

# ▣ Migracija obstoječih aplikacij na platforme za računalništvo v oblaku

<sup>1</sup>Robert Dukarić, <sup>2</sup>Matjaž B. Jurič

<sup>1</sup>XLAB, d. o. o., robert.dukaric@cloud.si

<sup>2</sup>Fakulteta za računalništvo in informatiko Univerze v Ljubljani, matjaz.juric@fri.uni-lj.si

## Izvleček

Danes je vedno več organizacij in podjetij, ki selijo svoje aplikacije v oblak z namenom, da bi modernizirali svojo informacijsko tehnologijo, znižali stroške ali da bi se pripravili na potrebe, ki bi jih organizacija morebiti imela v prihodnje. Migracija obstoječih aplikacij v oblak omogoča tako malim, srednjim kot tudi velikim podjetjem, da lansirajo nove produkte ali storitve na trg hitreje z minimiziranjem časa za namestitev konkretne informacijske tehnologije, kot so strežniki in druge omrežne komponente. Pri tem podjetja in organizacije eliminirajo investicijske stroške, zmanjšajo operativne stroške vzdrževanja in upravljanja infrastrukture ter si hkrati zagotovijo poslovno agilnost. V prispevku bomo predstavili osnovne tipe migracije v oblak, identificirali ključne prednosti in slabosti migracije ter jo analizirali na dveh različnih produktih računalništva v oblaku – Amazon AWS ter Windows Azure. Na koncu bomo predstavili nov, uniformen migracijski pristop, ki ga je mogoče aplicirati pri migraciji večine obstoječih informacijskih rešitev in je neodvisen od ponudnika računalništva v oblaku.

**Ključne besede:** računalništvo v oblaku, platforma kot storitev, infrastruktura kot storitev.

## Abstract

### Migrating Existing Applications to the Cloud

Several organizations and companies are nowadays moving their applications to the cloud in order to modernize their current IT assets, reduce their costs or to prepare for the requirements that an organization may have in the future. Migration of existing Word 2003.lnk applications to the cloud allows small, medium and large enterprises to bring new products or services to market faster by minimizing the time to install specific IT assets such as servers and other networking components, while substantially eliminating the investment costs and assuring business agility. In this article we present basic types of application migration, identify key advantages and disadvantages of migration and analyse it on two diverse cloud computing products – Amazon AWS and Windows Azure. Finally, we propose a novel, uniform migration approach that can be applied on several existing information systems migrations and is provider independent.

**Keywords:** Cloud computing, Platform as a Service, Infrastructure as a Service.

## 1 UVOD

Migracija aplikacije je proces ponovne postavitve aplikacije, tipično na novo platformo ali infrastrukturo. V primeru oblaka je lahko aplikacija prenesena iz obstoječega podatkovnega centra v poljuben ciljni oblak. Ciljno infrastrukturo lahko predstavlja javni, zasebni ali hibridni oblak. Zadnji transparentno kombinira zmožnosti prvih dveh. Aplikacije lahko ravnano tako migriramo na različne tipe oblaka, kot sta infrastruktura kot storitev (IaaS) ter platforma kot storitev (PaaS). Pri identifikaciji aplikacij, ki jih želimo prenesti v oblak, je najprej treba preučiti in razumeti poslovne in tehnične faktorje migracije. Zmanjšanje stroškov in poslovna agilnost sta dva tipična poslovna faktorja za migracijo obstoječih aplikacij v oblak. Računalništvo v oblaku lahko zaradi povečane utilizacije ponudi bistvene prihranke. Povečana utilizacija je

posledica večnajemniškega modela (Multitenancy) in standardizacije ter avtomatizacije oblačnih storitev. Iz operativnega vidika so upravljanje, zmogljivost ter skalabilnost tipični vzroki, zakaj podjetja razmišljajo o prisvojitvi oblaka.

V nadaljevanju bomo opisali glavne tipe migracije v oblak, predstavili ključne prednosti in slabosti migracij ter analizirali dve možnosti prenosa obstoječih aplikacij v oblak. V prvem primeru bomo obravnavali migracijo klasične aplikacije na platformo Windows Azure, medtem ko bo drugi del opisoval prenos aplikacije v Amazonov oblak AWS. Obravnavana bosta dva različna migracijska pristopa: štirifazni migracijski pristop, ki ga predlaga Microsoft, ter Amazonov šestfazni migracijski pristop. V nadaljevanju bomo identificirali bistvene pomanjkljivosti

posameznega pristopa ter predstavili nov uniformen pristop, ki uspešno rešuje identificirane slabosti ter ga je mogoče aplicirati pri migraciji večine aplikacij in podatkov ter je neodvisen od ponudnika računalništva v oblaku.

## 2 MIGRACIJA APLIKACIJ V OBLAK

### 2.1 Tipi migracije v oblak

Računalništvo v oblaku lahko ponuja storitve infrastrukture, kot so npr. strežniki, omrežja ter shramba podatkov, kar z drugimi besedami imenujemo infrastruktura kot storitev (IaaS). Lahko ponuja aplikacijsko platformo z aplikacijskimi storitvami, kot je relacijska podatkovna baza, razvojno okolje itn. Takšno obliko imenujemo platforma kot storitev (PaaS). Nazadnje lahko ponuja aplikacije, ki so svojim uporabnikom zaračunane na mesečni ravni in so nameščene pri ponudniku oblaka, tj. programska oprema kot storitev (SaaS).

Danes številni ponudniki storitev omogočajo oskrbovanje, upravljanje in skaliranje storitev več končnim uporabnikom. Hkrati ponujajo zmožnosti, ki so zasnovane na ravni IaaS, pri čemer so številne organizacije za porabljene storitve zaračunane glede na poslovni model »plačaj toliko, kot uporabiš« (Pay-as-you-go). Na drugi strani številna podjetja in organizacije vzpostavljajo lastne zasebne oblake, pri katerih gre za infrastrukturne storitve informacijske tehnologije, ki jih upravlja organizacija sama in ravno tako podpira koncepte, kot so samopostrežba, poslovni model v obliki storitve, oskrbovanje na zahtevo ter občutek neskončnosti virov.

Ne glede na to, ali se organizacije odločijo za zasebni ali javni oblak kot njihov storitveni model, morajo najprej identificirati, katere aplikacije je smiselno prenesti v oblak in kako izvesti samo migracijo. Aplikacije so lahko torej prenesene iz obstoječih podatkovnih centrov v ciljni oblak, ki je lahko javni, zasebni ali hibridni. Ko je določena aplikacija identificirana kot kandidat za migracijo v oblak glede na poslovne in tehnične faktorje, je pomembno ugotoviti, za kateri tip računalništva v oblaku – PaaS ali IaaS – je aplikacija najbolj primerna (Cisco Systems Inc., 2010).

#### Platforma kot storitev

Platforma kot storitev lahko predstavlja možnost za migracijo poslovnih aplikacij, ki temeljijo na standard-

nih aplikacijskih strežnikov, kot so aplikacijski strežniki Java ali strežniki Microsoft SQL Server. Pri tem modelu ponudnik storitev upravlja aplikacijsko platformo in ponuja dostop do aplikacijskih storitev, kot je npr. podatkovna baza SQL. Aplikacijska platforma je lahko skupna več aplikacijam, pri čemer vsaka pripada drugi stranki. Kako je aplikacijska platforma povezana s fizično infrastrukturo, običajno nadzira ponudnik računalništva v oblaku. Odločitveni faktorji pri migraciji aplikacije so odvisni od tipa in verzije aplikacijskega strežnika, ki ga je naša aplikacija uporabljala pred tem. Nekatera okolja PaaS ne podpirajo vseh funkcionalnosti aplikacijskih strežnikov in lahko zahtevajo spremembe aplikacij. Merila, ki jih je treba upoštevati pri postavitvi PaaS, so (Cisco Systems Inc., 2010; Kaisler & Money, 2011):

- **dogovor o ravni storitve (SLA):** ponudnik storitev PaaS mora ponuditi SLA za razpoložljivost in zmožljivost aplikacijske platforme. Ponudnik mora ravno tako definirati jasno politiko in smernice za vzdrževanje in upravljanje verzij (Version Management) za platformo ter politike za kompatibilnost verzij API-jev med platformo in aplikacijo;
- **portabilnost podatkov:** podatki aplikacije so običajno shranjeni v podatkovni bazi, ki jo priskrbi ponudnik storitev oblaka. Uporabnik mora imeti možnost izvoziti podatke v format, ki bo omogočil migracijo k drugemu ponudniku. Enako velja za statične podatke, shranjene znotraj omrežnih datotečnih sistemov;
- **dolgoročni stroški:** finančni model ravni PaaS mora biti primerjan z modelom v internih namestitvah infrastrukture in aplikacijske platforme, ki uporablja model IaaS za postavitev aplikacijskih strežnikov na strežnike oblaka;
- **upravljanje uporabnikov:** aplikacija, postavljena na ravni PaaS, zahteva administrativne in uporabniške račune. Za oba tipa računov se mora uporabnik zavedati, kako je upravljanje z uporabniki usklajeno z obstoječimi imeniškimi storitvami (Directory Services) in procesi upravljanja z uporabniki (User Management Processes);
- **varnost:** v okolju PaaS lahko isti fizični strežnik izvaja aplikacije različnih uporabnikov. V takšnih večnajemniških okoljih (Multi-tenant Environments) so zahtevani dodatni varnostni ukrepi, ki zagotavljajo varno izolacijo posameznih aplikacij;
- **upravljanje s platformo:** aplikacijski strežniki ponujajo upravljaljske konzole (Management

Consoles) in orodja za spremljanje in upravljanje aplikacij, ki se izvajajo na njih. Okolje PaaS mora uporabnikom ponujati analogen nabor orodij za upravljanje in optimiziranje zmogljivosti aplikacij.

### Infrastruktura kot storitev

Pri migraciji aplikacije na model IaaS je treba najprej ugotoviti, ali sta strežniška strojna oprema in operacijski sistem kompatibilna s trenutno strežniško strojno opremo in operacijskim sistemom. Če se aplikacija izvaja na strežnikih x86, mora ponudnik oblaka ravno tako implementirati inštrukcije x86. Če strojna oprema ni kompatibilna, je treba aplikacijo ponovno prevesti in postaviti na novo platformo. Če je operacijski sistem kompatibilen, bo pri migraciji aplikacije potrebnih le nekaj sprememb. Večino meril, ki smo jih upoštevali pri migraciji aplikacij v okolje PaaS, moramo upoštevati tudi na ravni IaaS (Cisco Systems Inc., 2010; Kaisler & Money, 2011):

- **dogovor na ravni storitve (SLA):** v okolju ravni IaaS je treba zagotoviti SLA za razpoložljivost in zmogljivost delovanja strežnikov, omrežja in shrambe podatkov. Ravno tako morajo biti zagotovljeni dogovori o vzdrževanju in upravljanju infrastrukture in dogovori o času prekinitve (Downtime);
- **portabilnost podatkov:** aplikacija lahko uporablja strežnik podatkovne baze, ki se prav tako nahaja v oblaku. V tem primeru mora ponudnik storitev oblaka zagotoviti replikacijo in migracijo blokovne shrambe ali datotečnega sistema, ki se nahaja na strežnikih obstoječega podatkovnega centra;
- **dolgoročni stroški:** stroški aplikacije, postavljene na modelu IaaS, morajo biti primerjani s stroški postavitve v podatkovnih centrih organizacij. V nekaterih primerih ima javna namestitev modela IaaS očitne prednosti zaradi dinamičnega skaliranja in zaračunavanja glede na porabo. Za aplikacije v javnem oblaku so lahko dolgoročni stroški lastništva v nekaterih primerih tudi večji od stroškov lastništva v zasebnem oblaku;
- **upravljanje uporabnikov:** v okolju IaaS so zahtevane vsaj tri različne vloge uporabnikov – strežniški administrator, aplikacijski administrator in uporabnik aplikacije;
- **varnost:** virtualni stroji, ki pripadajo različnim strankam, so lahko implementirani na skupni fizični infrastrukturi. Preden aplikacijo prenesemo

v oblak, je treba naprej preveriti varnostne politike za virtualno in fizično izolacijo ponudnikove infrastrukture ter skladnost z zakonodajo;

- **skalabilnost:** aplikacije, ki so zasnovane za horizontalno skaliranje, so običajno večslojne (Multitiered) in imajo funkcijo za izenačevanje obremenitev, ki omogoča, da so lahko aplikacije brez stanj (Stateless Applications) skalirane navzgor ali navzdol. V primeru uporabe te funkcije mora ponudnik storitev oblaka ponuditi jasno politiko o tem, kako bo deloval ta tip skaliranja in kako bo poskrbljeno za dodeljevanje virov med več uporabniki in aplikacijami.

## 2.2 Prednosti in slabosti migracije v oblak

Računalništvo v oblaku ponuja organizacijam številne prednosti. Prva večja prednost migracije v oblak je, da organizacijam ni treba veliko investirati v fizično in programsko infrastrukturo, potrebna je le dobra internetna povezava. Ravno tako so odpravljeni operativni stroški za vzdrževanje in upravljanje infrastrukture. Zmožnost oblaka, da razširi svoje kapacitete samodejno glede na potrebe (elastičnost), je ključna prednost, ki diferencira računalniški oblak od drugih oblik gostovanja (Buyya et al., 2009). Tipično računalništvo v oblaku operira s poslovnim modelom »Pay-as-you-go«, kar nam omogoča, da dejansko najamemo toliko računalniških virov in storitev, kot jih potrebujemo. Naslednja prednost migracije je možnost dostopa do naših aplikacij in podatkov iz katerih koli naprav, ki imajo dostop do interneta. Oblak prav tako zagotavlja visoko stopnjo zanesljivosti in razpoložljivosti, saj omogoča replikacijo podatkov tudi med različnimi podatkovnimi centri in številne mehanizme za okrevanje po katastrofi (Disaster Recovery) (Kaisler & Money, 2011; Lee & Lee & Cheun, 2009). Ena izmed pomembnih prednosti prenašanja aplikacij in podatkov v oblak je agilnost, ki omogoča organizacijam večjo odzivnost na spremembe pri poslovanju in posledično hitrejšo lansiranje novih storitev na tržišče. Migracija aplikacij v oblak ravno tako zagotavlja neodvisnost od naprave in lokacije uporabnika ter s tem poveča dostopnost do aplikacij in podatkov (Smith, 2009).

Računalništvo v oblaku podpira izvajanje storitev in aplikacij, ki so že nameščene in dostopne v enem ali več podatkovnih centrih. Aplikacije, ki najbolj ustrezajo okolju računalništva v oblaku, so običajno tiste, ki strežejo veliki količini uporabnikov, kot so

e-poštne aplikacije, različni tipi splošnonamenskih spletnih aplikacij, socialna omrežja, sistemi ERP in CRM (Kaisler & Money, 2011). Ena izmed ključnih ovir pri migraciji aplikacij v oblak je zaklepanje znotraj ponudnika (Vendor Lock-In). To pomeni, da je podatke in aplikacije zelo težavno prenesti drugam, ko se enkrat odločimo za določenega ponudnika storitev računalništva v oblaku. Razlog je v tem, da različni ponudniki ponujajo lastne nestandardizirane formate podatkov in aplikacij (Borenstein & Blake, 2011). Naslednji oviri, ki močno vplivata na odločitev migracije informacijskih rešitev v oblak, sta varnost in zasebnost podatkov, ki povzročata podjetjem in organizacijam največ skrbi, še posebno ko je govor o prenosu poslovno kritičnih podatkov v roke zunanjemu ponudniku (Kandukuri, Paturi & Rakshit, 2011). Geografska lokacija je naslednja večja ovira. Če se podjetja odločijo gostovati svoje aplikacije v podatkovnih centrih na območju Združenih držav Amerike, bi v skladu s predpisi zakona U. S. Patriot Act imela vlada možnost vpogleda v njihove podatke. Ponudnik mora torej omogočiti svojim uporabnikom možnost izbire, v katerih podatkovnih centrih se bodo nahajale njegove aplikacije. Obstajajo še nekateri drugi pomisleki, ki so povezani z zagotavljanjem stabilne internetne povezave, visoke stopnje razpoložljivosti ter pomanjkanje robustnih dokumentov SLA (Smith, 2009; Kandukuri et al., 2011; Alhamed, 2010).

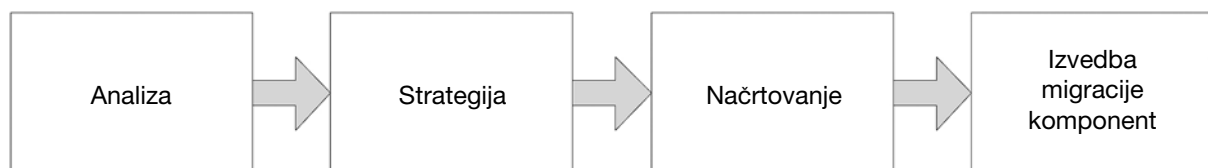
### 2.3 Migracija na Windows Azure

Zmogljivosti platforme Windows Azure, ki omogočajo razvijalcem uporabo različnih programerskih tehnik v razvojnih okoljih Visual Studio in Eclipse, so primarna prednost platforme Windows Azure, ki mora konkurirati drugim produktom računalništva v oblaku, kot so Amazonov AWS, Force.com ter Googlov App Engine. Upravljanje in vzdrževanje aplikacij lahko zahteva več časa in energije kot razvoj popolnoma nove programske opreme ali nadgradnja obstoječe. Ključne ovire migracije aplikaci-

je na Azure so premik spletne aplikacije, migracija podatkov v tabele Azure, blobe ali relacijsko bazo SQL Azure, premik projektov spletnih aplikacij v spletne in delovne vloge (Web/Worker roles), povezava s SQL Azure Database in prepričevanje vodstva informatike ter organizacijskega vodstva, da gostovanje aplikacij v Windows Azure ne pomeni bistvenih tveganj, poslovnih motenj oz. regularnih kršitev. Zadnja ovira je v začetni fazi migracije na platformo Windows Azure še največji problem, ki ga ni lahko premagati. Prvi korak pri razvoju aplikacij Azure je torej prepričati informatike ter organizacijsko vodstvo, da je oblak primerno okolje za gostovanje obstoječih ali novih projektov. Začetni ugovori in argumenti bodo verjetno povezani z zupanjem tretji organizaciji, ki bo ponujala razpoložljivost aplikacij, boljših od lokalno nameščenih v podjetjih oz. organizacijah. Vzdrževanje popolne zaupnosti dragocenih poslovnih informacij in njihova shramba sta največja skrb v vrhu vseh organizacij (Jennings, 2009).

#### Analiza, strategija ter načrtovanje migracije

Organizacije se sprašujejo, kako jim lahko zmožnosti in storitve računalništva v oblaku pomagajo oz. koristijo. Da bi to razumele, morajo najprej analizirati trenutno stanje svoje tehnološke infrastrukture in aplikacij ter oceniti, kje bi želele biti v prihodnje. Ko organizacije uspejo identificirati, katere oblačne prednosti – razširljivost, elastičnost, rapidna postavitve itn. – jim bodo najbolj ustrezale, želijo izbrati ocenjevalno ter migracijsko strategijo, ki bo najbolje ustrezala poslovnim potrebam. Ko so aplikacije identificirane kot del celotne strategije, je treba začeti razmišljati o njihovi arhitekturi. Aplikacije, ki ustrezajo določenim arhitekturnim konfiguracijam, bodo ustrezale tudi določenim migracijskim vzorcem. Iz tega razloga bo identifikacija aplikacij, ki ustrezajo določenemu migracijskemu vzorcu, olajšala evalvacijo ter načrtovanje uspešnih migracij (Cunningham, 2010). Ta postopek je prikazan na sliki 1.



Slika 1: **Postopek analize, izbire strategij ter načrtovanja migracije**

Da bi dosegli optimalne rezultate migracije, je treba najprej analizirati številne tehnične, poslovne ter finančne aspekte.

### Izvedba migracije

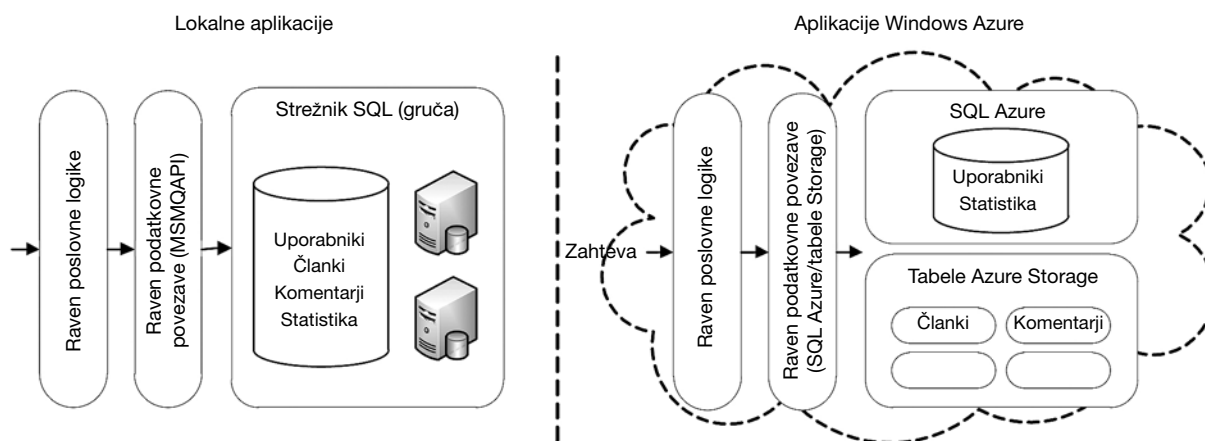
Medtem ko je vsak scenarij migracije Windows Azure različen, ta razdelek ponuja pristop ter tehnični pogled o tem, kako opraviti migracijo nekaterih podsistemov določene aplikacije. Za primer bomo vzeli poljubno lokalno aplikacijo z možnostjo rapidne razširljivosti, ki je gostovana v podatkovnih centrih s spletnimi strežniki v ospredju ter gručami podatkovne baze v ozadju. V nadaljevanju bomo prikazali sce-

narij migracije strežnika Microsoft SQL kot premik enega izmed podsistemov spletne aplikacije (slika 2).

Lokalna aplikacija, ki jo vzamemo za primer, uporablja centralizirano podatkovno bazo strežnika Microsoft SQL. V našem primeru je relacijska baza sestavljena iz dveh zelo obsežnih tabel. Za migracijo v oblak obstajata dve možnosti:

- premik velikih tabel v tabele Windows Azure,
- segmentacija podatkov teh dveh tabel prek več primerkov storitve SQL Azure.

Za nadaljnje delo smo se za shrambo podatkov odločili izbrati tabele Windows Azure, predvsem zaradi nekompleksne razširljivosti ter nižje cene uporabe.



Slika 2: Primer migracije podatkovne baze SQL Server v oblak

## 2.4 Migracija na Amazonov oblak AWS

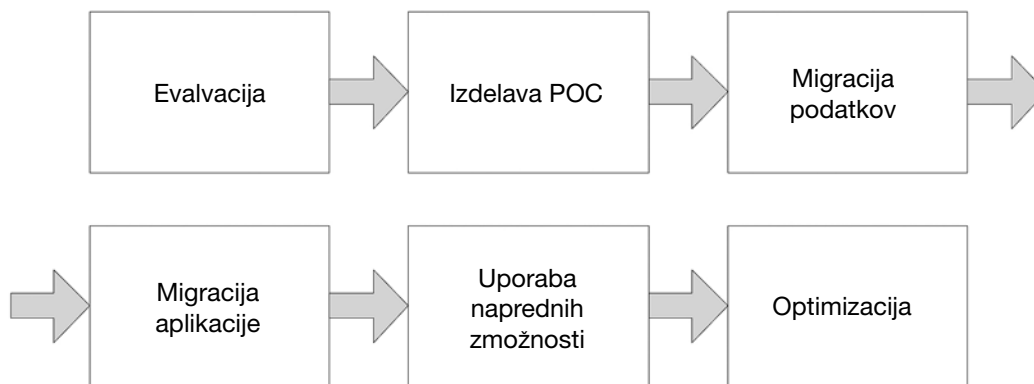
V tem poglavju bomo opisali korake in tehnike, ki so potrebni za migracijo obstoječih aplikacij v Amazonov oblak AWS, pri čemer bomo uporabili šestfazni pristop (slika 3). Ta pristop je sestavljen iz šestih faz:

- faze evalvacije,
- faze izdelave PoC (Proof of Concept),
- faze migracije podatkov,
- faze migracije aplikacije,
- faze uporabe naprednih zmožnosti ter
- faze optimizacije.

Razvijalci in arhitekti, ki imajo namen razviti novo aplikacijo za oblak, lahko preprosto oblikujejo komponente, procese in delovne tokove za njihovo aplikacijo, uporabijo API-je izbranega oblachnega ponudnika in sledijo dobrim praksam za načrtovanje, razvoj, testiranje ter postavitve aplikacij. Če se razvijalci odločijo postaviti svoje rešitve v oblachno infrastrukturo, kot je Amazon Web Services (AWS), lahko uporabi-

jo številne prednosti, kot so skalabilnost, elastičnost, izolacija procesov, zmanjšan operativni trud, oskrbovanje na zahtevo ter avtomatizacija (Varia, 2010).

Hkrati veliko podjetij išče načine, kako najbolje prenesti svoje obstoječe aplikacije v oblachno infrastrukturo, tako da lahko uporabijo prej naštete prednosti. Ključni diferenciator Amazonovih storitev pred platformo Windows Azure je njegova fleksibilnost. Podjetjem daje svobodo izbire programskih modelov in jezikov, operacijskih sistemov ter podatkovnih baz, s katerimi so že dodobra seznanjeni. Posledično danes veliko organizacij seli svoje aplikacije v oblak. Nekaterih aplikacij oz. informacijskih sredstev, postavljenih in nameščenih znotraj podatkovnih centrov organizacij, verjetno s poslovnega in tehničnega vidika ni smiselno premikati v oblak. Definitivno pa obstajajo informacijska sredstva oz. deli aplikacije znotraj organizacije, ki jih lahko brez večjega napora prenesemo v oblak in s tem veliko pridobimo.

Slika 3: **Večfazni pristop migracije obstoječih aplikacij v oblak**

Postavitev aplikacije v oblak AWS zniža infrastrukturne stroške, poveča poslovno agilnost in odstrani upravljanje z lastnim podatkovnim centrom. Uspešna migracija je odvisna od treh dejavnikov: kompleksnosti arhitekture aplikacije, stopnje šibke sklopljenosti aplikacije ter od truda, ki smo ga pripravljene vložiti v samo migracijo (HyperStratus, 2009; Khajeh-Hosseini, Greenwood & Sommerville, 2010).

### 3 PRIMERJAVA MIGRACIJ NA AWS IN AZURE

Ko organizacije razmišljajo o migraciji aplikacij na Windows Azure ali v oblak AWS, je kritičnega pomena, da razumejo težave, ki jih želijo rešiti pri tem – tako s poslovnega kot tudi s tehnološkega vidika. Dobro načrtovana aplikacija z robustno arhitekturo lahko razširi svoje zmožnosti in bistveno zniža stroške, če jo prenesemo na eno izmed opisanih možnosti. Na drugi strani pa slabo načrtovane aplikacije niso nič bolj učinkovite v oblaku, kot če bi bile postavljene v klasičnem podatkovnem centru. Vsi že prej znani ključni elementi dobro načrtovane aplikacije veljajo tudi za postavitev v oblak, pri čemer pa so še toliko bolj pomembni.

Za razliko od Amazonovega šestfaznega migracijskega pristopa Microsoft predlaga selitev aplikacije v štirih fazah. Organizacije, ki želijo prenesti svoje obstoječe aplikacije na platformo Windows Azure, morajo najprej skozi fazo analize. V tej fazi je treba analizirati trenutno stanje tehnološke infrastrukture in aplikacij, da bi ocenili trenutno in hkrati želeno

prihodnje stanje. Prav tako je v tej fazi treba identificirati, katere oblačne prednosti – razširljivost, samopostrežba, agilnost, elastičnost, hitra postavitve, federacija itn. – bodo organizaciji najbolj ustrezale. V drugi fazi migracije bo organizacija morala izbrati ocenitev ter migracijsko strategijo, ki bo najbolj ustrezala njenim poslovnim potrebam. Ko so aplikacije identificirane kot del celotne strategije, je treba začeti razmišljati o njihovi arhitekturi. Identifikacijo aplikacij, ki ustrezajo določenemu migracijskemu vzorcu, izvedemo v fazi načrtovanja. S tem bistveno olajšamo ocenitev uspešno načrtovanih migracij. Ko smo enkrat izvedli vse tri faze predlaganega migracijskega pristopa, je treba izvesti ločeno migracijo vsakega izmed podsistemov aplikacije. Migracije, ki jih je treba realizirati, so migracija podatkovne baze, migracija omrežnega datotečnega sistema, migracija spletne aplikacije, migracije sporočilnih vrst, migracija procesnih vrst ter migracija distribuiranega predpomnilnika. Tabela 1 prikazuje primerjavo migracije posameznega podsistema aplikacije Amazon AWS in aplikacije, ki se izvaja na Windows Azure. Štirifazni pristop, ki ga predlaga Microsoft, sicer uspešno premakne aplikacijo v oblak, vendar ima nekatere pomanjkljivosti, in sicer daje premalo poudarka tehnični evalvaciji, evalvaciji stroškov, varnosti ter evalvaciji zakonodaje. Ravno tako ne predpisuje izgradnje pilotne aplikacije kot zelo pomembnega člena migracije ter ne vpeljuje koncepta optimizacije (tabela 2).

Tabela 1: Primerjava migracij logičnih komponent aplikacij

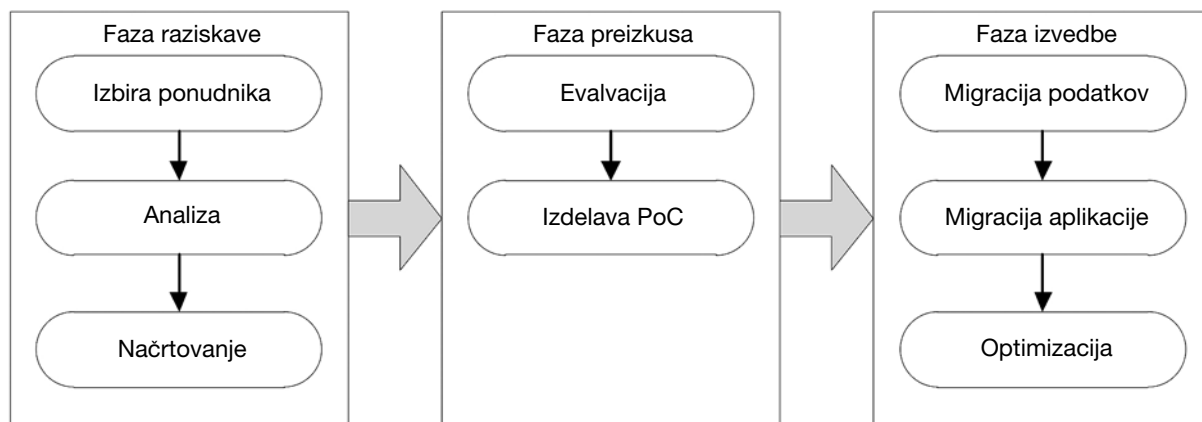
Komponente	Lokalno (On-premises)	Platforma Windows Azure	Amazon AWS
<b>Spletna aplikacija</b>	Internet Information Services (IIS), IBM Websphere, Apache HTTP Server, Tomcat itn.	Spletna ali delovna vloga (Web role/Worker role)	Elastic Compute Cloud (EC2)
<b>Relacijska podatkovna baza</b>	Oracle, DB2, SQL Server, MySQL ali kateri drugi SUPB	SQL Azure	Relational Database Service (RDS), EC2 + Elastic Block Store (EBS)
<b>Nerelacijska podatkovna baza (NoSQL)</b>	/	Storage Tables	SimpleDB
<b>Sporočilne vrste</b>	Java Message Service (JMS), Microsoft Message Queuing (MSMQ), JBoss Messaging itn.	Storage Queues	Simple Queue Service (SQS)
<b>Distribuirani predpomnilnik</b>	Memcached, Hazelcast, JBoss TreeCache itn.	AppFabric Caching	Elastic MapReduce
<b>Statični podatki</b>	Omrežni datotečni sistem (NFS)	Azure Blobs + Content Delivery Network (CDN)	Simple Storage Service (S3) + CloudFront

Večfazni pristop za migracijo obstoječih aplikacij v oblak AWS, ki ga razvijalcem priporoča Amazon, sestavlja šest soodvisnih faz. V prvi fazi, ki jo imenujemo faza evalvacije, je najprej treba izvesti ocenitev varnosti in tehnično evalvacijo. Pri ocenitvi varnosti je treba odgovoriti na vprašanja, kot so, kakšne so naše skrbi glede zaupnosti, integritete in dostopnosti podatkov, kakšne varnostna tveganja obstajajo ali kakšna je celotna toleranca tveganja. V primeru tehnične evalvacije nas zanima, katere aplikacije so bolj primerne za oblak z arhitekturne ter s strateške perspektive. Organizacije v tem primeru med drugim določijo, katere aplikacije premakniti v oblak najprej, katere kasneje in katere naj ostanejo v internem podatkovnem centru. Ko smo identificirali ustrezne kandidate za oblak in ocenili trud, ki je potreben za migracijo, je čas za izdelavo rešitve »Proof of Concept«. Cilj te faze je preučiti AWS in zagotoviti, da bodo točne naše predpostavke glede ustreznosti migracije v oblak. V tej fazi lahko razvijemo in postavimo testno aplikacijo v oblak in tako potipamo teren v oblaku AWS. To storimo tako, da razvijemo pilotno rešitev, ki pomeni fragment naše aplikacije in testira kritične funkcionalnosti aplikacije v okolju oblaka. V fazi migracije podatkov, ki je tretja faza, si morajo organizacije postaviti tale vprašanja: katere možnosti za shrambo podatkov obstajajo v oblaku, katere relacijske podatkovne baze (komercialne in odprtokodne) so na voljo v oblaku, kakšna je naša strategija, povezana s segmentacijo podatkov, ter koliko truda je treba vložiti za migracijo vseh podatkov v oblak. V fazi migracije aplikacije moramo odgovoriti na ključno vprašanje, kako premakniti ves poslovni sistem oz. njegov del v oblak, ne da bi pri tem oškodovali oz. onemogočili trenutno poslovanje. Po uspešnem prenosu aplikacije v oblak in opravljenih vseh po-

trebnih testih sledi faza uporabe oblaka. V tej fazi je treba ugotoviti, kakšne dodatne prednosti nam lahko prinese oblak, kaj je treba storiti za uspešno realizacijo elastičnosti in skalabilnosti ter kako lahko avtomatiziramo procese za lažje upravljanje in vzdrževanje aplikacij v oblaku. V zadnji, šesti fazi, ki jo imenujemo faza optimizacije, se je treba osrediniti na možnosti optimizacije oblačne aplikacije z razlogom, da bi še dodatno zmanjšali stroške. Ker smo zaračunani le za tiste storitve in vire, ki jih dejansko porabimo, je zelo zaželeno prizadevanje po optimizaciji. Amazonov šestfazni pristop sicer odpravlja vse pomanjkljivosti, ki smo jih v prejšnjem paragrafu očiteli pristopu Microsofta, vendar na drugi strani premalo poudarja analizo, strategijo in samo načrtovanje migracije (tabela 2). Slabost, ki je skupna obema pristopoma, je odsotnost pogajanj SLA in jasne definicije formalnih dogovorov o storitvenih pogojih med ponudnikom in uporabnikom računalništva v oblaku.

#### 4 PREDLAGANI MIGRACIJSKI PRISTOP

Danes številne organizacije in podjetja želijo prenesti svojo informacijsko okolje oz. del tega okolja v oblak. Vsak izmed ponudnikov računalništva v oblaku predlaga svoje migracijske strategije, ki se razlikujejo od drugih. To pri vodilnih zaposlenih v organizacijah vzbuja zmedo in nezaupanje. Predlagani migracijski postopek (slika 3) je uniformen način, kako se lotiti migracije informacijskih rešitev in aplikacij v oblak, ter zagotavlja neodvisnost od ponudnika računalništva v oblaku. Ravno tako rešuje pomanjkljivosti komercialnih pristopov, ki smo jih identificirali v prejšnjem razdelku in prikazali v tabeli 2. Pristop, ki ga predlagamo, sestavljajo tri ključne faze: faza raziskave, faza preizkusa ter faza izvedbe. V nadaljevanju bomo vse tri predstavili bolj podrobno.

Slika 4: **Predlagani migracijski pristop**

Faza raziskave je sestavljena iz treh korakov:

- **izbira ponudnika:** v prvem koraku faze raziskave je najprej treba odločiti, na katero raven oblaka želimo seliti aplikacije (IaaS ali PaaS) ter kateri namestitveni model nam najbolj ustreza – javni, zasebni ali hibridni. Ko sta tip in namestitveni model določena, je treba izbrati ponudnika storitev računalništva v oblaku, ki bo ponudil informacijsko okolje za postavitve naše aplikacije. Pri izbiri je smiselno upoštevati dogovor SLA, da bi definirali zaupanje in kakovost storitev – QoS, ki jo različni ponudniki ponujajo svojim uporabnikom, kakor tudi dogovorjeno ogrodje (Agreed Framework) za stroške in ceno (Andrieux et al., 2004). SLA je zelo pomemben kot pogodba med ponudnikom in uporabnikom storitev. Ključna ideja SLA je dati jasno definicijo formalnih dogovorov o storitvenih pogojih (Service Terms), kot so zmogljivost, razpoložljivost in zaračunavanje. Zelo pomembno je, da SLA vključuje obveze in akcije, ki bodo izvršene v primerih kršitve z jasno definirano semantiko med posamezno entiteto, ki je vključena v elektronski pogodbi (Alhamed, Dillon & Chang, 2010);
- **analiza:** v drugem koraku je treba analizirati trenutno stanje tehnološke infrastrukture in aplikacij ter oceniti, kje bi želeli biti v prihodnje. Da bi dosegli optimalne rezultate migracije, je najprej treba analizirati številne tehnične, poslovne ter finančne aspekte. Identificirati je torej treba, katere prednosti oblaka – razširljivost, elastičnost, rapidna postavitve, agilnost, redukcija stroškov, visoka razpoložljivost, zanesljivost – bodo najbolj ustrezale organizacijam in podjetjem;
- **načrtovanje:** v zadnjem koraku faze raziskave je treba uskladiti poslovne potrebe z identificiranimi prednostmi oblaka in izbrati ocenjevalno ter migracijsko strategijo, ki bo najbolj ustrezala poslovnim potrebam. Ko so aplikacije identificirane kot del celotne strategije, je treba začeti razmišljati o njihovi arhitekturi. Aplikacije, ki ustrezajo določenim arhitekturnim konfiguracijam, bodo ustrezale tudi določenim migracijskim vzorcem. V tem koraku je torej ključnega pomena, da identificiramo aplikacije, ki ustrezajo določenemu migracijskemu vzorcu in s tem bistveno olajšamo ocenitev ter načrtovanje uspešnih migracij.

Druga faza (faza preizkusa) vsebuje dva koraka:

- **evalvacija:** v tem koraku je treba najprej izvesti oceno varnosti, stroškov ter zakonodaje in tehnično evalvacijo. Pri oceni varnosti je treba odgovoriti na vprašanja, kot so: kakšne so naše skrbi glede zaupnosti, integritete in dostopnosti podatkov, kakšna varnostna tveganja obstajajo ali kakšna je celotna toleranca tveganja. V primeru tehnične evalvacije nas zanima, katere aplikacije so bolj primerne za oblak z arhitekturne ter s strateške perspektive. Organizacije v tem primeru med drugim določijo, katere aplikacije premakniti v oblak najprej, katere kasneje in katere naj ostanejo v internem podatkovnem centru. Pri tem si lahko pomagamo z izdelavo drevesa odvisnosti (Dependency Tree) ter klasifikacijske tabele (Classification Chart) posameznih logičnih komponent aplikacije. Pri oceni stroškov je treba primerjati stroške, ki jih trenutno imamo v klasičnem podatkovnem centru, in stroške, ki bi jih imeli v primeru izvajanja aplikacije pri izbranem ponudniku. Pri oceni zakonodaje je tre-



ba identificirati politike posameznih ciljnih držav o shranjevanju podatkov ter lokalno zakonodajo, ki v večini primerov ne omogoča selitve zasebnih in poslovnih podatkov zunaj matične države;

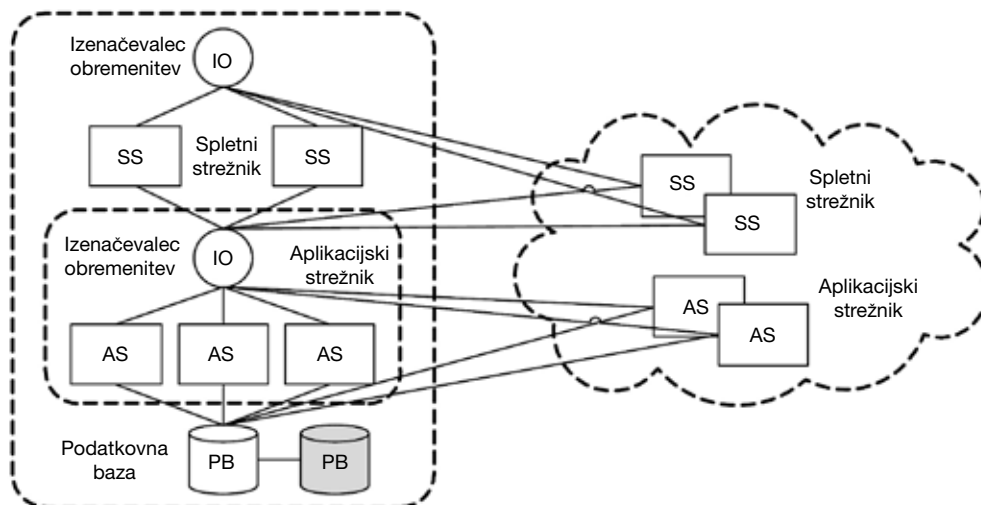
- **izdelava PoC:** v zadnjem koraku faze preizkusa je treba preučiti tehnologijo, ki jo ponuja izbrani ponudnik, in preveriti, ali so bile točne naše predpostavke glede ustreznosti migracije v oblak. V tej fazi razvijemo in postavimo testno aplikacijo in tako potipamo teren pri izbranem ponudniku storitev računalništva v oblaku. To storimo tako, da razvijemo pilotno rešitev, ki je jedro naše aplikacije in testira kritične funkcionalnosti v okolju oblaka. V tem koraku lahko izboljšamo podporo pri vodilnih členih organizacije, validiramo tehnologijo, testiramo »legacy« programsko opremo v oblaku ter definiramo glavna pričakovanja. V tem koraku je treba temeljito preučiti tehnologijo izbranega ponudnika ter ugotoviti, ali bomo s pomočjo pilota prepričali vodstvo organizacije in pridobili ustrezno podporo. Ravno tako je treba raziskati, koliko truda je treba vložiti v premik pilota v produkcijo, ter prepoznati, katere aplikacije so naslednji kandidati za premik v oblak.

Faza izvedbe, ki je zadolžena za dejansko migracijo, predpisuje tri korake:

- **migracija podatkov:** v prvem koraku faze izvedbe morajo organizacije najprej identificirati, katere možnosti za shrambo podatkov obstajajo v oblaku – od relacijske podatkovne baze in nerelacijske podatkovne baze vse do shranjevanja statičnih podatkov (tabela 1). V naslednjem koraku je treba

ugotoviti, katere podatkovne baze (komercialne in odprtokodne) so na voljo v oblaku ter katere so njihove prednosti in slabosti. V primeru relacijske podatkovne baze je treba uskladiti strategijo organizacije s procesom segmentacije podatkov ter oceniti, koliko truda je treba vložiti za migracijo vseh podatkov v oblak. V tem koraku je smiselno vključiti omrežje za dostavo vsebin (Content Delivery Network – CDN), da bi še dodatno izboljšali zmogljivost aplikacij predvsem z vidika končnega uporabnika;

- **migracija aplikacije:** v drugem koraku faze izvedbe moramo odgovoriti na ključno vprašanje, kako premakniti ves poslovni sistem v oblak, ne da bi pri tem oškodovali oz. onemogočili trenutno poslovanje. Pri tem sta na voljo dve možnosti: ali premaknemo celotno aplikacijo v oblak ali se lotimo migracije postopno. Glede na klasifikacijo tipov aplikacij iz prve faze se odločimo, katero možnost bomo aplicirali za kateri tip aplikacije. V primeru aplikacije brez stanja ali šibko sklopljene aplikacije je prva možnost ustrežnejša. V primeru kompleksnejših aplikacij, kot so npr. bančniški sistemi, spletne trgovine ter drugi poslovno orientirani informacijski sistemi, je primernejši postopen migracijski pristop, ki mu z drugimi besedami rečemo hibridni pristop (slika 5) in zagotavlja delovanja aplikacije v t. i. fazi koeksistence. Ko je aplikacija uspešno prenesena, je v tem koraku faze izvedbe treba prenesti še druge logične komponente aplikacije, kot so distribuirani predpomnilnik ter sporočilne vrste;



Slika 5: Hibridni pristop migracije aplikacij v oblak

- **optimizacija:** v tem koraku je smiselno identificirati dodatne prednosti oblaka, ki jih morebiti še nismo upoštevali. Ravno tako je treba narediti dodatne analize in ugotoviti, kako lahko še dodatno zmanjšamo stroške, da bomo zagotovili avtomatizirano elastičnost naših kapacitet v oblaku. Čeprav večina ponudnikov računalniškega oblaka poskrbi za varnost infrastrukture in platform z uporabo dobrih praks in tehnologij, je odgovor-

nost za varnost na aplikacijski ravni popolnoma prepuščena uporabnikom oblaka. V koraku optimizacije je tudi treba ugotoviti, kako lahko izboljšamo varnost ter zanesljivost naših aplikacij in podatkov. Določitev metrik za merjenje kritičnih zmogljivosti aplikacije ter zagotovitev uporabe administrativnih orodij za upravljanje in vzdrževanje aplikacij je prav tako smiselno obravnavati v tem koraku.

Tabela 2: Primerjava predlaganega pristopa s pristopoma komercialnih produktov

Predlagani pristop	Štirifazni migracijski pristop (Microsoft)	Šestfazni migracijski pristop (Amazon)
<b>Faza raziskave</b>		
• Izbira ponudnika	-	-
• Analiza	+	-
• Načrtovanje	+	-
<b>Faza preizkusa</b>		
• Evalvacija	-	+
• Izdelava PoC	-	+
<b>Faza izvedbe</b>		
• Migracija podatkov	+	+
• Migracija aplikacije	+	+
• Optimizacija	-	+

## 5 SKLEP

Migracija obstoječih aplikacij v oblak omogoča različnim tipom organizacij in podjetjem, da lansirajo nove produkte ali storitve na trg hitreje in z bistveno manj stroški. Prehod v oblak podjetjem omogoča tudi hitrejše vodenje analize tržišča ter omejevanje izgub v primeru hitrega padca, če npr. produkt ali storitev ne doseže zelenih pričakovanj. Danes je vse več organizacij, ki selijo svojo programsko opremo in podatke v oblak z razlogom, da bi modernizirali svoja trenutna informacijska sredstva, zmanjšali stroške, izrabili številne prednosti, ki jih prinaša oblak – elastičnost, zanesljivost, visoka razpoložljivost, ali da bi se pripravili na potrebe, ki bi jih morebiti imeli v prihodnje. Fazni pristop, ki smo ga predlagali v prispevku, omogoča identifikacijo idealnih projektov za migracijo ter povečuje zaupanje in podporo k migraciji znotraj posameznega podjetja.

V prispevku smo opisali osnovne tipe migracije v oblak, identificirali ključne prednosti in slabosti migracij ter predstavili dve možnosti prenosa obstoječih aplikacij v oblak. V prvem primeru smo obravnavali migracijo tradicionalne aplikacije na platformo Windows Azure, medtem ko je drugi del opisoval prenos

obstoječe aplikacije v Amazonov oblak AWS. Opisali smo dva različna migracijska pristopa: štirifazni migracijski pristop, ki ga predlaga Microsoft, ter Amazonov šestfazni migracijski pristop. V nadaljevanju smo identificirali bistvene pomanjkljivosti posameznega pristopa. Ugotovili smo, da štirifazni migracijski pristop sicer uspešno prenese aplikacijo v oblak, vendar ima nekatere pomanjkljivosti, kot so premajhen poudarek na tehnični evalvaciji, evalvaciji stroškov, evalvaciji varnosti ter zakonodaji. Ta pristop ravno tako ne predpisuje izgradnje pilotne aplikacije kot zelo pomembnega člena migracije ter ne vpeljuje koncepta optimizacije. Šestfazni pristop odpravlja vse pomanjkljivosti, ki smo jih očiteli pristopu Microsofta, vendar na drugi strani daje premalo poudarka na analizi, strategiji in načrtovanju migracije. Na koncu smo predstavili nov uniformni pristop, ki uspešno rešuje identificirane slabosti komercialnih pristopov in ga je mogoče aplicirati pri migraciji večine aplikacij in podatkov ter je neodvisen od ponudnika računalništva v oblaku. V prihodnjem delu bomo predlagani model še dodatno rafinirali z upoštevanjem novih pristopov ter ga z migracijami resničnih informacijskih sistemov validirali na številnih komercialnih in odprtokodnih produktih.

## 6 VIRI IN LITERATURA

- [1] Jennings, R. (2009). *Cloud Computing with the Windows Azure Platform*. Wiley Publishing, Inc., Indianapolis.
- [2] Cunningham, R. S. (2010). *Windows Azure – Application Profile Guidance: Custom Web (Rapid Scaling Focus) Application Migration Scenario*. Logic 20/20, Inc.
- [3] HyperStratus (2009). *Migrating Applications to the Cloud – An Amazon Web Services Case Study (A HyperStratus White Paper)*. Dosegljivo na [www.hyperstratus.com](http://www.hyperstratus.com).
- [4] Varia, J. (2010). *Migrating your Existing Applications to the AWS Cloud – A Phase-driven Approach to Cloud Migration (Amazon AWS White Paper)*.
- [5] Cisco Systems, Inc. (2010). *Planning the Migration of Enterprise Applications to the Cloud – A Guide to Your Migration Options: Private and Public Clouds, Application Evaluation Criteria, and Application Migration Best Practices*.
- [6] Armbrust, M., Fox, A. et al. (2010). *Above the Clouds: A Berkeley View of Cloud Computing*. Electrical Engineering and Computer Sciences, University of California at Berkeley.
- [7] Kaisler, S. & Money, W. H. (2011). *Service Migration in a Cloud Architecture*. System Sciences (HICSS), 2011 44th Hawaii International Conference on, pp. 1–10, 4–7.
- [8] Buyya, R. et al. (2009). *Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility*. Future Generation Computer Systems, Volume 25, Issue 6, Pages 599–616.
- [9] Smith, P. (2009). *Cloud Computing – A Strategy Guide for Board Level Executives*. Kynetix Technology Group.
- [10] Lee, J. Y., Lee, J. W., Cheun, D. W. & Kim, S. D. (2009). *A Quality Model for Evaluating Software-as-a-Service in Cloud Computing*, Software Engineering Research, Management and Applications, 2009. SERA '09. 7th ACIS International Conference on, pp. 261–266, 2–4.
- [11] Borenstein, N. & Blake, J. (2011). *Cloud Computing Standards: Where's the Beef?*. Internet Computing, IEEE, vol.15, no.3, pp.74–78.
- [12] Kandukuri, B. R., Paturi, V. R., & Rakshit, A.. (2009). *Cloud Security Issues*. Services Computing, 2009. SCC '09. IEEE International Conference on, pp. 517–520, 21–25.
- [13] Alhamed, M., Dillon, T. & Chang, E. (2010). *Conceptual SLA framework for cloud computing*, Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on, pp. 606–610, 13–16.
- [14] Khajeh-Hosseini, A., Greenwood, D. & Sommerville, I. (2010). *Cloud Migration: A Case Study of Migrating an Enterprise IT System to IaaS*, Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, pp. 450–457, 5–10.
- [15] Andrieux, A. et al (2004). *Web services agreement specification (WSAgreement)*.

■

Robert Dukarič je bil po končanem študiju računalništva in informatike na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru zaposlen kot mladi raziskovalec iz gospodarstva v podjetju XLAB, d. o. o. Je tudi doktorski študent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Raziskovalno se ukvarja s področjem računalništva v oblaku in sodeluje v številnih raziskovalnih in aplikativnih projektih za industrijo.

■

Matjaž B. Jurič je redni profesor na Fakulteti za računalništvo in informatiko Univerze v Ljubljani, kjer vodi laboratorij za integracijo informacijskih sistemov. Je avtor oz. soavtor štirinajstih knjig. Sodeloval je pri številnih projektih doma in v tujini, med drugim tudi pri razvoju RMI-IIOP, sestavnega dela platforme Java 2, in je član BPEL Advisory Boarda. Leta 2007 je od SDA World Journal dobil nagrado za najboljšo knjigo s področja SOA, leta 2010 pa nagrado za najodmevnejši znanstveni članek s področja storitev iz NMS. Ima naziva Java Champion in Oracle ACE Director.