javno dostopne AMI-je iz javnih zbirk. AMI-ji so shranjeni v skladišču S3 in se lahko uporabijo v EC2 storitvi. Ta možnost ponuja nov distribucijski kanal za prodajalce programske opreme, ki lahko v celoti izkoristijo možnost dobave tega modela aplikacij za avtomatsko izravnavanje obremenitev in samodejno luščenje. Primer je združitev IBMa z Amazonom, s čimer zagotavljajo večje možnosti, kjer lahko razvijalci izberejo poljubno razpoložljivost AMI-jev in jih preizkusijo ali jih vključijo v njihovo virtualno infrastrukturo (Amazon Elastic Compute Cloud – Amazon Machine Images, 2016).

Amazon EC2 deluje na načelu visokozmogljivega razširjanja oz. skalabilnosti aplikacij pod komandnimi ukazi razvijalca. Če AMI zazna, da navidezna naprava deluje

preobremenjeno iz vidika procesorske moči, ustvari instanco, na katero se prevali določen odstotek preobremenjenosti (Amazon Elastic Compute Cloud – Amazon Machine Images, 2016; Sharma et al., 2015, str. 26). Tako kot v nekaterih drugih storitvah AWS platforme tudi EC2 storitev omogoča razvijalcu, da posamezne virtualne naprave postavi na različne možne geografske regije Amazona. Regije so zemljepisno razpršene lokacije, ustvarjene z namenom, da če pride do odpovedi na eni lokaciji, aplikacija še vedno deluje s pomočjo druge lokacije (Yehuda-Ben, Yehuda-Ben, Schuster, & Tsafirir, 2013, str. 2). Uporablja pa se največkrat skupaj z Amazon Elastic Beanstalk storitvijo (Slika 7).

Slika 7: Arhitektura oblačne platforme Amazon Web Services



Vir: R. Dukarić, R. Povše, & M.B. Jurič, Računalništvo v oblaku, 2011, str. 105.

3.3.2 Amazon Elastic Beanstalk

Storitev Amazon Elastic Beanstalk (v nadaljevanju Amazon EBS), je na voljo razvijalcem za pomoč pri hitri postavitvi in upravljanju aplikacij brez nadzora in oskrbovanja infrastrukture v zaledju aplikacije. Poskrbi za postavitvene podrobnosti, potrebne kapacitete, izenačevanje in prevelitev obremenitve, avtomatizirano skaliranje in varno uporabo aplikacije za uporabnike. Storitev EBS je sestavljena iz številnih komponent, ki so potrebne za gladko delovanje aplikacije, kot so aplikacijska komponenta, komponenta okolja, komponenta konfiguracije in ostale potrebne komponente. Uporablja se v povezavi z ostalimi storitvami AWS, predvsem S3 in CloudWatch (Amazon Elastic Beanstalk – What is AWS Elastic Beanstalk, 2016).

3.3.3 Amazon Simple Storage Service

Še nedolgo nazaj smo za lokalno večjo kapaciteto uporabljali naprave, kot so trdi zunanji diski, USB pogone ali dodatne notranje pomnilnike oz. strežniške rezine (angl. *Network Attached Storage*), ki so priključeni na računalniško omrežje, za katere ni bilo treba plačati dodatne uporabe za storitev skladiščenja, razen izdatka za investicijo. Pomanjkljivost tovrstno shranjenih podatkov na lokalni ravni je, da niso dostopni izza požarnega zidu (angl. *Firewall*) (Sharma et al., 2015, str. 20).

Z novim Amazonovim modelom ponujanja storitev se je pokazalo dejstvo, da lahko dostopamo do svojih podatkov kjer koli ter shranjujemo skoraj neomejene količine podatkov, ki jih izberemo kot storitev začasne hrambe ali jih izkoristimo na druge načine. Značilnosti storitve Amazon Simple Storage Service (v nadaljevanju Amazon S3) je, da razvijalcu omogoča pisanje, branje in brisanje predmetov iz velikosti 1 bajta do 5 gigabajtov za potrebe njegovih spletnih aplikacij, ki velikokrat potrebujejo nadgradnjo oz. so aplikacije z visoko skalabilnostjo (Han, 2015, str. 263–265).

Zagotavlja mehanizme za preverjanje pristnosti za zagotovitev, da so podatki shranjeni varno pred nepooblaščenim dostopom. Omogoča določanje pravic, da se izvede javno ali zasebno dostopanje za uporabnike ter uporablja standarde, ki temeljijo na REST in SOAP vmesnikih (Amazon Simple Storage Service – What is Amazon S3, 2016).

Ustanovitev računa Amazon S3 je preprost in enostaven proces. Vse, kar potrebujemo, je veljavna kreditna oz. debetna kartica, tako da celoten postopek traja le nekaj minut (Amazon Simple Storage Service – What is Amazon S3, 2016). Butler (2016) pravi, da je osnovna varnost zagotovljena na dva načina, ki jih je mogoče najti s pomočjo nadzorne plošče za upravljanje računa. Vsak lastnik računa ima svoj enoličen Access Key ID, ki identificira račun AWS. Druga opcija varnosti je vklop varnostnega ključa (angl. *Secret Access Key*), ki ga je treba vnesti na zahtevo. Ključ lahko regeneriramo kadar koli in ga je treba hraniti v tajnosti.

Upravljanje podatkov računa AWS zagotavlja preko API-jev, ki temeljijo na REST protokolu in izpostavljajo funkcionalnost AWS preko protokola HTTP (angl. *HyperText Transfer Protocol* – HTTP). Obstaja pa tudi dodatek (angl. *Add-on*), ki komercialno omogoča svobodo strank za upravljanje podatkov v S3. Amazon omenja, da je eden izmed najbolj uporabljenih dodatkov S3Fox Firefox add-on. Amazon S3 storitev lahko služi kot popolno orodje za varnostno kopiranje v malih, srednjih ter velikih podjetjih. Poleg tega je funkcionalnost S3 računa tudi, da ga lahko uporabljamo z drugimi storitvami AWS, predvsem z EC2 storitvijo (Amazon Simple Storage Service – What is Amazon S3, 2016).

3.3.4 Amazon Simple Queue Service

Amazon Simple Queue Service (v nadaljevanju Amazon SQS) ponuja zanesljive in prilagodljive storitve za shranjevanje sporočil, ki potujejo med računalniki. Funkcionalnost SQS je, da lahko podatke premikamo med komponentami, porazdeljenimi v aplikaciji, ki opravljajo različne naloge, ne da bi med samim potekom izgubili podatek. Razvijalec lahko na več geografskih območjih ustvari neomejeno kapaciteto številnih sporočil vrste Amazon SQS (Amazon Simple Queue Service, 2016).

Deluje na načelu porazdelitvenega sistema čakalne vrste, da omogoči aplikacijam spletnih storitev hitro, zanesljivo in nemoteno sporočanje, in je hierarhično porazdeljena funkcionalna storitev na zahtevo. Za lažjo predstavo si lahko čakalno vrsto predstavljamo kot skladišče podatkov, ki čakajo na obdelavo. Uporabniki lahko sporočila berejo in pošiljajo istočasno, v vrsti za obdelavo pa jih lahko zadržujemo največ dva tedna (Sharma et al., 2015, str. 23; Murty, 2008, str. 56).

Čakalna vrsta deluje kot blažilec med sestavinami za ustvarjanje in shranjevanje podatkov. To pomeni, da čakalna vrsta rešuje vsa vprašanja, ki se pojavijo, če uporabnik ustvarja podatke hitreje, kot jih lahko odjemalec obdeluje. Sporočila, ki jih shranimo, lahko vsebujejo do 256 KB besedila v kakršni koli obliki. Vsako komponento shranimo s pomočjo API-ja Amazona SQS. Sporočila, ki so večja od razpoložljivega prostora, pa je mogoče upravljati s pomočjo razširljivosti Client Library Java, ki ga uporablja Amazon S3 za shranjevanje podatkov večjih velikosti (Amazon Simple Queue Service, 2016).

Glavne lastnosti Amazon SQS so, da pošilja sporočila in omogoča sočasni dostop in visoko razpoložljivost za priklic in pošiljanje. Mehanizem deluje na načelu več piscev in bralcev (angl. *Multiple writers and readers*), kar pomeni, da lahko hkrati pišemo in upravljamo več sporočil, medtem pa jih Amazon SQS zaklene in istočasno obdeluje drugo sporočilo. Z nadzorom do dostopa pa lahko spremljamo in nadziramo, kdo pošilja sporočila v čakalnico in kdo vse lahko sprejema sporočila iz čakalne vrste (Murty, 2008, str. 295).

3.3.5 Amazon SimpleDB

Amazon Simple DB je prilagodljiva, visoko razpoložljiva storitev za shranjevanje podatkov v strukturi nerelacijske podatkovne baze na AWS platformi (Sharma et al., 2015, str. 26). Vmesnik zagotavlja spletne storitve za ustvarjanje in shranjevanje več naborov podatkov, podatkovnih poizvedb, vrnjeni rezultat in podatke pa sistem samodejno indeksira. Tako ne potrebuje administratorja in kar sama poskrbi za celotno vzdrževanje infrastrukture (Murty, 2008, str. 497).

Amazon Simple DB storitev ponuja v svojem naboru relativno majhne količine shranjevanja podatkov do 1024 KB. Za večje količine shranjevanja podatkov je potrebno skladiščiti na storitvi Amazon S3. Prednost shranjenih podatkov na storitvi Simple DB je učinkovitost zaradi hitrega dostopa preko indeksiranja, podatki v tej bazi so shranjeni v domenah, vsaka domena pa omogoča do deset GB prostora za shranjevanje. Ker je Simple DB nerelacijska podatkovna baza in ne podpira Select operacije, priključuje attribute na podlagi kriterijev, ki jih določi razvijalec s poizvedbo (Amazon Simple DB, 2016).

3.3.6 Amazon Relational Database Service

Amazonova relacijska podatkovna baza (angl. *Amazon Relational Database Services*, v nadaljevanju RDS) je spletna storitev, ki je enostavna za upravljanje, nastavljanje in upravljanje obsega relacijske zbirke podatkov v oblaku in prevzame vse naloge administratorja. RDS ima funkcijo, da ščiti uporabnika pri kompleksnih nalogah v podatkovni zbirki. Vrh strežniške RDS baze podatkov dodaja tudi pomoč procesorja, pomnilnika in skladiščenje ter upravlja varnostne kopije. Razvijalec ustvari podatkovno bazo preko dveh možnih načinov: z vmesnikom API ali pa preko glavne administracijske nadzorne plošče RDS. Samodejno zaznava napake in pomaga pri popravilju odkritih napak. Amazon RDS je tudi ena izmed najbolj celovitih spletnih storitev, ki glede na svojo konkurenco (Azure SQL Database in Google Cloud SQL) ponuja najbolj preprosto uporabo in različne priljubljene podatkovne strežnike, kot so MySQL, PostgreSQL, ORACLE in Microsoft SQL server (Amazon Relation DataBase Service, 2016; Sharma et al., 2015, str. 26).

3.3.7 Amazon Elastic MapReduce

Amazon Elastic MapReduce je spletna storitev, ki omogoča preprosto, hitro in stroškovno učinkovito obdelavo velikih količin podatkov. Za implementacijo uporablja Googlovo orodje MapReduce, ki ga poznamo pod imenom Hadoop in ki služi temu, da razvijalec razvije rešitev za opravljanje intenzivnih informacijskih opravil oz. služi za podatkovno rudarjenje, indeksiranje in druga poslovno inteligenčna opravila. Kot vir podatkovne baze podatkov EMR za namen analize se uporablja storitev podatkovne baze S3, kamor se tudi shranjujejo analize in analizirani podatki (Amazon EMR, 2016).

3.3.8 Amazon CloudFront in CloudWatch

Amazon CloudFront je globalna mreža storitev za dostavo vsebin, ki pospešujejo dostavo do spletnih strani, API-jev, video vsebin in drugih spletnih sredstev. Združuje se z drugimi Amazonovimi storitvami, ki omogočajo podjetjem in razvijalcem preprost način za pospešeno dostavo vsebin končnim uporabnikom (Amazon CloudFront – Content Delivery Network, 2016).

Amazon CloudWatch je spletna storitev za spremljanje oblačnih virov AWS in vseh storitev, ki jih ponuja in delujejo znotraj platforme. Storitve lahko uporabimo za zbiranje, sledenje in meritve vseh Amazonovih storitev znotraj AWS. Funkcije, ki jih ponuja, so nastavljanje alarmov ter samodejno odzivanje na spremembe znotraj virov v primeru potreb povečanja zmogljivosti. Največkrat s storitvijo spremljamo Amazon EC2, Amazon RDS, in Amazon SimpleDB – torej vire v podatkovnih storitvah. S pomočjo podatkov se lahko odzovemo hitreje in zagotovimo aplikacijam nemoteno delovanje (Amazon CloudWatch, 2016).

4 PRIMERJAVA ANALIZIRANIH PLATFORM

4.1 Google App Engine

Leta 2008 je Google predstavil GAE platformo, ki služi kot razvojno okolje razvijalcem, da ustvarijo svoje aplikacije. Podjetje je pridobilo velike svetovne korporacije, kot so General Mills, Best Buy in Coca-Cola, kar govori o njihovi dobri poslovni integriteti. Google je s svojim modelom in filozofijo odprtokodne programske opreme (angl. *Open-Source*) načel dvorezni meč med razvijalci in izgubil nekaj zaupanja s strani, ki so argumentirano proti odprtokodnemu sistemu. Google je tako postal resen tekmelec v ponudbi računalništva v oblaku med Amazonom ter Microsoftom, saj tekmuje tako s ceno kot s tehnologijo (Priloga 2 in 3) (Weinberger, 2015).

Ključno za Googlovo platformo je, da obstaja samo ena metoda zaračunavanja storitev v zakupu večjih količin podatkov, ki jo plačamo mesečno in ne ponuja količinskih popustov. Po mnenju uporabnikov je GAE ena najbolj privlačnih platform ter ima najboljšo cenovno politiko (Priloga 3) med vsemi ponudniki storitev v oblaku pri shranjevanju v različnih stopnjah shranjevanja. Azure ima zelo podobne cene kot AWS in cenovno sta tesna tekmeča med ponudniki storitev v oblaku. GAE je platforma, ki ima za svoj poslovni model gostovanje spletnih aplikacij z možnostjo velike elastičnosti. Lahko trdimo, da je stopnja abstrakcije na tej platformi visoka zaradi možnosti velike razširljivosti. Z uvedbo podpore MySQL podatkovnih baz so omogočili lažji prehod na oblak, kar je predhodna rešitev Bigtable onemogočala ter podpirala lasten sistem podatkovne baze, zato MySQL med razvijalci postaja vse bolj priljubljena (Aldiab, 2015).

4.2 Microsoft Windows Azure

Microsoft in ostala podjetja s programsko opremo so oblak najprej obravnavala kot novo modno muho oz. trend, kasneje pa kot konkurenčno grožnjo. Microsoft se ponaša kot eden od vodilnih v ponudbi programske opreme, kot sta Windows Server in SQL Server, ki prinašajo več milijard dolarjev ter se uporabljajo v podatkovnih centrih večine podjetij. Računalništvo v oblaku za Microsoft predstavlja največji dejavnik za spodbujanje rasti

njihovega poslovanja in rasti prihodkov. Azure je optimiziran za aplikacije, ki so zgrajene v .NET-u, zato je za veliko podjetij enostavneje, da migrirajo aplikacije, ki so zgrajene na .NET programskem jeziku, v Azure oblak (Finnegan, 2016, Ovčjak et al., 2011, str. 205).

Veliko uporabnikov, ki poslujejo z Microsoftom, predvsem podjetja, poslujejo s pogodbo, ki jim daje popust na njihovo programsko opremo (angl. *Enterprise Agreement*). Microsoft opredeljuje, da njegova ključna prednost ni samo v tehnologiji, ki jo ponuja, temveč tudi njihov know-how oz. strokovno znanje in izkušnje, ki so jih pridobili s poslovanjem v svoji bogati karieri (Greene, 2013). MWA platforma predstavlja različne možnosti uporabe z različno obravnavo potreb. Tako Microsoft Azure nudi okolje za shranjevanje podatkov v oblaku, medtem ko SQL Azure nudi podatkovne baze v oblaku skupaj z možnostjo poročanja in sinhronizacije podatkov. Tretja storitev AppFabric ponuja infrastrukturo v oblaku, ki podpira tako okolje samega oblaka, kot aplikacijo na kraju samem. Z zadnjo storitvijo, Azure Marketplace, pa imamo možnost spletne trgovine za iskanje in nakup podatkovnih nizov od ponudnikov vsebin za aplikacije v oblaku. MWA zaračunava storitve po modelu na zahtevo ter ponuja tudi kratkoročne obveznosti do storitev s popustom (Aldiab, 2015). Po mnenju Gartnerja Microsoft Azure po prihodkih raste hitreje kot AWS. Prihodki Microsoft Azure znašajo 8,2 milijarde dolarjev v letu 2015 (Weinberger, 2015).

4.3 Amazon Web Service

Raziskava analitske hiše Gartner in skupnosti tehnologov, svetovalcev in praktikov Wikibone je pokazala, da Amazon kaže prevlado v ponudbi storitve IaaS. AWS je leta 2006 začel le z nizom osnovnih infrastrukturnih storitev na virtualnih strežnikih. Sprva je bil AWS na voljo manjšim razvijalcem kot poceni alternativa za preizkušanje delovanja preprostih spletnih strani. Gartner ocenjuje, da bo AWS s svojim uspešnim poslovnim modelom do leta 2020 ustvaril do okoli 50 milijard dolarjev poslov, saj ponuja toliko računalniške zmogljivosti, kot jo redkokateri ponudnik te storitve na trgu (Gartner IoT – Says IoT Adoption Is Driving the Use of Platform as a Service, 2016; Weinberger, 2015).

AWS predstavlja najbolj pestro ponudbo storitev v oblaku v primerjavi s konkurenčnimi ponudniki. Storitve je mogoče uporabljati integrirano ali posamično za zagotavljanje računalniških virov, zato razvijalci po njej posegajo vse pogosteje. Ponuja več možnosti politik zaračunavanja storitev – npr. imajo tudi možnost zaračunavanja po urah. Minimalna uporaba je ena ura in obstaja možnost zakupa enega od treh modelov. Prvi je na zahtevo, kjer kupec plača vire, ki jih je porabil; drugi je model pridržanja, kjer so pridržane stopnje v primeru, da se zavežemo za njihove storitve AWS platforme za leto, dve ali več; in model plačila na kraju samem, kjer kupec zakupi storitve, ko jih dejansko potrebuje (Finnegan, 2016; Greene, 2013).

Posebnost AWS platforme je tudi storitev sistema domenskih imen (Priloga 2), ki razvijalcu oz. organizacijam omogoča zanesljiv in stroškovno učinkovit način končnih poti uporabnikov do internetnih aplikacij s prevajanjem domene v številčne IP-naslove, ki jih uporabljajo strežniki za medsebojno povezavo (Amazon Route 53, 2016).

SKLEP

V prvem delu zaključne naloge smo predstavili koncept računalništva v oblaku, poslovni pomen računalništva v oblaku ter ekonomski vidik, ki ga je pametno preučiti še preden se odločimo za investiranje v oblačne storitve. Predstavili smo tri glavne storitvene modele računalništva v oblaku ter okolja vzpostavitve, v katerih lahko storitve uporabljamo za namen poslovanja. Na raziskovalni vprašanji, ki smo si jih zastavili na samem začetku strokovne naloge, lahko odgovorimo, če skrbno preučimo ter pretehtamo vse vidike investicij računalništva v oblaku. Ta nam dejansko znižuje stroške informatike v podjetju, poveča se učinkovitost poslovanja, ker se osredotočamo na primarno dejavnost organizacije in naše poslovne aplikacije delujejo nemoteno zaradi velike razpoložljivosti in hitre vzpostavitve. PaaS s svojim poslovnim modelom omogoča razvijalcu nezamudno in ne dolgotrajno uvajanje rešitev, čemur smo bili priča v preteklih letih, pred pojavom platforme v oblaku z razvijanjem rešitev na lokalni platformi. Z modelom PaaS lahko do svojih podatkov in spletnih aplikacij dostopamo od kjer koli in kadar koli zaradi odprave geografskih meja. Edina zahteva za gostitev aplikacije v oblaku je dostop do internetnega omrežja.

Model PaaS predstavlja veliko prednosti za razvijalce v primerjavi z razvojem na lokalni ravni, saj so stroški plačila storitev skoraj nični v primerjavi z investicijo v celotno infrastrukturo IT, ki jo potrebujemo na lokalni ravni. Tako praktiki kot akademska sfera so si enotni, da bo v prihodnje vse več razvoja aplikacij potekalo na storitvah v oblaku z velikim poudarkom na IoT, ki predstavlja v današnji informacijski dobi veliko tehnološko revolucijo. Z gotovostjo lahko trdimo, da bo glavni model za razvoj vseh vrst aplikacij potekal v hitrem načinu razvoja aplikacij.

V zadnjem delu strokovne naloge smo predstavili primerjalno analizo treh največjih ponudnikov storitev platforme v oblaku: GAE, MWA in AWS. Vodilni ponudniki storitev v oblaku so zgradili velike količine računalniške infrastrukture za vzpostavitev storitev v oblaku. Podatkovni centri teh podjetij so zmogljivejši, večji in bolj učinkoviti kot kateri koli prostor s strežniki, ki bi ga podjetje gradilo in vodilo za svoje potrebe. Ponudniki tako podjetjem in razvijalcem ponujajo, da samo ustvarijo račun z veljavno kreditno kartico in že imajo dostop do velike računalniške moči preko spleta. Predvsem za podjetja to pomeni, da lahko programska oprema deluje v okolju visoke zmogljivosti in razpoložljivosti za veliko manj stroškov, ker niso deležni nikakršnega vzdrževanja podatkovnega centra, saj za to poskrbijo ponudniki storitev v oblaku. Omenimo lahko tudi ekonomije obsega, ki so nastale z računalništvom v oblaku, saj je vsota prihodkov veliko višja od stroškov vpliva

na okolje. Z modelom računalništva v oblaku so tako organizacije kot tudi posamezniki postali bolj družbeno odgovorni in ohranjajo konkurenčnost v panogi.

LITERATURA IN VIRI

1. Aggarwal, S., & McCabe, L. (2009). The Compelling TCO Case for Cloud Computing in SMB and Mid-Market Enterprises. *Hurwitz & Associates*. Najdeno 3. julija 2016 na spletnem naslovu <http://www.netsuite.com/portal/pdf/wp-hurwitztco-study-dynamics.pdf>
2. *Agilepoint – What is Low Code Application Platform as a Service (aPaaS)?* Najdeno 26. junija 2016 na spletnem naslovu <http://agilepoint.com/what-is-low-code-application-platform-as-a-service-apaas/>
3. Aldiab, M. (2015, 4. marec). Public Cloud War: AWS vs Azure vs Google. *Cloud Academy Blog*. Najdeno 4. avgusta 2016 na spletnem naslovu <http://cloudacademy.com/blog/public-cloud-war-aws-vs-azure-vs-google/>
4. *Amazon CloudFront – Content Delivery Network*. Najdeno 4. avgusta 2016 na spletnem naslovu <https://aws.amazon.com/cloudfront/>
5. *Amazon CloudWatch*. Najdeno 4. avgust 2016 na spletnem naslovu <https://aws.amazon.com/cloudwatch/>
6. *Amazon Elastic Beanstalk – What Is AWS Elastic Beanstalk*. Najdeno 4. avgusta 2016 na spletnem naslovu <http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/Welcome.html>
7. *Amazon Elastic Compute Cloud – Amazon Machine Images*. Najdeno 2. avgusta 2016 na spletnem naslovu <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html>
8. *Amazon EMR*. Najdeno 3. avgusta 2016 na spletnem naslovu <https://aws.amazon.com/emr/>
9. *Amazon Relation Database Service*. Najdeno 3. avgust 2016 na spletnem naslovu <https://aws.amazon.com/rds/>
10. *Amazon Route 53*. Najdeno 3. avgusta 2016 na spletnem naslovu <https://aws.amazon.com/route53/>
11. *Amazon Simple Queue Service*. Najdeno 3. avgusta 2016 na spletnem naslovu <https://aws.amazon.com/sqs/>
12. *Amazon Simple Storage Service – What is Amazon S3?* Najdeno 2. avgusta 2016 na spletnem naslovu <http://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html>
13. *Amazon SimpleDB*. Najdeno 3. avgusta 2016 na spletnem naslovu <https://aws.amazon.com/simpledb/>
14. *Amazon Web Services*. Najdeno 3. avgusta 2016 na spletnem naslovu <https://aws.amazon.com/>
15. *Apprenda – Future of Cloud Computing*. Najdeno 21. maja 2016 na spletnem naslovu <https://apprenda.com/library/cloud/future-of-cloud-computing/>
16. Bhadauria, R., Chaki, R., Chaki, N., & Sanyal, S. (2014). Security Issues In Cloud Computing. *Acta Technica Corviniensis - Bulletin of Engineering*, 7(4), 159–177.

17. Bojanc, R., & Šušmak, B. (2013). Logical – platforme računalništva v oblaku in orodja za logistične centre in skupnosti. *Uporabna informatika*, 21(1), 43–48.
18. Butler, B. (2016, 27. januar). 5 Amazon Web Services security tips for businesses 2016: How to secure Amazon Web Services: AWS security tips. *Network World US*. Najdeno 25. julij 2016 na spletnem naslovu <http://www.computerworlduk.com/galleries/security/5-amazon-web-services-security-tips-businesses-should-know-3634104/v>
19. Chakravarthy, H., Kannan, M., & Kannan E. (2014). A Review on secured Cloud computing environment. *American Journal of Applied Sciences*, 11(8), 1224–1228.
20. Chappell, D. (2010). Introducing The Windows Azure Platform. *DavidChappell Associates*. Najdeno 20. julija 2016 na spletnem naslovu http://www.davidchappell.com/writing/white_papers/Introducing_the_Windows_Azure_Platform,_v1.4--Chappell.pdf
21. *Cloud360.com – The Top 20 Platform as a Service Vendors*. Najdeno 31. julija 2016 na spletnem naslovu <http://www.clouds360.com/paas.php>
22. *Computer Business Review - AWS vs Google Cloud vs Microsoft Azure: How do they compare and which one is right for your business?* Najdeno 3. junija 2016 na spletnem naslovu <http://www.cbronline.com/news/cloud/public/aws-vs-google-cloud-vs-microsoft-azure-how-do-they-compare-and-which-one-is-right-for-your-business-4802310>
23. Delak, B., & Terčelj, M. (2011). Računalništvo v oblaku – varovanje osebnih podatkov. *Revizija.si*. Najdeno 25. julija 2016 na spletnem naslovu <http://www.si-revizija.si/sites/default/files/publikacije/ris-19.pdf>
24. Dukarić, R., & Jurič, B.M. (2013). Towards a unified taxonomy and architecture of cloud frameworks. *Future Generation Computer Systems*, 29(1), 1197–1210.
25. Dukarić, R., Povše, R., & Jurič, M.B. (2011). *Računalništvo v oblaku* (interno gradivo). Ljubljana: Fakulteta za računalništvo in informatiko.
26. Eurich, M., Giessmann, A., Mettler, T., & Slabeva, S.K. (2011). Revenue Streams of Cloud-based Platforms: Current State and Future Directions. *Proceedings of the Seventeenth Americas Conference on Information Systems*, (str. 1–10). Najdeno 1. avgusta 2016 na spletnem naslovu https://www.researchgate.net/publication/220891594_Revenue_Streams_of_Cloud-based_Platforms_Current_State_and_Future_Directions
27. Finnegan, M. (2016, 19. januar). Microsoft Azure vs Amazon AWS public cloud comparison: Which Cloud is best for the enterprise? *Computer World UK*. Najdeno 3. julija 2016 na spletnem naslovu <http://www.computerworlduk.com/it-vendors/microsoft-azure-vs-amazon-aws-public-cloud-comparison-which-cloud-is-best-for-enterprise-3624848/>
28. *Gartner – Application Platform as a Service*. Najdeno 31. julija 2016 na spletnem naslovu <http://www.gartner.com/it-glossary/application-platform-as-a-service-apaas/>
29. *Gartner – Integration Platform as a Service*. Najdeno 31. julija 2016 na spletnem naslovu <http://www.gartner.com/it-glossary/information-platform-as-a-service-ipaas/>

30. *Gartner IoT– Says IoT Adoption Is Driving the Use of Platform as a Service*. Najdeno 20. maja 2016 na spletnem naslovu <http://www.gartner.com/newsroom/id/3241817>
31. *Geministyle – Storitveni modeli računalništva v oblaku*. Najdeno 31. julija 2016 na spletnem naslovu <http://www.geministyle.si/print/racunalnistvo/splosno/racunalnistvo-v-oblaku-4.html>
32. *Geministyle – prihodnost računalništva v oblaku*. Najdeno 22. maja 2016 na spletnem naslovu <http://www.geministyle.si/racunalnistvo/splosno/racunalnistvo-v-oblaku-7.php#.V59N37iyOkr>
33. Ghaffari, K., Delgosha, S.M., & Abdolvand, N. (2014). Towards Cloud Computing: A Swot Analysis On Its Adoption In Smes. *International Journal of Information Technology Convergence and Services*, 4(2), 13–20.
34. Gheorghe, M. (2012). Investment decision analysis in information security. *Revista Economică*, 6(5), 85–93.
35. Giessmann, A., & Slabeva, S.K. (2012). Business Models of Platform as a Service (PaaS) Providers: Current State and Future Directions. *Journal of Information Technology Theory and Application*, 13(4), 31–55.
36. *Google Cloud SQL – Google Cloud Platform*. Najdeno 26. julija 2016 na spletnem naslovu <https://cloud.google.com/sql/docs/>
37. Gouda, K.C., & Acharjee, D. (2013). An Outlook of cloud computing. *International Journal of Science, Engineering and Technology Research*, 2(3), 1403–1406.
38. Gouda, K.C., Kar, S. & Patra, R. (2013). Cloud Computing: What, How and Why. *International Journal of Engineering Trends and Technology*, 4(6), 2453–2457.
39. Goyal, S. (2014). Public vs Private vs Hybrid vs Community – Cloud Computing: A Critical Review. *Computer Network and Information Security*, 3(1), 20–29.
40. Greene, T. (2013, 17. april). Microsoft Azure enters price war with Amazon Web Services. *Network World US*. Najdeno 4. julija 2016 na spletnem naslovu <http://www.computerworlduk.com/news/it-vendors/microsoft-azure-enters-price-war-with-amazon-web-services-3442529/>
41. Gupta, P., Seetharaman, A., & Raj, J.R. (2013). The usage and adoption of cloud computing by small and medium businesses. *International Journal of Information Management*, 33(1), 861–874.
42. Hada, D., & Agrawal, A. (2013). Performance analysis based on Google App Engine and Amazon Web Service. *International Journal of Engineering Technology and Computer Research*, 3(2), 107–112.
43. Haile, N., & Altmann, J. (2016). Value creation in software service platforms. *Future Generation Computer Systems*, 55(1), 495–509.
44. Han, Y. (2015). Cloud storage for digital preservation: optimal uses of Amazon S3 and Glacier. *Library Hi Tech*, 33(2), 261–271.
45. Herskowitz, N. (2016, 26. junij). Microsoft Named a Leader in Gartner’s Public Cloud Storage Services for Second Consecutive Year. Microsoft Azure. Najdeno 16. julij 2016 na spletnem naslovu <https://azure.microsoft.com/en-us/blog/microsoft-named-a-leader-in-gartners-public-cloud-storage-services-for-second-consecutive-year/>

46. *IKT Informator – Finance* (2016, 7. april). Do leta 2017 več kot polovica velikih podjetij s hibridnimi oblaki. *Finance*. Najdeno 20. junij 2016 na spletnem naslovu <http://ikt.finance.si/8843445/Do-leta-2017-vec-kot-polovica-velikih-podjetij-s-hibridnimi-oblaki?cctest&>
47. ISACA. (2012). *Calculating Cloud ROI: From the Customer Perspective*. Illinois: ISACA.
48. Jaiswar, S. (2015). PAAS: Demystified. *International Journal of Computer Applications*, 112(12), 10–15.
49. Karunakaran, S., Krishnaswamy, V. & Rangaraja S.P. (2015). Business view of cloud. *Management Research Review*, 38(6), 582–604.
50. Kaur, M., Sood, R., Garg, S., & Palta, P. (2016). A Novel Approach to Data Filtration against Packet Flooded Attacks in Cloud Service. *Journal of Network Communications and Emerging Technologies*, 6(5), 37–41.
51. Kazenski zakonik. *Uradni list RS*, št. 55/2008-KZ-1, 66/2008-KZ-1, 39/2009-KZ-1A, 91/2011-KZ-1B, 50/2012-UPB2.
52. Komšo, A. (b.l.). Poslovna vrednost računalništva v oblaku. *Gospodarska zbornica Slovenije – Združenje za informatiko in telekomunikacije*. Najdeno 31. julija 2016 na spletnem naslovu <https://www.gzs.si/pripone/EuroCloud%20-Poslovna%20vrednost%20CC.pdf>
53. Kragelj, P., & Rajkovič, V. (2012). Kako oceniti ponudnika storitev v oblaku. *Uporabna informatika*, 20(4), 244–249.
54. Kulkarni, G., Khatawkar, P., & Gambhir, J. (2011). Cloud Computing-Platform as Service. *International Journal of Engineering and Advanced Technology* 1(2), 115–120.
55. Marešová, P., & Kuča, K. (2015). Technological environment and SWOT Analysis of Cloud Computing in Europe. *International Journal of Information Technology & Computer Science*, 18(1), 57–77.
56. Marian, M. (2012). The mark of unexpected financial crisis left upon cloud computing. *Revista Economică*, 6(5), 135–138.
57. Masrom, M., & Rahimli, A. (2015). Cloud Computing Adoption in the Healthcare Sector: A SWOT Analysis. *Asian Social Science*, 11(10), 12–18.
58. Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing. *NIST*. Najdeno 30. julija 2016 na spletnem naslovu <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
59. *Microsoft Azure Marketplace*. Najdeno 1. avgusta 2016 na spletnem naslovu <https://azure.microsoft.com/en-us/>
60. *Microsoft Azure*. Najdeno 1. avgusta 2016 na spletnem naslovu <https://azure.microsoft.com/en-us/>
61. *Microsoft Trust Center*. Najdeno 1. avgust 2016 na spletnem naslovu <https://www.microsoft.com/en-us/trustcenter>

62. Murty, J. (2008). *Programming Amazon Web Services: S3, EC2, SQS, FPS, and SimpleDB*. Sebastopol: O'Reilly.
63. Nath, A., Aleya, F.K., Santra, M., & Maji, S. (2016). Security Issues and Challenges in Cloud Computing: A Brief Overview. *International Journal of Emerging Technology and Advanced Engineering*, 6(1), 315–318.
64. Natis, V.Y., Pezzini, M., Iijima, K., Thomas, A., & Dunie, R. (2015). Magic Quadrant for Enterprise Application Platform as a Service, Worldwide. *Gartner*. Najdeno 26. junija 2016 na spletnem naslovu https://www.ibm.com/cloud-computing/files/GARTNER_COMP_magic_quadrant_for_enterpris_271188_Bluemix.pdf
65. Ojala, A. (2016). Discovering and creating business opportunities for cloud service. *The Journal of Systems and Software*, 113(1), 408–417.
66. Ovčjak, B. Jošt, G., Polančič, G., & Heričko, M. (2011). Primerjava učinkovitosti delovanja aplikacij v oblaku in lokalnem okolju. *Uporabna informatika*, 19(4), 193–206.
67. Pardeshi, V.H. (2014). Cloud Computing for Higher Education Institutes: Architecture, Strategy and Recommendations for Effective Adaption. *Procedia Economics and Finance*, 11(1), 590–599.
68. Pastore, S. (2013). The platform as a Service (PaaS) Cloud Model: Opportunity of Complexity for a Web Developer? *International Journal of Computer Applications*, 81(18), 29–37.
69. Prasanth, A. (2012). Cloud Computing Services: A Survey. *International Journal of Computer Applications*, 46(3), 25–29.
70. Računalništvo v oblaku. (b.l.) V *iSlovarju*. Najdeno 7. avgusta 2016 na spletnem naslovu <http://www.islovar.org/islovar>
71. Radut, C., Popa, I., & Codreanu, D. (2012). Cloud Computing Security. *Revista Economică*, 6(5), 171–174.
72. Rani, D., & Ranjan, R.K. (2014). A comparative Study of SaaS, PaaS and IaaS in Cloud Computing. *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(6), 458–461.
73. Rao, M. N. (2015). *Cloud Computing*. New Delhi: SCET Engineering College Narsapuram.
74. Reed, A., & Bennett, S.G. (2011). *Silver Clouds, Dark Linings: A concise guide to cloud computing*. Boston: Prentice Hall.
75. Rittinghouse, J.W., & Ransome, J.F. (2009). *Cloud Computing: Implementation, Management, and Security*. New York: Taylor & Francis Group.
76. Sedlar, U., Bešter, J., & Kos, A. (2011). Računalništvo v oblaku v telekomunikacijah in primeri uporabe. *Laboratorij za telekomunikacije, fakultete za elektrotehniko*. Najdeno 30. julija 2016 na spletnem naslovu <http://www.ltfe.org/wp-content/uploads/2011/11/2-Urban-Sedlar-Janez-Bester-Andrej-Kos-VITELnov2011.pdf>
77. Severance, C. (2009). *Using Google App Engine*. Sebastopol: O'Reilly.

78. Sharma, A., & Garg, S. (2015). Comparative Study of Cloud Computing Solutions. *International Journal of Computer Science and Technology*, 6(4), 231–233.
79. Sharma, S., Sah, M.K., & Pani, A.K. (2015). Cloud based »Online Art Gallery« using PaaS with its Security, Privacy, and Compliance Challenges. *International Journal of Computer Applications*, 123(12), 20–27.
80. Sridevi, R., & Rahinipriyadharshini, R. (2016). Cloud data Security using ECC and Searching of Cloud data using KNN Algorithm. *International Journal of Advanced Research in Science, Engineering and Technology*, 3(3), 1748–1754.
81. Srinivasan, S., & Raja, K. (2014). Security Challenges In Cloud Computing. *International Journal of Emerging Technology and Advanced Engineering*, 4(2), 1–6.
82. Tiwari, P.K., & Mishra, B. (2012). Cloud Computing Security Issues, Challenges and Solution. *International Journal of Emerging Technology and Advanced Engineering*, 2(8), 306–310.
83. Tomić, Ž., & Jovanović, M. (2016). ERP and CRM Data Integration. *Management*. Najdeno 30. julija 2016 na spletnem naslovu http://www.management.fon.rs/management/e_management_78_english_07.pdf
84. Vouk, M.A. (2008). Cloud Computing – Issues, Research and Implementations. *Journal of Computing and Information Technology*, 16(4), 235–246.
85. Weinberger, M. (2015, 7. november). The cloud wars explained: Why nobody can catch up with Amazon. *Business Insider*. Najdeno 5. avgusta 2016 na spletnem naslovu <http://www.businessinsider.com/why-amazon-is-so-hard-to-topple-in-the-cloud-and-where-everybody-else-falls-2015-10>
86. Winkler, V. (2011). *Securing the Cloud*. Waltham: Elsevier.
87. Yehuda-Ben. O.A., Yehuda-Ben, A., Schuster, A., & Tsafir, D. (2013). Deconstructing Amazon EC2 Spot Instance Pricing. *Journal Transactions on Economics and Computation*, 1(3), 1–8.
88. Zakon o varstvu osebnih podatkov. *Uradni list RS*, št. 86/2004, 113/2005-ZInfP, 51/2007-ZUstS-A, 67/2007, 94/2007-UPB1.

PRILOGE

KAZALO PRILOG

PRILOGA 1:

Primerjalna analiza izbranih platform v javnem oblaku glede na skladiščenje podatkovnih baz in Big Data 1

PRILOGA 2:

Primerjalna analiza izbranih platform v javnem oblaku glede na specifikacijo internetne povezave 2

PRILOGA 3:

Primerjalna analiza izbranih platform v javnem oblaku glede na cenovno politiko ponudnika 3

PRILOGA 4 :

Seznam kratic 4

PRILOGA 1: Primerjalna analiza izbranih platform v javnem oblaku glede na skladiščenje podatkovnih baz in Big Data

	SKLADIŠČENJE VIROV IN BIG DATA					
	KRATKOTRAJNO (ZAČASNO)	BLOK ZA SHRANJEVANJE	PREDMET SHRANJEVANJA	RELACIJSKE PODATKOVNE BAZE	ARHIVIRANJE	NOSQL IN BIG DATA
GOOGLE APP ENGINE	DA	DISKI ZA SHRANJEVANJE	GOOGLE CLOUD STORAGE	GOOGLE CLOUD SQL	NEARLINE	CLOUD DATA STORE BIG QUERY HADOOP
MICROSOFT WINDOWS AZURE	/	BLOCK BLOBS (OBJEKTNA SHRAMBA PODATKOV)	OBJEKTNA SHRAMBA IN DATOTEKE	SQL RELACIJSKE PODATKOVNE BAZE	/	WINDOWS AZURE TABLE HDI Insight
AMAZON WEB SERVICE	DA	EBS	S3	RDS	GLACIER	DYNAMO DB EMR KINES READSHIFT

PRILOGA 2: Primerjalna analiza izbranih platform v javnem oblaku glede na specifikacijo internetne povezave

	INTERNETNA POVEZAVA				
	VIRTUALNO OMREŽJE	JAVNI IP	HIBRIDNI OBLAK	SISTEM IMENSKIH DOMEN (DNS)	POŽARNI ZID
GOOGLE APP ENGINE	SUBNET	DA	DA	/	DA
MICROSOFT WINDOWS AZURE	VNET	DA	DA	/	DA
AMAZON WEB SERVICE	VPC	DA	DA	ROUTE 53	DA

PRILOGA 3: Primerjalna analiza izbranih platform v javnem oblaku glede na cenovno politiko ponudnika

	CENOVNA POLITIKA	
	CENITEV UPORABE STORITEV	MODELI CENITVE
GOOGLE APP ENGINE	ZARAČUNAVANJE NA MINUTO (ZAKROŽEVANJE NAVZGOR) MIN. UPORABA 10 MINUT	NA ZAHTEVO NA KRAJU SAMEM
MICROSOFT WINDOWS AZURE	ZARAČUNAVANJE NA MINUTO (ZAKROŽEVANJE NAVZGOR) PREDPLAČNIŠKO ALI MESEČNO ZARAČUNAVANJE	GLEDE NA POVPRASEVANJE NA KRATKOROČNE OBVEZNOSTI (PREDPLAČNIŠKO ALI MESEČNO OBRAČUNAVANJE)
AMAZON WEB SERVICE	ZARAČUNAVANJE NA URO (ZAKROŽEVANJE NAVZGOR) MIN. UPORABA 1 URA	NA ZAHTEVO PRIDRŽANO (POGODBENA VEZAVA)

PRILOGA 4 : Seznam kratic

AMI – Amazonov programski sklop operacijskih sistemov
aPaaS – aplikacijska platforma kot storitev
API – aplikacijski programski vmesnik
AWS – Amazon Web Services platforma
B2B – elektronsko poslovanje med sistemi organizacij
CIA – ameriška civilna obveščevalna služba
CRM – management odnosov s strankami
EC2 – Elastic Compute Cloud
ERP – celovite informacijske rešitve
GAE – Google App Engine platforma
HTTP – niz pravil za prenos datotek (besedil, slik, zvokov, idr.) na svetovnem spletu
IaaS – Infrastruktura kot storitev
IP – internetni protokol
iPaaS – povezovalna platforma kot storitev
IT – informacijska tehnologija
MWS – Microsoft Windows Azure platforma
NIST – ameriški Nacionalni inštitut za standarde in tehnologijo
OData – odprti protokol, ki temelji na poizvedbah preko REST in API vmesnika
PaaS – platforma kot storitev
REST – množica arhitekturnih sistemov za razvoj spletnih storitev
SaaS – programska oprema kot storitev
SCM – management oskrbovalne verige
SFA – Sales Force programska oprema za avtomatizacijo prodaje
SOAP – Standard za razvoj spletnih storitev, ki temelji na XML jeziku
SQL – strukturni povpraševalni jezik
TCO – skupni stroški lastništva