Arhitektura programiranja oblaka

Balatinac, Ivan

Master's thesis / Diplomski rad

2015

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:741219

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

ELEKTROTEHNIČKI FAKULTET

Sveučilišni Diplomski studij

ARHITEKTURA PROGRAMIRANJA OBLAKA

Diplomski rad

Ivan Balatinac

Osijek, 2015.

ELEKTROTEHNIČKI FAKULTET OSIJEK
Sveučilište Josipa Jurja Strossmayera u Osijeku

Obrazac D1: Obrazac za imenovanje Povjerenstva za obranu diplomskog rada

Osijek,24.08.2015

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za obranu diplomskog rada

Ime i prezime studenta:	Ivan Balatinac
Studij, smjer:	Diplomski studij računarstva, smjer procesno računarstvo
Mat. br. studenta, godina upisa:	D-537R, 2012.
Mentor:	Prof. Dr. Sci Ivica Crnković
Sumentor:	-
Predsjednik Povjerenstva:	Prof. Dr. Sci. Goran Martinović
Član Povjerenstva:	-
Naslov diplomskog rada:	Arhitektura programiranja oblaka
Primarna znanstvena grana rada:	Računarstvo
Sekundarna znanstvena grana (ili polje) rada:	Programsko inženjerstvo
Zadatak diplomskog rada:	Programiranje oblaka je jedan od najnovijih trendova u računarskoj znanosti. Kao nova tehnologija ne postoji sustavna informacija o njoj. Zadatak teme je dati pregled tehnologija programiranja oblaka, današnjih problema i izazova, te postojeća rješenja i smjerove daljnjeg razvoja.
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: Opće znanje iz računarske znanosti se uspješno koristilo Postignuti rezultati u odnosu na složenost zadatka: izvrsni Jasnoća pismenog izražavanja: jasno i precizno izražavanje Razina samostalnosti: Potpuna

Potpis sumentora:	Potpis mentora:

Dostaviti:

1. Studentska služba

U Osijeku, 24.08.2015. godine

Potpis predsjednika Odbora:

Sveučilište Josipa Jurja Strossmayera u Osijeku
ELEKTROTEHNIČKI FAKULTET OSIJEK

IZJAVA O ORIGINALNOSTI RADA

Osijek,
9.09.2015

Ime i prezime studenta:	Ivan Balatinac
Studij :	Diplomski studij računarstva
Mat. br. studenta, godina upisa:	D-537R, 2012.

Ovom izjavom izjavljujem da je rad pod nazivom**:**
Arhitektura programiranja oblaka

izrađen pod vodstvom mentora

Prof. Dr. Sci. Ivica Crnković

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 1

2. PROGRAMIRANJE U OBLAKU 3

2.1. Ključne karakteristike programiranja u oblaku 3

2.2. Modeli implementacije programiranja u oblaku 4

2.3. Modeli pružanja usluge programiranja u oblaku 5

2.4. Sudionici programiranja u oblaku 6

3. ARHITEKTURA PROGRAMIRANJA OBLAKA 8

3.1. Prednosti izgradnje aplikacija koristeći arhitekturu oblaka 8

3.2. Problemi s postojećim arhitekturama oblaka 9

3.3. Usporedba aktualnih arhitektura oblaka 10

4. PRIJEDLOG ARHITEKTURE OBLAKA 12

4.1. Prijedlog arhitekture oblaka iz različitih perspektiva sudionika 12

4.2. Kako dizajnirati arhitekturu oblaka – principi i razmatranja 17

4.3. Arhitektura privatnog oblaka 21

4.4. Arhitekture hibridnih oblaka 23

4.5. Arhitekture zajedničkog oblaka 25

4.6. Arhitekture javnih oblaka 26

4.7. Arhitektura različitih slojeva oblaka 27

4.8. Arhitektura autonomnog upravljanja oblakom 38

4.9. Dvoslojna nasuprot troslojnoj nasuprot višeslojnoj arhitekturi oblaka 43

4.10. Ostale arhitekturalne paradigme i rješenja za dizajniranje oblaka 47

5. EVOLUCIJA RAČUNALNOG OBLAKA, ŠTO SLIJEDI? 61

6. ZAKLJUČAK 62

LITERATURA 63
SAŽETAK 64
ABSTRACT 65
ŽIVOTOPIS 66
PRILOG – POPIS STUDIJA 67

1. UVOD

Programiranje u oblaku (*eng. Cloud Computing*) je postalo jedna od najvažnijih novih računalnih strategija u poduzećima. Najpoznatija definicija programiranja u oblaku je određena po Nacionalnom institutu za standarde i tehnologiju (NIST) [S72] kao: „Cloud computing (računarstvo u oblaku) je model za omogućavanje sveprisutnog, odgovarajućeg, mrežnog pristupa na zahtjev (on-demand) za dijeljenje konfigurabilnih računalnih resursa (npr., mreže, servera, spremišta podataka, aplikacija i servisa/usluga) koji se mogu brzo omogućiti i dodijeliti uz minimalan napor i interakciju sa davateljem usluge. Ovakav model oblaka se sastoji od pet osnovnih karakteristika, tri modela usluga, te četiri implementacijska modela.“

Programiranje u oblaku povećava integritet, povjerljivost, skalabilnost, i dostupnost za korisnike i poduzeća. Programiranje u oblaku omogućava optimiziranu i efikasnu računalnu platformu, i smanjuje trošak ulaganja u dodatni hardver i softver. Primjerice, kad se tvrtka Netflix počela naglo razvijati, i tradicionalni podatkovni centar nije zadovoljio sve potrebe, tvrtka je odlučila preseliti website i „streaming“ servis u računalni oblak. Ovaj postupak je omogućio porast kompanije i proširenje baze korisnika bez dodatne izgradnje i uzdržavanja većeg podatkovnog centra, koji bi bio potreban zbog razvojnih zahtjeva [1].

Uz brzi razvoj svakodnevnog života i naglo povećanje internet prometa, gospodarstvo i akademija su krenuli u potragu za pomoć pri održavanju usluga koje pružaju. Statistički podatci prikupljeni od znanstvenog časopisa „Your Digital Space“ [2] prikazuju da šaljemo gotovo 3 milijuna e-mailova u jednoj sekundi, u jednoj minuti prenosimo 20 sati videa na YouTube, Google obrađuje 24 petabajta podataka, 50 milijuna tweetova se objavlju dnevno, i gotovo 73 proizvoda se naručuju svake sekunde na Amazonu. Također, Facebook je 2012. doživio rast od 1382% u jednom mjesecu. Prema istraživanju IDC (International Data Corporation) provedenog 2013. godine [3], potrošnja na javne IT usluge u oblaku na globalnoj razini će doći do 47,4 milijarde dolara u 2013. godini, a očekuje se da će biti više od 107 milijarde dolara u 2017. Zbog svega navedenog, važno je odabrati odgovarajuću arhitekturu računalnog oblaka prema različitim potrebama tvrtke.

Ovaj se diplomski rad sastoji od šest poglavlja, uključujući uvod i zaključak. Prvo je dan pregled programiranja u oblaku, dana je njegova definicija, navedene su ključne karakteristike kao i

sudionici programiranja u oblaku. Treće poglavlje detaljnije ulazi u problematiku arhitekture oblaka i daje uvid u probleme s kojima se suočavaju dizajneri aplikacija za oblak, kao i korisnici oblaka. U četvrtom poglavlju je dan prijedlog postojećih arhitektura za rješavanje navedenih problema po različitim kategorijama. U petom poglavlju se navode buduće smjernice i predviđanja za oblak, te na kraju dolazi zaključak.

2. PROGRAMIRANJE U OBLAKU

Pojam računarstva u oblaku postoji od 1950-ih godina kada su „poslužiteljske sobe“ bile na raspolaganju školama i tvrtkama. Mnogo nezavisnih korisnika je moglo pristupiti „poslužiteljskim sobama“ sa terminala. Ovo poglavlje predstavlja ključne karakteristike, modele implementacije i poslovne modele, i sudionike računarstva u oblaku.

2.1. Ključne karakteristike programiranja u oblaku

Ključne karakteristike programiranja u oblaku su:

- Pružanje usluge na zahtjev korisnika (*eng. On-demand self-service*) – korisnik usluge može samostalno odabrati i pokrenuti računalne resurse ovisno o potrebi, bez potrebe za interakcijom s djelatnicima pružatelja usluge;
- Širok mrežni pristup koristeći standardne mehanizme (*eng. Broad network access*) – usluga mora biti dostupna putem mreže u bilo kojem trenutku, ukoliko politika tvrtke to dopušta. Uslugama se pristupa putem standardnih mehanizama koji promoviraju korištenje od strane raznovrsnih platforma (laptopa, stolnih računala, mobilnih telefona..);
- Udruživanje resursa i višezakupnički model (*eng. Resource pooling*) – dijeljenje resursa između više zakupljenih jedinica može povećati iskoristivost usluge, i smanjiti operacijske troškove. Primjeri resursa su memorija, procesne jedinice, spremišta podataka..;
- Brza elastičnost (*eng. Rapid Elasticity*) – mogućnost proporcionalnog povećanja ili smanjenja po potrebi potrošnje, nudi fleskibilnost dodjeljivanja resursa prema zahtjevima korisnika;
- Dostupnost resursa (*eng. Resource pooling*) – automatsko provjeravanje uporabe resursa, mjerenje usluga omogućava optimizaciju uporabe resursa;
- Mogućnosti provođenja revizije i certifikacije (*eng. Measured service*) – Korištenje resursa može biti nadgledano i kontrolirano te se mogu dobiti izvještaji o učinku korištenja resursa. Cilj provođenja revizije je transparentnost podataka o korištenju usluge kako za potrošače tako za davatelje usluge.

2.2. Modeli implementacije programiranja u oblaku

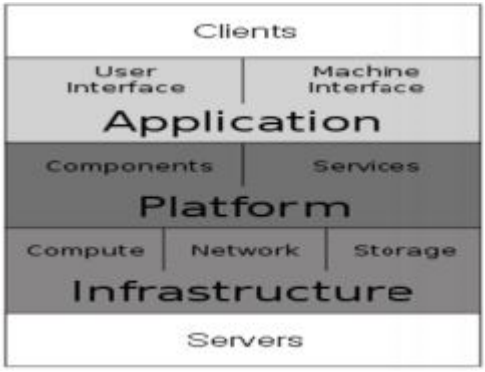
Postoje četiri osnovna modela implementacije računarstva u oblaku: javni oblak, privatni oblak, hibridni oblak, i zajednički oblak.

- Javni oblak *(eng. Public cloud)*: „Infrastruktura oblaka je dostupna za otvoreno korištenje javne publike ili poslovnih organizacija, i nalazi se u vlasništvu tvrtke koja daje u najam usluge u oblaku [S72]“. Korisnici plaćaju usluge po ovisnosti o vremenu i obujmu u kojem ih koriste (eng. *pay-per-use*), što pomaže u smanjenju troškova. Javni oblaci su manje sigurni u odnosu na ostale modele računalnih oblaka jer sve aplikacije i podaci imaju manju otpornost na zlonamjerne napade. Predloženo rješenje za ovaj problem je sigurnosna provjera sa obje strane.
- Privatni oblak (*eng. Private cloud*): „Infrastruktura oblaka se koristi isključivo za poduzeće. Infrastruktura može biti upravljana, korištena i u vlasništvu organizacije, treće strane i može postojati na lokaciji organizacije ili izvan [S72]“. To je središte podataka u vlasništvu pružatelja usluga u oblaku. Glavna prednost je lakše upravljanje sigurnosti, održavanja i nadogradnje te također pruža veću kontrolu nad implementacijom i korištenjem. U usporedbi s javnim oblakom gdje su svi resursi i aplikacije upravljani od strane pružatelja usluga, u privatnom oblaku usluge su sakupljene i dostupne korisnicima na organizacijskoj razini. Resursi i aplikacije su pod upravom same organizacije.
- Zajednički oblak (*eng. Community cloud*): „Infrastrukturu oblaka dijeli nekoliko organizacija i podržava određenu zajednicu korisnika koja ima zajednički interes (tj. misiju, sigurnosne zahtjeve, politiku, i usklađenja). Infrastruktura može biti upravljana, korištena i u vlasništvu organizacije ili više njih, treće strane i može postojati na lokaciji organizacije ili izvan nje [S72] “.
- Hibridni oblak (*eng. Hybrid cloud*): „Infrastruktura oblaka je sastav dva ili više oblaka (privatnih, zajedničkih ili javnih) koji ostaju jedinstveni entiteti, ali su međusobno vezani standardiziranom ili vlasničkom tehnologijom koja omogućuje prenosivost podataka i aplikacija (npr. razvoj oblaka za balansiranje opterećenja između oblaka) [S72]“. Hibridni

oblak je sigurniji način kontrole podataka i aplikacija, a omogućuje korisnicima pristup informacijama preko interneta.

2.3. Modeli pružanja usluge programiranja u oblaku

Postoji pet različitih slojeva u arhitekturi programiranja u oblaku: klijent, aplikacija, platforma, infrastruktura, i poslužitelj, prikazano na slici 2.1.



Slika 2.1. Slojevi računarstva u oblaku [4]

Model pružanja usluge predstavlja slojevitu apstrakciju na visokoj razini od glavne skupine usluga koje pruža model programiranja u oblaku, i kako su ti slojevi međusobno povezani [5].

Prvi sloj, klijent, se sastoji od računalnog hardvera i/ ili računalnog softvera koji se oslanja na programiranje u oblaku za isporuku aplikacija. Slojevi aplikacije, platforme i infrastrukture u oblaku, isporučuju modele usluga u oblaku.

- Softver kao usluga/SaaS (*eng. Software as a service*) poznat je kao usluga koja omogućuje krajnjem korisniku (potrošaču) pristup i korištenje usluga softverske aplikacije koji su u vlasništvu i upravi pružatelja usluge. Softver kao usluga omogućuje licenciranje softvera korisniku na zahtjev. Korisnik ne posjeduje softver nego ga iznajmljuje, najčešće za mjesečnu naknadu.
- Platforma kao usluga/PaaS (*eng. Platform as a Service*) je servis upravljan u oblaku i korisnici mu pristupaju putem web-preglednika. Platforma kao usluga je model pružanja usluga koje omogućuju stvaranje web aplikacija, bez potrebe za kupovinom i održavanjem

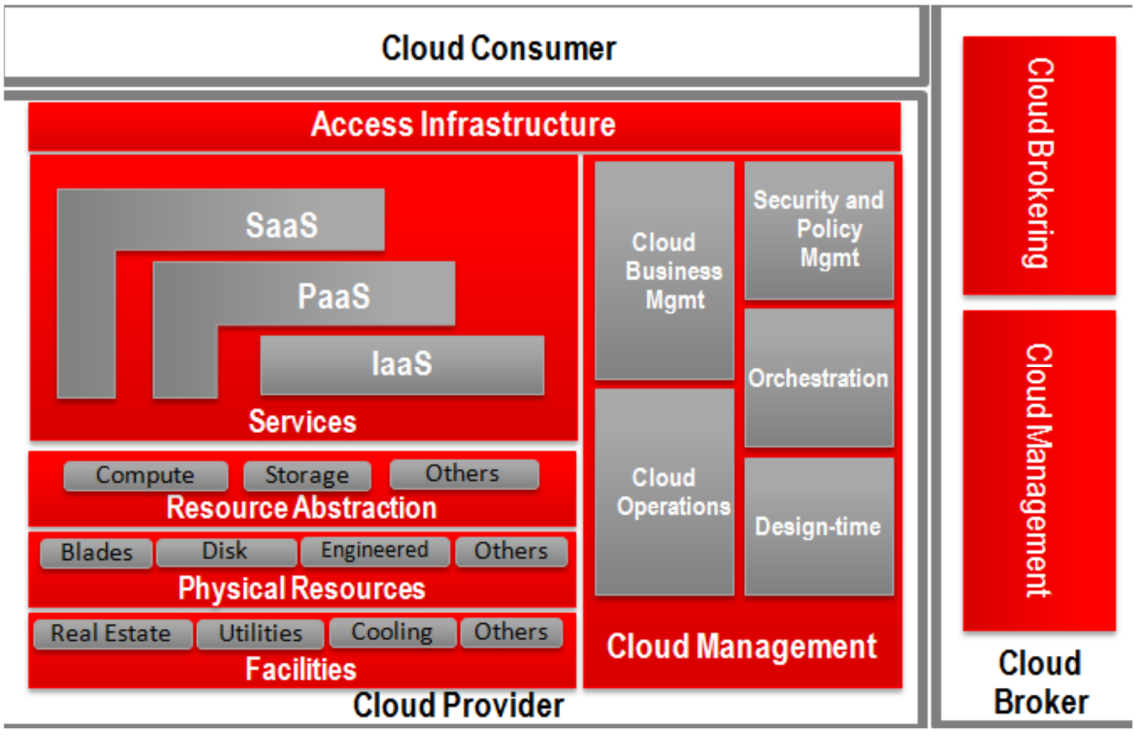
skupe infrastrukture poput hardvera i softvera. PaaS omogućuje korisnicima stvaranje softverskih aplikacija pomoću alata kojeg su dobili od pružatelja usluge.

- Infrastruktura kao usluga/IaaS (*eng. Infrastructure as a Service*) je usluga koja omogućuje korisniku iznajmljivanje infrastrukturne sposobnosti na zahtjev. Korisnici nemaju izravan pristup resursima, ali imaju mogućnost odabrati i konfigurirati resurse na temelju njihovih potreba [5].

Posljednji sloj je poslužitelj koji se sastoji od karakteristika hardvera i/ili softvera potrebnog za isporuku navedenih usluga [4].

2.4. Sudionici programiranja u oblaku

Na slici 2.2 prikazan je konceptualni pregled arhitekture računarstva u oblaku s tri glavna sudionika –poslužitelj, korisnik, i posrednik [6].



Slika 2.2. Konceptualan pregled arhitekture [6]

- Poslužitelj usluga (*eng. Cloud provider*) u oblaku je poduzeće ili pojedinac koji pruža korisnicima usluge i resurse programiranja u oblaku.
- Korisnik usluga *(eng. Cloud consumer*) u oblaku je poduzeće ili pojedinac koji koristi usluge koje daje pružatelj usluga u oblaku, izravno ili preko posrednika.
- Posrednik usluga (*eng. Cloud broker*) u oblaku je zagovornik između pružatelja i korisnika usluga u oblaku.

3. ARHITEKTURA PROGRAMIRANJA OBLAKA

U ovom poglavlju analizirani su problemi s aktualnim arhitekturama računalnog oblaka i predstavljene smjernice za izradu rješenja za arhitekturu oblaka. Također navode se prednosti izgradnje aplikacija koristeći arhitekturu oblaka. Također priložena je statistika usporedbe aktualnih poslužitelja oblaka vezano uz jesu li riješili sve probleme korisnika ili nisu.

3.1. Prednosti izgradnje aplikacija koristeći arhitekturu oblaka

- Smanjivanje vremena izvođenja i vremena odziva – „Korištenjem programiranja u oblaku korisnici imaju pristup aplikacijama koje im mogu ponuditi brzo vrijeme odziva, jer se korisnički zahtjev obrađuje na mnoštvu virtualnih strojeva. Izvođenje zadatka na virtualnom stroju može optimirati vrijeme odziva raspodjelom poslova na zahtjev korisnika. [7]„.
- Plaćanje infrastrukture unaprijed gotovo ne postoji
- Smanjivanje rizika – „U sve većem broju slučajeva davatelji usluga oblaka imaju veliku infrastrukturu koja može podnijeti rast zakupljenog prostora i rad individualnog korisnika, smanjujući tako financijske rizike kojima korisnici mogu biti podložni. [7]“.
- Infrastruktura koja funkcionira točno na vrijeme - korisnici mogu zakupljivati nove resurse kako im rastu potrebe, a ako u početku zakupe previše prostora arhitekture oblaka isto tako mogu i osloboditi resurse.
- Mali početni troškovi - Korištenjem arhitekture oblaka korisnici iznajmljuju infrastrukturu pa troškovi nisu veliki (pošto ju ne kupuju), a kapitalne investicije mogu čak biti jednake nuli.
- Učinkovitije korištenje resursa - učinkovitije upravljanje resursima u smislu da korisnici imaju mogućnost pristupa aplikacijama samo kada su im one potrebne, a nakon toga ih jednostavno mogu prestati koristiti.
- Troškovi na temelju uporabe –„ stil naplaćivanja troškova po uporabi omogućuje naplaćivanje samo onih infrastruktura koje su korištene. Korisnik nije odgovoran za cijelu infrastrukturu oblaka. [7]“.

3.2. Problemi s postojećim arhitekturama oblaka

Prema studiji [S16], postojeće arhitekture računalnog oblaka rješavaju neke probleme, kao što su migracija usluge, potpora više zakupničkog modela, principi arhitektura računalnog oblaka itd, ali postoje neki problemi koje aktualne arhitekture oblaka nisu uspjele riješiti. Ti problemi su [S16]:

- Korisnici su često vezani uz jednog pružatelja usluga. Drugim riječima, migracija aplikacije na drugi oblak je izuzetno otežana;
- Aktualne implementacije računalnog oblaka ne dopuštaju fleksibilnost za prilagodbu izbora uključenih resursa;
- Nedostatak sigurnosti i podrške kontrole pristupa – većina aktualnih arhitektura ne uzimaju u obzir sigurnost i upravljanje kontrole pristupa;
- Nedostatak zajedničke podrške – izgradnja skalabilne i višekratne arhitekture računalnog oblaka za podršku dijeljenja resursa se i dalje suočava s izazovima;

Osim navedenih problema, studija [S37] ističe jedan dodatni:

- Nedostatak fleksibilnosti za korisničko sučelje – okviri sastava korisničkog sučelja nisu integrirani u arhitekturu oblaka.

Prema studiji [S67], najvažnija pitanja tokom osmišljavanja arhitekture računalnog oblaka su umrežavanje i upravljanje podacima. Neki drugi problemi s aktualnim arhitekturama računalnog oblaka su upućene u studijama [S35, S58] s obzirom na skalabilnost, složenost razvoja, neuravnoteženo opterećenje i nedostatak dostupnosti. Primjeri takvih arhitektura su [S35]: Salesforce.com, Yahoo! PNUTS hostirana platforma koja poslužuje podatke, usluga Amazon DynamoDB i Bigtable Obitelj distribuirani sustav za pohranu. Studija [S31] izjavila je da aktualni pristupi koji omogućavaju dinamične infrastrukture računalnog oblaka u stvarnom vremenu su neadekvatni, skupi i nisu skalabilni da bi mogli podržati masovne zahtjeve korisnika.

3.3. Usporedba aktualnih arhitektura oblaka

Na slici 3.1. prikazana je statistika usporedbe aktualnih arhitektura oblaka u pogledu na navedene probleme vezane uz atribute kvalitete (jesu li oni riješeni ili ne). Crvena boja označava da problem nije riješen, dok zelena boja označava riješeni problem.

„IN: Intelligent Network, VRML: Virtual Resource Mediation Layer, FCAPS: Service Collaboration Network, V: Virtualization, SOC: Service-Oriented COmputing, ECP: Eucalyptus cloud platform, OCP: Opennebula cloud platform, ICC: Incoporative cloud computing, GOA: Google's Open Social API, EC2: Elastic Compute Cloud, ECP: Enterprise computing power, VMM: Virtual machine monitor, IVIC: Virtual Computing environment, NFS: Network File System, NetInf: Networks of Information, NLP: Non-Linear Programming model, RLE: Rationale of an altruistic economy, EM-Economy management, CM- Cost Model, TCO- Cloud Total Cost of Ownership, TCP/UDP: Transmission Control Protocol /User Datagram Protocol, RF: Risk Factor, CM: C-Meter, SOCCA: Service- Oriented Cloud Computing Architecture“ [S33]

Kao što je prikazano na slici 3.1., Elastični računalni oblak (EC2) iz Amazon Web Services nudi najbolju uslugu. Prema studiji [S70] EC2 prodaje 1,0 GHz x86 instanci oblaka za 10 centi po satu, a nova instanca može za dvije do pet minuta. Međutim, postoje istraživanja i eksperimenti koji su pokazali da je moguće razbiti sigurno okruženje tog oblaka, unatoč jakoj enkripcije [S47]. U idućem poglavlju rada bit će dan osvrt na probleme vezane uz atribute kvalitete i prijedlozi arhitektura za rješenje tih problema.

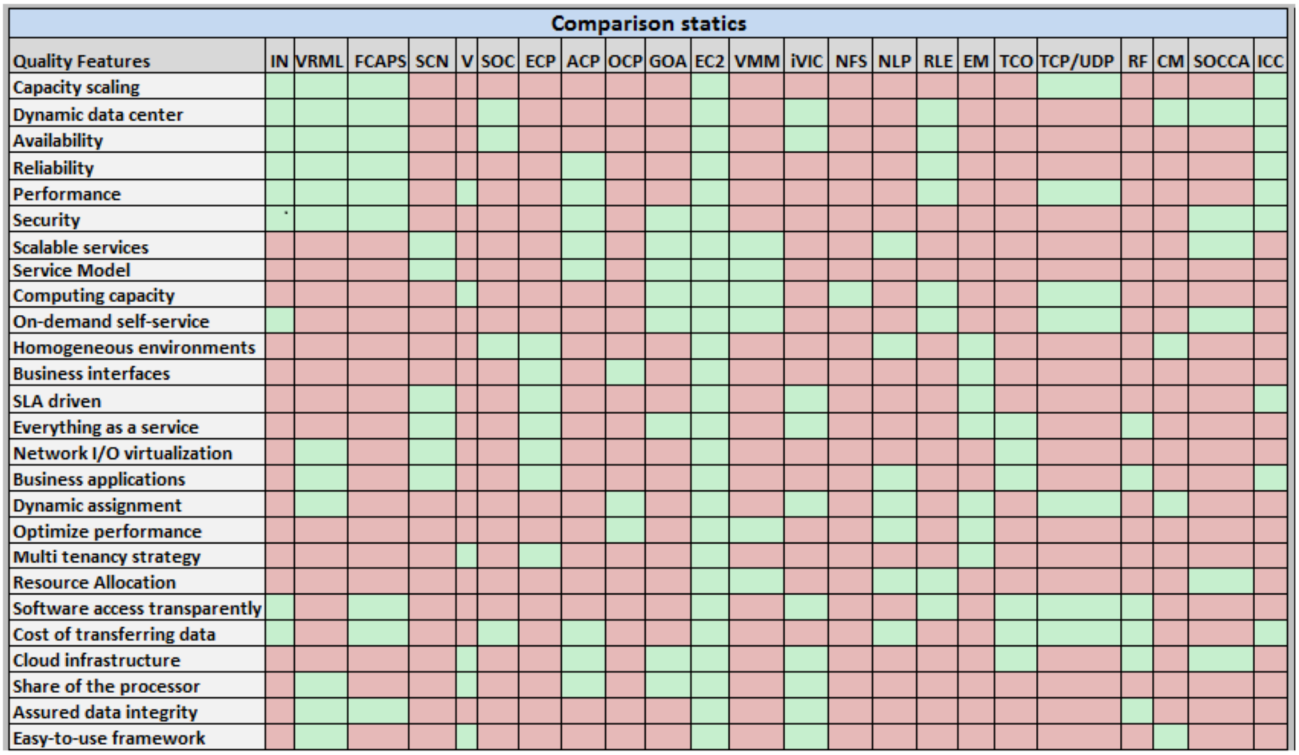


Figure 1. Comparison sketch of multiple expressions for cloud computing

Slika 3.1. Usporedna statistika brojnih atributa kvalitete u aktualnim arhitekturama oblaka

4. PRIJEDLOG ARHITEKTURE OBLAKA

U ovom poglavlju analizirani su problemi s aktualnim arhitekturama računalnog oblaka i predstavljene smjernice za izradu rješenja za arhitekturu oblaka. Nadalje, analizirane su predložene arhitekture u pročitanim studijama, razvrstane po modelu oblaka (privatni, zajednički, hibridni, intercloud i javni oblak) ili usluge (IaaS, PaaS, SaaS i XaaS).

4.1. Prijedlog arhitekture oblaka iz različitih perspektiva sudionika

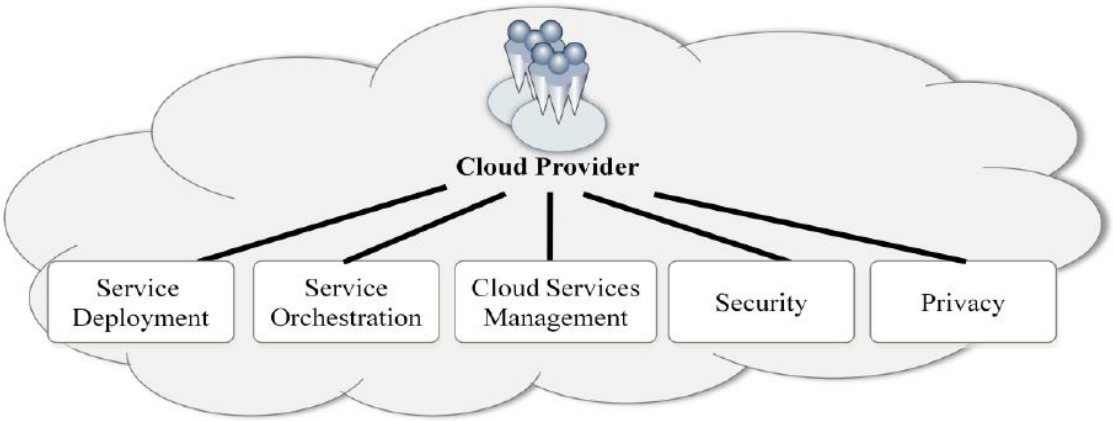
Kao što je opisano u drugom poglavlju, postoje tri glavna sudionika programiranja u oblaku: poslužitelj oblaka, korisnik oblaka i posrednik oblaka. U ovom poglavlju predstavljene su glavne aktivnosti, pitanja i prijedlozi arhitekture za svakog sudionika.

Poslužitelj oblaka

Postoji uzrečica da postati poslužitelj računalnog oblaka u smislu „izgradnje, te pokretanja takvog objekta je pothvat od sto milijuna dolara [S70], ali bez obzira na to, zbog brzog rasta interesa korisnika za korištenje nove tehnologije, mnoge velike tvrtke (Amazon, eBay , Google, Microsoft, itd) su postali poslužitelji oblaka.

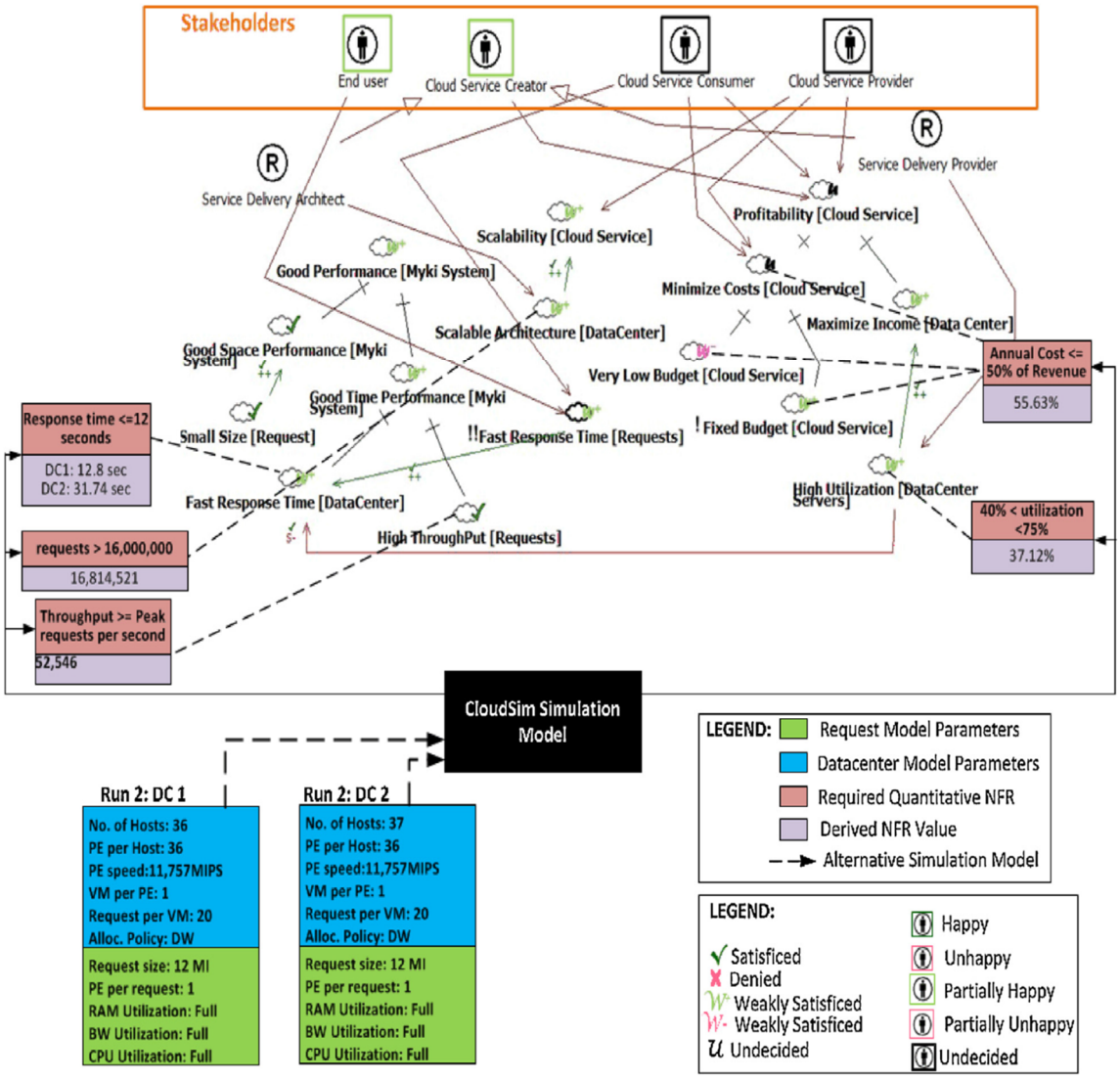
Aktivnosti i izazovi poslužitelja oblaka

Održavanje, nadzor, poslovanje i upravljanje su glavne aktivnosti poslužitelja oblaka [S40, S68, S72]. Studija [S72] definira aktivnosti (prema slici 4.1) poslužitelja oblaka u vezi s tri glavna sloja (IaaS, SaaS, PaaS). Prema toj studiji, za SaaS, poslužitelj oblaka postavlja, održava i ažurira rad softverskih aplikacija u oblaku. Za PaaS, glavne aktivnosti poslužitelja oblaka su upravljanje računalne infrastrukture za platformu i podrška procesa razvoja i upravljanja korisnicima PaaS oblaka. Za IaaS, poslužitelj oblaka pokreće softver u oblaku što je potrebno za uspostavljanje dostupnosti računalnim resursima korisnicima IaaS oblaka.



Slika 4.1. Glavne aktivnosti poslužitelja oblaka [S72]

Studija [S57] definira izazove poslužitelja oblaka kao kako postići dobru iskorištenost resursa oblaka, točno posredovanje i politiku raspodjele resursa. Studija predlaže usmjeren pristup simulacije prema cilju (CloudSim) za sustav u oblaku u kojem su registrirani ciljevi sudionika (*eng. Stakeholders*), zajedno s karakteristikama kao što su radni procesi, i koji su na kraju upotrijebljeni u izradi modela simulacije kao zamjena za arhitekturu oblaka (*myki*). Rezultati simulacije pokazuju da korištenje dva podatkovna centra (DC1 i DC2) ima veću mogućnost ispunjavanja ciljeva sudionika (prema slici 4.2). Obje studije [S57, S68] definiraju zadatke i izazove sudionika računalnog oblaka tokom osmišljavanja arhitekture računalnog oblaka.

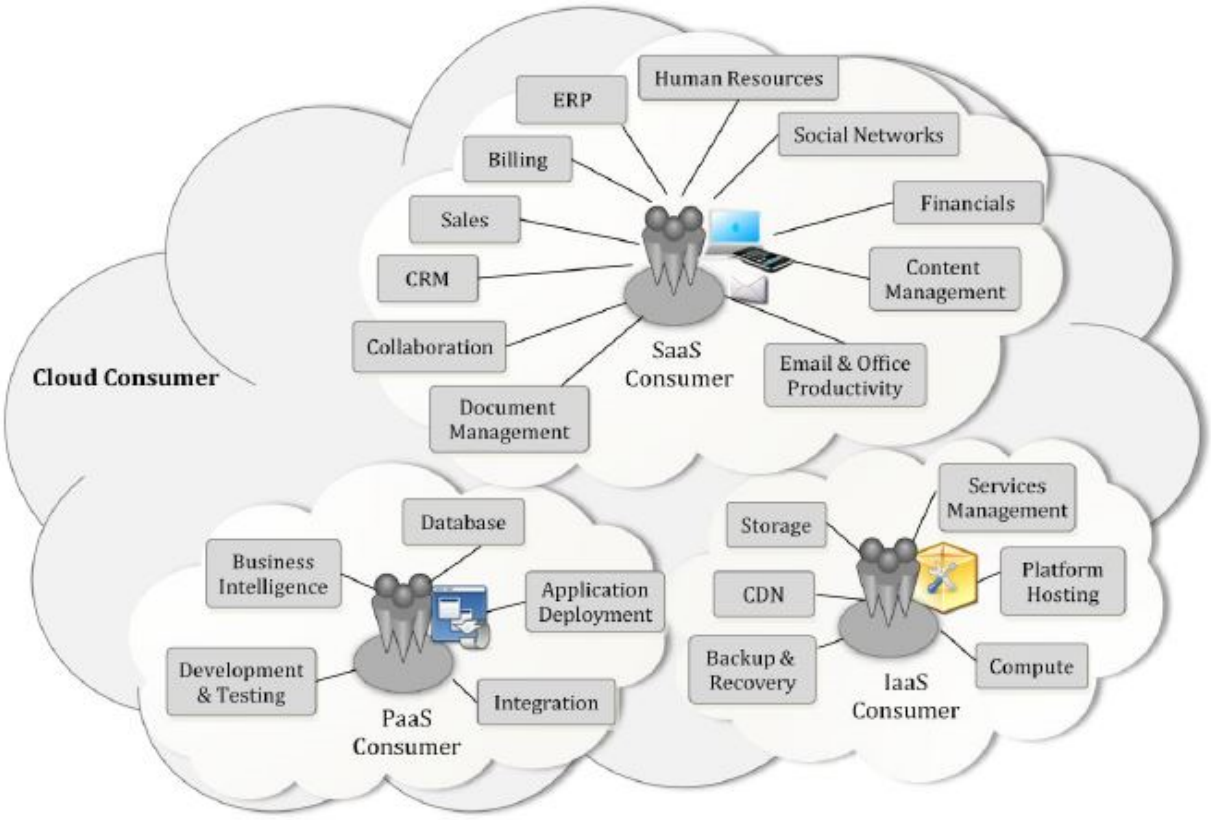


Slika 4.2. Predložena myki simulacija koji ispunjava ciljeve sudionika

Korisnik oblaka

Karakteristike javnih ili privatnih korisnika usluga u oblaku su samoposluga, standardni API za pristup uslugama u oblaku, brzo pružanje usluga i plaćanje po korištenju. Korisnici oblaka SaaS mogu biti poduzeća koja pružaju pristup softverskim aplikacijama, krajnji korisnici koji koriste program ili administratori softver aplikacija koji konfiguriraju aplikacije. Korisnici oblaka PaaS mogu biti programeri aplikacija koji osmišljavaju i provode softver aplikacije, testeri aplikacija koji isprobaju aplikacije, osobe zadužene za razvoj aplikacija koje objavljuju u oblaku, ili administratori

aplikacije koji konfiguriraju i prate performanse. Korisnici oblaka IaaS mogu biti programeri sustava i administratori sustava.



Slika 4.3. Osnovne zadaće korisnika oblaka [S72]

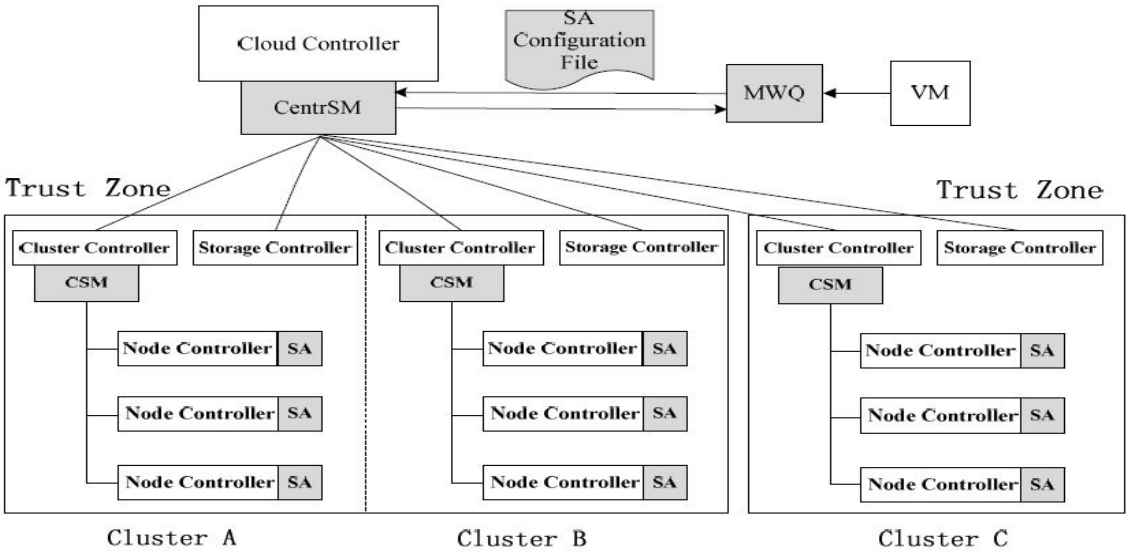
Problemi i rizici za korisnike oblaka (*eng. Cloud Consumers*)

Glavno pitanje za korisnike tijekom korištenja arhitekture oblaka je što ne znaju lokaciju podataka, pošto poslužitelji oblaka ne pružaju tu informaciju. Prema istraživanju [S45] korisnicima oblaka su najvažniji atributi kvalitete kao fleksibilnost, troškovi, opseg i performanse, sigurnost i usklađenost, pouzdanost i usklađenost, te upravljanje uslugama oblaka. Također su u istraživanju klasificirani glavni rizici za korisnike primjenom arhitekture računalnog oblaka, kategorizirani u pet glavnih kategorija:

- Operativni rizici – dostupnost, pouzdanost, integritet, prikladnost, održavanje;
- Potencijalne opasnosti – opstanak, veliki prekidi usluga, učestalost prekida i otpornost na prekide, kompatibilnost, fleksibilnost;

- Sigurnosni rizici – sigurnost servisa, sigurnost podataka, autentifikacija i autorizacija, osjetljivost na napade uskraćivanja usluga (*eng. Denial of service*);
- Poslovni rizici – cijene, korisnička podrška, prekršaj privatnosti, poštivanje propisa;

Pošto je platforma računalnog oblaka transparentna za korisnike koji uglavnom ne znaju gdje i na kojoj virtualizacijskoj platformi su njihovi virtualni strojevi sagrađeni, korisnici nikada neće doznati da su 'susjedi' njihovih virtualnih strojeva korišteni od strane napadača. Zbog ovog problema, studija [S55] predlaže Pouzdani oblak na temelju razine sigurnosti (TCSL); integriranu, osiguranu i pouzdanu arhitekturu (prema slici 4.9).



Slika 4.4 Pouzdani oblak temeljen na sigurnosnoj razini arhitekture (TCSL) [S55]

TCSL (*eng. Trusted Cloud based on Security Level Architecture*) arhitekura dijeli okruženje računalnog oblaka u različita područja s različitim razinama sigurnosti. Postoje četiri principa dizajna koji su zaslužni za osmišljavanje TCSL arhitekture [S55]:

- Jednostavnost – TCSL uključuje arhitekturu IaaS oblaka i nekoliko funkcionalnih modula kao što je pouzdana migracija modula;
- Izolacija – sigurnosne usluge pružaju sigurna područja s različitim razinama sigurnosti koji sprječavaju odgovarajuće sigurnosne probleme;
- Fleksibilnost – pruža opći okvir sa sigurnosnim odmakom. Arhitektura na temelju sigurnosti se može prilagoditi svakoj praktičnoj aplikaciji oblaka;

- Skalabilnost – usluge u oblaku mogu poboljšati razinu sigurnosti u skladu s eventualnim potrebama kupca, odgovarajući sigurnosni atributi mogu se stalno dodavati ili mijenjati.

4.2. Kako dizajnirati arhitekturu oblaka – principi i razmatranja

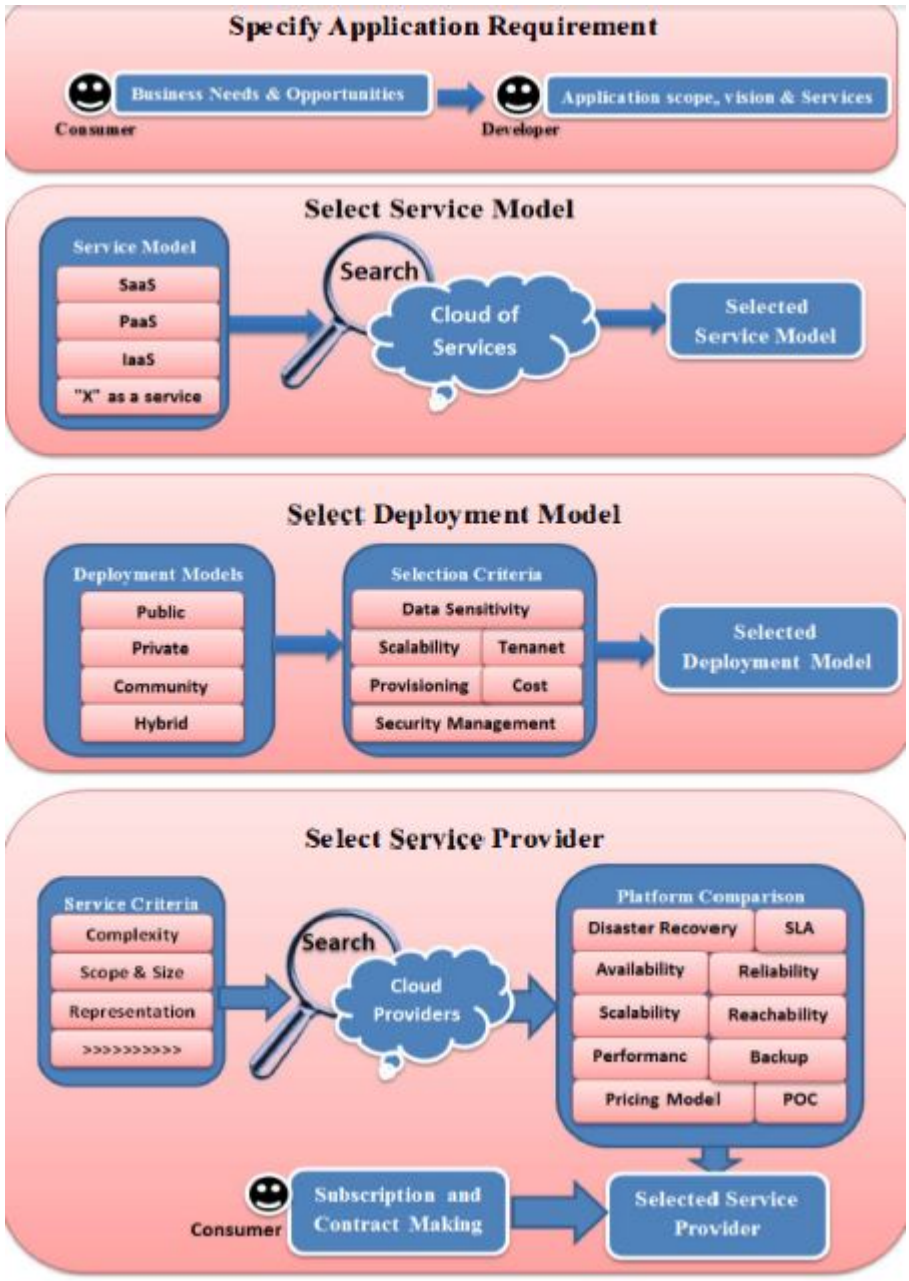
Za razliku od razvoja tradicionalnih web aplikacija, razvoj aplikacije za oblak ima fazu simulacije aplikacije na lokalnom računalnom okruženju i fazu postavljanja aplikacije u oblak u svrhu provjere prije prelaska u fazu proizvodnje. Studija [S38] klasificira faze životnog vijeka dizajniranja arhitekture oblaka na sljedeći način:

1. Dizajniranje aplikacije
2. Lokalna implementacija aplikacije
3. Lokalna simulacija aplikacije
4. Izvođenje aplikacije u računalni oblak
5. Režiranje aplikacije u računalnom oblaku
6. Korištenje aplikacije u računalnom oblaku

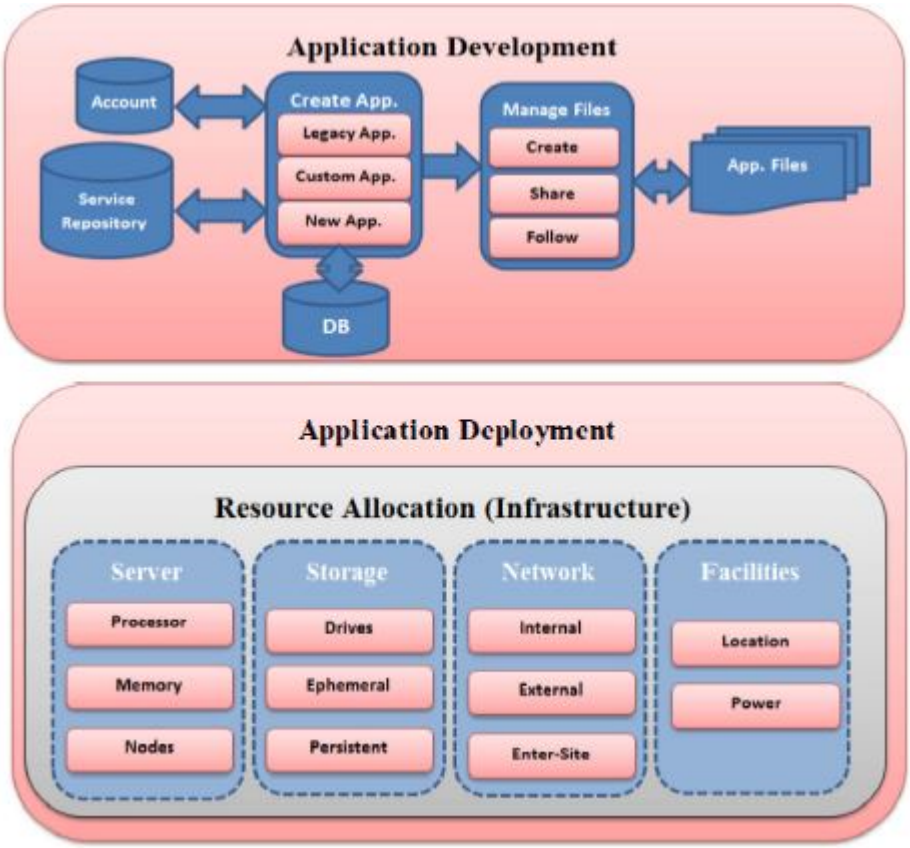
Platforme na strani klijenta pokrivaju prvu do četvrte faze, dok platforme na strani oblaka pokrivaju petu i posljednju fazu. Studija [S71] je dizajnirala jezik za osmišljavanje arhitekture aplikacija u oblaku. Predstavljena je nova arhitektura (*STAR*), koja pomaže programerima da razviju aplikacije za računalni oblak na sistematski način [S38]. Predložena arhitektura se sastoji od tri faze i osam slojeva (prema slikama 4.5, 4.6, 4.7):

- Faza zahtjeva/specifikacija– ova faza počinje s poslovnim zahtjevima primjene, a završava s odabranim modelom usluga aplikacije , pružateljem usluga i modela implementacije. Sastoji se od četiri sloja:
    - Specifikacija sloja zahtjeva aplikacije – Definiraju se sve usluge, poslovne potrebe i procjene modela određivanja cijene i daje se programerima.
    - Biranje sloja modela usluga – odabire se jedan ili više modela usluga ovisno o funkcijama, uslugama, sigurnosti i troškovima aplikacije.
    - Biranje sloja implementacije modela – odluka na kojoj će infrastrukturi oblaka aplikacija biti implementirana.

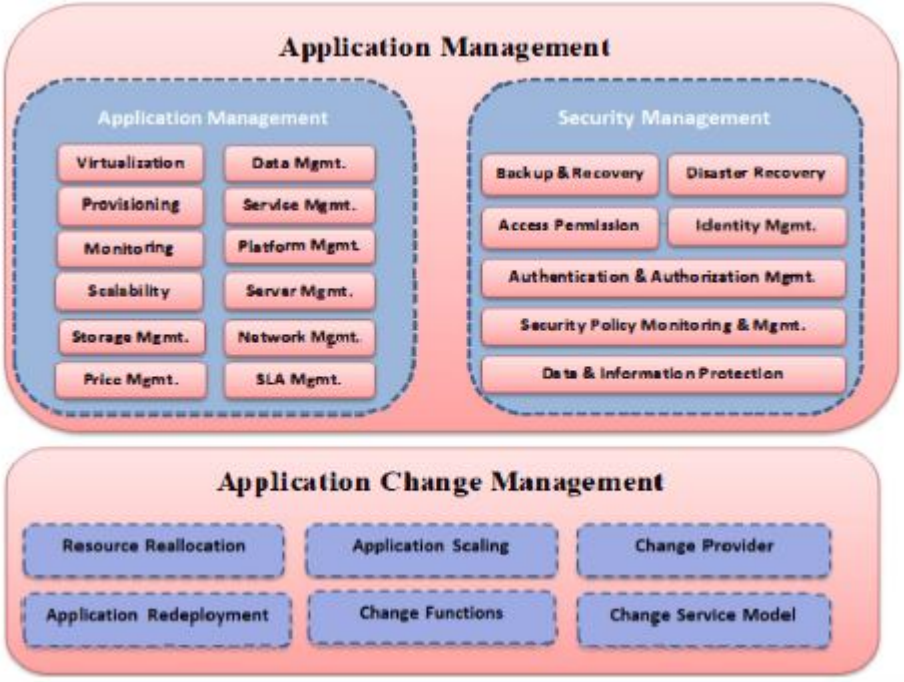
    - Biranje sloja pružatelja usluga – izbor poslužitelja usluga kako bi se ostvarile funkcije aplikacije.
- Faza razvoja i implementacije – kraj ove faze je pokretanje aplikacije u oblaku. Ona se sastoji od dva sloja:
    - Sloj razvoja aplikacije – aplikacija je razvijena putem upravljanjem datotekama koje mogu biti: nova datoteka, zajednička datoteka ili naknadna datoteka.
    - Sloj implementacije aplikacija – svi predmeti i datoteke aplikacije su objavljeni na odabranom poslužitelju usluga.
- Faza upravljanja i održavanja – uključuje sve komponente servisa vezanih uz potrebe za upravljanje razvijene aplikacije i njenih funkcija. Ona se sastoji od dva sloja:
    - Sloj upravljanja aplikacije – uključuje dva dijela: upravljanje aplikacijom i upravljanje sigurnošću.
    - Sloj upravljanja aplikacijskih promjena – obrađuje promjene zahtjeva aplikacija u oblaku.



Slika 4.5. Predložena STAR arhitektura; faza specifikacije / zahtjeva [S38]



Slika 4.6. Predložena STAR arhitektura; faza razvoja i implementacije [S38]



Slika 4.7. Predložena STAR arhitektura; faza upravljanja i održavanja [S38]

Kad se aplikacija prebacuje u oblak, mora se razmisliti o nekoliko ključnih faktora vezanih uz dizajn kako bi se omogućila elastičnost i dinamično korištenje: pozicija usluga, interoperabilnost i dizajn usmjeren prema informaciji. Šest najvažnijih preporučenih pravila vezanih uz sigurnost za dizajnere, arhitekte, programere i testere sustava za oblak, po studiji [S3] su: minimizirati osobne podatke poslane i pohranjene u oblaku, zaštititi osobne podatke u oblaku, povećati kontrolu korisnika, omogućiti korisniku izbor, odrediti i ograničiti svrhu korištenja podataka, pružiti povratnu informaciju.

Novi naraštaj usluge

Studija [S47] identificira šest važnih aspekta životnog ciklusa *novog naraštaja usluge* koji su: dizajn, inženjering, implementacija, mjerenje korištenja usluge, podrška i održavanje, te iskustvo. Novi naraštaj servisa se definira kao transformirana tradicionalna usluga koju nudi tvrtka IT usluga, koje će biti mjerljive, vrlo pristupačne nudeći izuzetno odmjereno korištenje. Studija [S68] daje pregled scenarija (stvaranje usluge u oblaku i objavljivanje u katalogu usluga, traženje/korištenje usluge, i upravljanje okruženjem oblaka) koji su uključeni u bilo kojoj provedbi oblaka i kako arhitektura pruža odgovarajuće mogućnosti za rješavanje tih scenarija.

4.3. Arhitektura privatnog oblaka

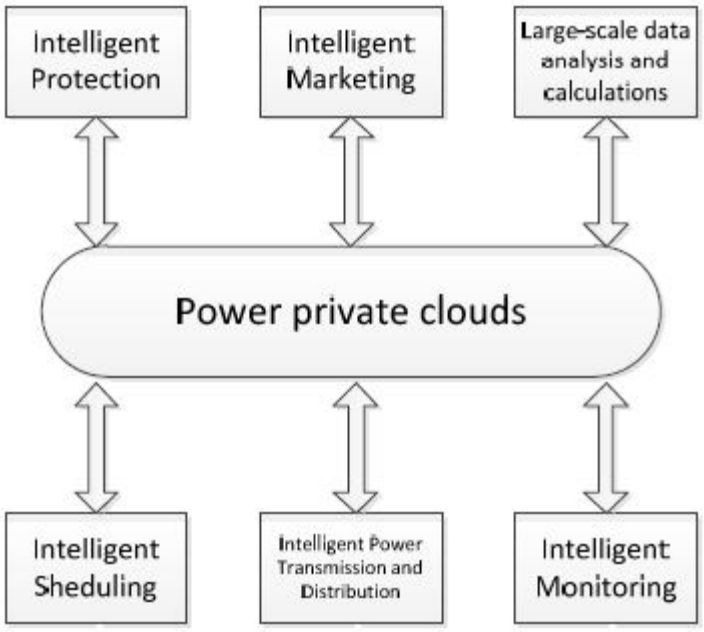
Izazovi i problemi

Prema istraživanjima najveći problemi privatnih oblaka su heterogenost mreže i ograničenja mobilnih uređaja. Nadalje, studija [S34] uključuje glavne komponente za pružanje kvalitete usluge u svojem predloženom modelu, kao što su: praćenje, balansiranje opterećenja, upravljanje prometom i sigurnost. Korisnici privatnih oblaka se slažu da treba povećati fleksibilnost podataka i pouzdanost. Također jedan od velikih značajki je potreba za što većom uštedom prilikom korištenja privatnih oblaka.

Predložena rješenja

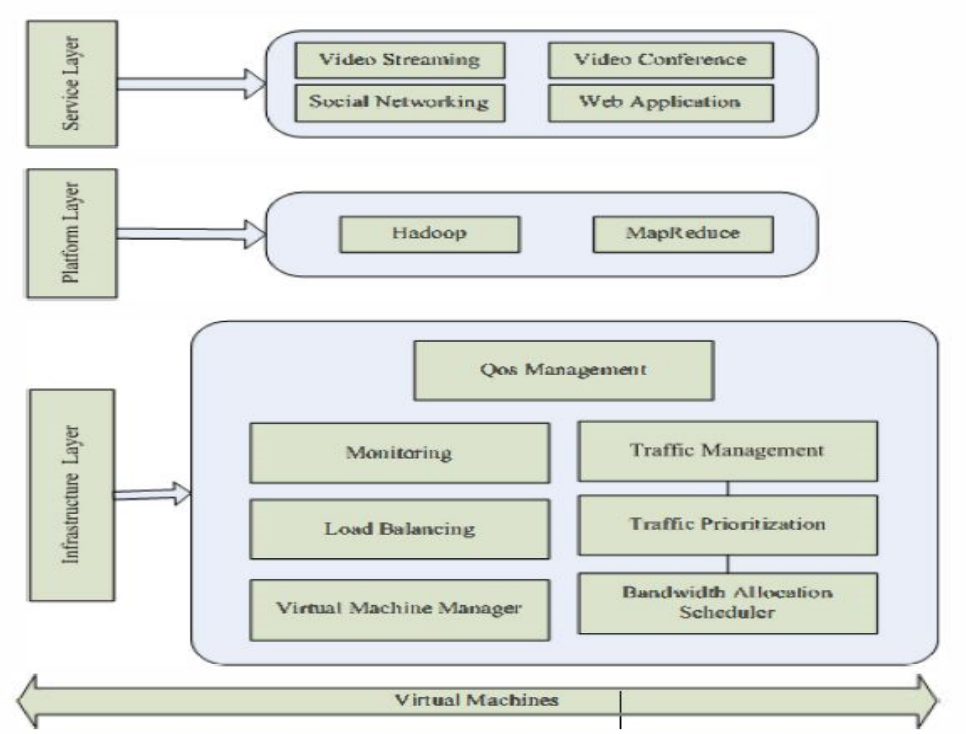
„Budući da privatni oblak omogućava poduzećima da sačuvaju svoju postojeću IT infrastrukturu i osiguraju fleksibilnost i skalabilnost, poduzeća mogu imati vlastite privatne oblake za povećanje

učinkovitosti. Tvrtke koje su predložile svoja rješenja privatnog oblaka su: *vCloud od VMware, ECI Datacenter od Microsofta, Virtualized Multi-Tenant Data Centre od Cisco-a* [S34]. Studija [S57] je usmjerena na model implementacije privatnog oblaka, u kojem jedna osoba (stvaratelj oblaka) posjeduje i upravlja podatkovnim centrom, ali daje zajednički pristup drugim osobama koji su se pretplatili na osnovi plaćanja po upotrebi. Izgradnja sustava računalnog oblaka ili mrežnog sustava može povećati integraciju podataka resursa i računalne snage elektroenergetskog sustava. U kombinaciji s karakteristikama elektroenergetskog sustava (nakupljanje velikih količina podataka) istraživanja u studiji [S56] predlažu privatni oblak (prema slici 4.8) elektroenergetskog sustava (*eng. Electric power private cloud*).



Slika 4.8. Model privatnog oblaka [S56]

Pohrana podataka u predloženoj arhitekturi je servisno orijentirana i sastoji se od pet slojeva: sloj servisa za pohranu, sloj upravljanja podacima, transportni sloj, sloj poslužitelja aplikacija i sloj pristupa. Sloj poslužitelja aplikacija je najfleksibilniji dio predloženog modela oblaka i to uključuje tehnologije, kao što su sigurnost, oporavak i tehnologije za povrat podataka. U studiji [S34], prikazana je arhitektura za izgradnju poduzetničkog privatnog oblaka za multimedijalne usluge (prema slici 4.9). Kvaliteta upravljanja usluge se provodi u sloju infrastrukture. Korisnici mogu sustavu dodavati nove aplikacije i dijeliti resurse.



Slika 4.9. Predložena arhitektura multimedijskog računalnog oblaka [S34]

4.4. Arhitekture hibridnih oblaka

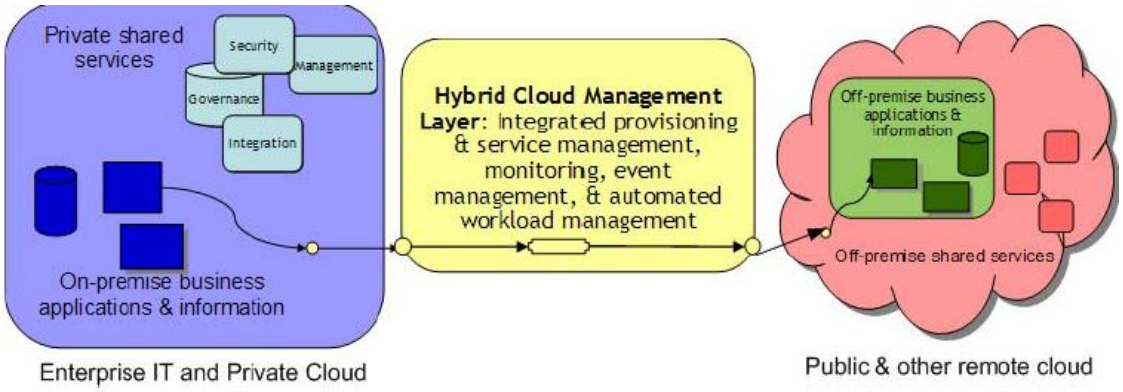
Izazovi i problemi

Glavni probleme dizajniranja arhitekture hibridnih oblaka su fleksibilnost i skalabilnost. Studija [S44] je sa svojim predloženim rješenjem predstavila riešenje na pitanja interoperabilnosti, proširivosti, jedinstvenih i centraliziranih zahtjeva upravljanja. Studija [S60] se bavi pitanjima prihvaćanja rješenja za oblak u nerazvijenim zemljama. Također jedni od problema koji nisu riješeni su: sigurnost podataka, izolacija, integrirani nadzor i upravljanje događanjima okruženja hibridnih oblaka. Potrebno je napraviti identifikaciju najbolje prakse za rješavanje pedagoških, tehničkih, političkih i ekonomskih pitanja za prihvaćanje predložene infrastrukture oblaka.

Predložena rješenja

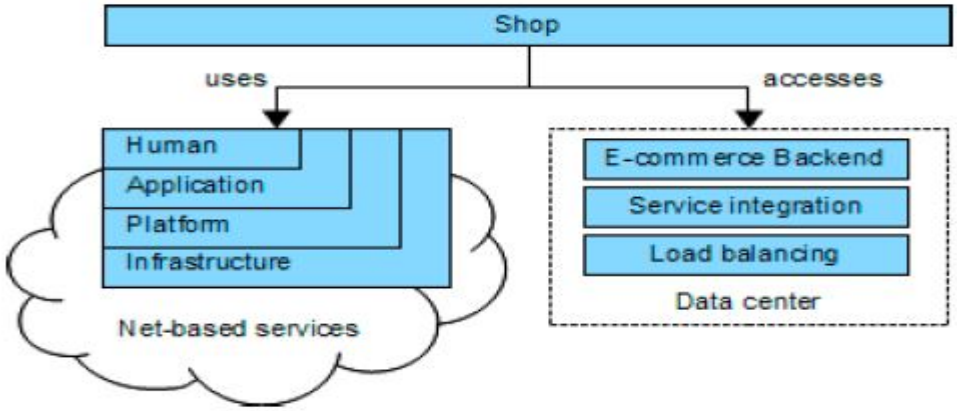
Hibridni oblak doživljava sve veću pozornost; ali kako bi se ostvario puni potencijal hibridnog oblaka, potreban je okvir arhitekture za učinkoviti spoj javnih i privatnih oblaka. Studija [S59] predlaže fleksibilnu i skalabilnu arhitekturu hibridnog oblaka zajedno sa politikom osviještenosti za

pogreške (*eng. Failure aware policies).* Studija [S44] opisuje podršku za sva tri modela integracije (rezerviranje, praćenje i integracija podataka) u okvir hibridnog oblaka (prema slici 4.10). Ključno obilježje te arhitekture je integracija platforme koja omogućuje razvoj i implementaciju specifičnih funkcija komponenti integracije na i mogućnost kontrole integracije pomoću određenih zakonitosti.



Slika 4.10. Predložena arhitektura hibridnog oblaka [S44]

Studija [S60] predlaže arhitekturu hibridnog oblaka (prema slici 4.11) za web trgovine sa gledišta trgovaca.



Slika 4.11. Integracija različitih vrsta usluga u web trgovini koristeći oblak aplikaciju [S60]

Studija [S60] integrira usluge u web trgovini. Podatakovno orijentiran katalog komponenta se distribuira u online podatakovnm centrima da bi dao podršku skaliranja aplikacije u vršnim situacijama (*eng. Peak situations*). Hibridna arhitektura kombinira model stabilnog razvoja i model

implementacije jezgre trgovine s brzim razvojem aplikacija i implementacije internetske platforme. Ovaj model također omogućuje testiranje prihvaćanja eksperimentalnih značajki prije integracije u jezgru sustava.

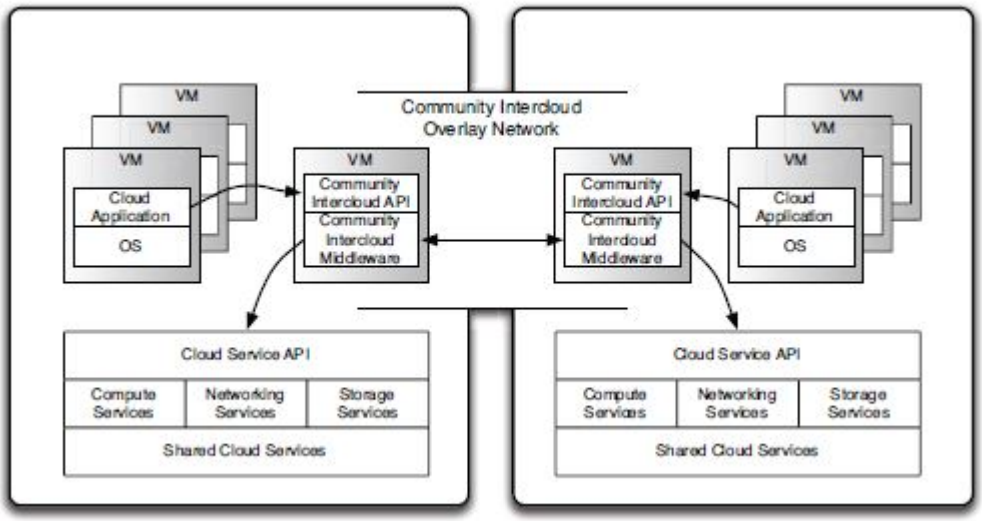
4.5. Arhitekture zajedničkog oblaka

Izazovi i problemi

Prema istraživanju [S9] najveći izazov s kojim se dizajneri arhitekture zajedničkog oblaka susreću je problem sa skalabilnosti i dostupnosti.

Predložena rješenja

Zajednički oblak koristi grupa korisnika oblaka sa zajedničkim problemima, kao što su ciljevi misije, sigurnost, privatnost i poštivanje politike, a ne služi samo jednoj organizaciji kao privatni oblak. Studija [S9] uvodi izraz *Zajednički Intercloud* (prema slici 4.12). Koncept Zajedničkog Intercloud-a (*eng. Community Intercloud*) je veliki distribuirani sustav koji povezuje oblake iz različitih administrativnih područja; problemi takvih sustava su da postoji mogućnost da dijelovi sustava mogu privremeno biti nedostupni. Studija [S9] predlaže arhitekturu koja osigurava korisnicima oblaka da će čak i ako su dijelovi sustava nisu dostupni ostali dijelovi sustava raditi ispravno.



Slika 4.12. Predložena arhitektura zajedničkog InterCloud-a [S9]

Aplikacije oblaka su izvedene u oba oblaka. Međusloj (*eng. Middleware*) djeluje kao sučelje između aplikacija oblaka.

4.6. Arhitekture javnih oblaka

Izazovi i problemi

Prema istraživanju [S10] najveći problemi vezani uz dizajniranje arhitekture javnih oblaka su pitanja izolacije, sigurnosti i zaštite podataka. Također, nastoje se ukloniti ograničenja potrebe za migracijom na model programiranja oblaka.

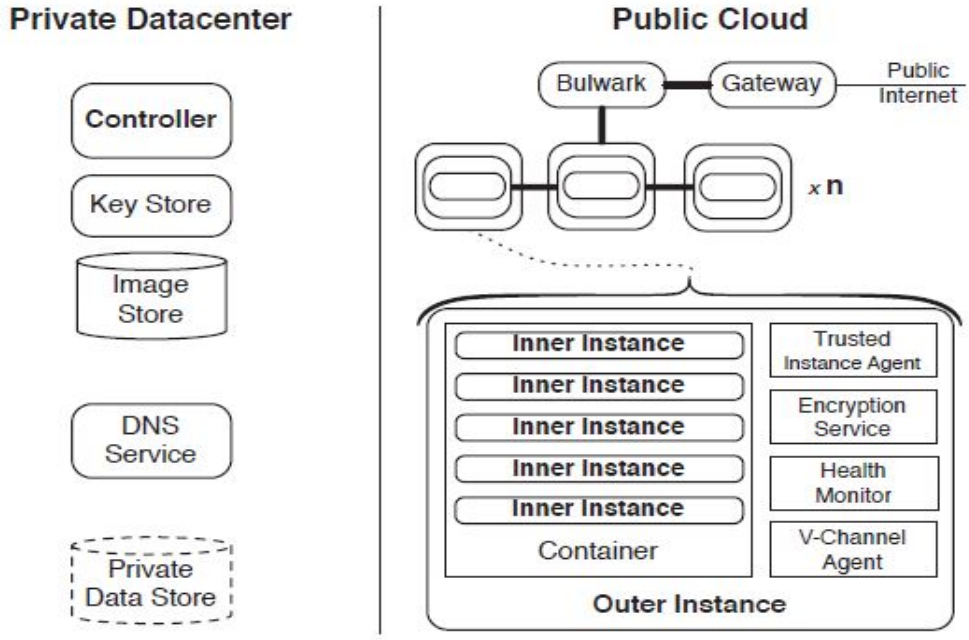
Predložena rješenja

Poslužitelji usluga oblaka grade javne oblake kako bi mogli ponuditi sigurnu, višekorisničku, plati-po-korištenju IT infrastrukturu na zahtjev poduzeća i vladinih agencija koje koriste usluge u oblaku, i koje žele riješiti, ili povećati, početne resurse korištenjem infrastrukture javnog oblaka. Poduzeća koja se prebacuju na infrastrukture javnog oblaka suočavaju se sa potencijalnim preprekama kontrole i sigurnosti. Obzirom na to, mora se steći novi skup najboljih praksi razvoja i implementacije infrastrukture oblaka. Studija [S10] predlaže referentnu arhitekturu (AERIE) za virtualni privatni oblak (prema slici 4.13) izgrađen na križanju usluga koji povećava kontrolu i izolaciju, poboljšava sigurnost i zaštitu podataka.

Referentna arhitektura AERIE definira dvije vrste komponenti: komponente sadržane na vanjskom instanci i komponente za potporu koje upravljaju vanjskim instancama i potpornim čvorovima mreže. Vanjska instanca je pokrenuta od strane regulatora koristeći API koji pružen od strane poslužitelja usluga oblaka. Referentna arhitektura AERIE postiže [S10]:

- Izolaciju – kako bi se poboljšala izolacija između vanjske i unutarnje instance, unutarnja instanca može pokrenuti drugi operativni sustav;
- Sigurnost mreže – povezivanje svake ugniježdene instance preko virtualnog kanala omogućuje usmjeravanje i šifriranje svog prometa između aplikacije i baze podataka koje možda nisu šifrirane u trenutku objavljivanja na privatni podatkovni centar;

- Povjerljivost – problem je riješen pohranjivanjem osjetljivih podataka u privatni oblak, dostupan samo virtualnim kanalom koji već postoji na implementiranoj aplikaciji;
- Integritet – ključevi i certifikati potrebni za ulazak u topologiju se daju samo instanci koja radi na nepromijenjenoj slici, što pomaže u zaštiti vanjskih instanca;
- Dostupnost – osigurana je implementacijom u više podatkovnih centara javnih oblaka i pomoću algoritma kružnoga dodjeljivanja DNS za učitavanje ravnoteže (*eng. Load balance*).



Slika 4.13. Logičke komponente referentne arhitekture AERIE [S10]

4.7. Arhitektura različitih slojeva oblaka

IaaS

Izazovi i problemi

Prema istraživanju [S12], u ovom trenutku, razni poslužitelji IaaS oblaka ne podržavaju usluge kao što su auto-skaliranje i napredni nadzor. Potrebno je poboljšati metodu modeliranja servisno orijentiranih usluga i alat za modeliranje i predviđanje elastičnosti aplikacija IaaS oblaka. Također, u pojedinim istraživanjima glavna motivacija je istraga kako vlasnici upravljaju troškovima njihovih aplikacija (implementiranih u IaaS oblacima) zadržavajući kvalitetu usluga koje pružaju. Članak

[S66] obraća pažnju na mnoga pitanja arhitekture IaaS oblaka, kao što su upravljanje mrežom, federacije podataka u oblaku, performanse i složenosti virtualizacije itd. Nadalje, potrebno je poboljšati fleksibilnost tijekom dizajniranja arhitekture za IaaS oblak.

Predložena rješenja

Neki od javnih IaaS poslužitelja su: *Amazon Web Services, CloudSigma, GoGrid, SliceHost, Storm* itd. Istraživanje [S12] uzima u obzir 17 javnih IaaS poslužitelja i njihove karakteristike (prema slici 4.14). Identificirano je sedam glavnih značajki koje karakteriziraju usluge koje nude poslužitelji IaaS usluga:

- Prilagodba modela – cijeli model (krajnji korisnik može odrediti i mijenjati tijekom izvršavanja aplikacije količinu procesorske snage, memorije i prostor za pohranu svakog virtualnog stroja), te djelomični model (ograničen skup predefiniranih virtualnih strojeva. Instanca virtualnog stroja se ne može mijenjati);
- Model naplate – kako se korištenje virtualnih strojeva naplaćuje od strane IaaS poslužitelja;
- Tip sučelja;
- Balansiranje opterećenja – sposobnost distribucije ulaznog opterećenja među različitim virtualnim strojevima sljedeći zakonitosti balansiranja;
- Ugovor o usluzi (*eng. Service Level Agreement*);
- Usluge nadzora – mehanizmi koji omogućuju korisnicima da prate rad sustava;
- Auto-skaliranje usluge – sposobnost automatskog dodavanja ili uklanjanja virtualnih strojeva na temelju praćenja performansi.

Provider	Customization model	Billing model	Interface type	Load Balancing	SLA (availability)
Amazon Web Services (aws.amazon.com)	Partial	1 hour	SSH + GUI + API	yes (LL)	99.95%
AT&T Synaptic (synaptic.att.com)	Full	1 hour	SSH + GUI + API	yes	99.9%
CloudSigma (cloudsigma.com)	Full	5 minutes	SSH + GUI + API	no	100%
ElasticHosts (elastichosts.com)	Full	1 hour	SSH +GUI + API	no	100%
FlexiScale (flexiant.com)	Full	1 hour	SSH + GUI + API	no	100%
GoGrid (gogrid.com)	Partial	1 hour	SSH + GUI + API	yes (RR, LL)	100%
JoyentCloud (joyentcloud.com)	Partial	1 month	SSH + GUI + API	yes	100%
Layeredtech (layeredtech.com)	Full	1 month	SSH + GUI + API	no	100%
Locaweb (locaweb.com.br)	Partial	1 month	SSH	no	99.9%
Opsource (opsource.net)	Full	1 hour	SSH + GUI + API	yes	100%
Rackspace (rackspacecloud.com)	Partial	1 hour	SSH + GUI + API	no	100%
ReliaCloud (reliacloud.com)	Partial	1 hour	SSH + GUI + API	yes (RR, LL, SI)	100%
RSAWEB (rsaweb.co.za)	Partial	1 month	SSH + GUI	no	ND
SliceHost (slicehost.com)	Partial	1 month	SSH+GUI + API	no	ND
Storm (stormondemand.com)	Partial	1 hour	SSH + GUI + API	yes (RR, LL, HI)	100%
Terremark (vcloudexpress.terremark.com)	Partial	1 hour	SSH + GUI + API	yes (LL)	100%
VPSNET (vps.net)	Partial	1 minute	SSH + GUI + API	no	100%

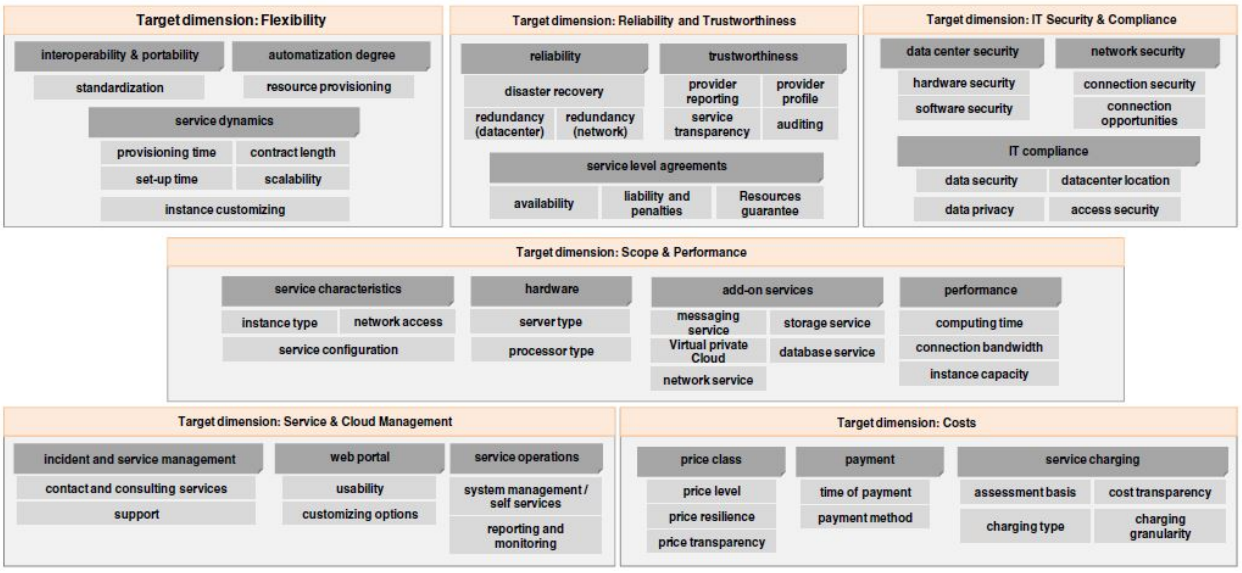
Slika 4.14. Obilježja IaaS poslužitelja [S12]

Rezultati istraživanja su pokazali da nijedan IaaS poslužitelj ne omogućuje sve usluge potrebne za provedbu autonomnih rješenja za upravljanje uslugama. IaaS omogućava pristup infrastrukturama računala, pohrane i mreže održavane u podatkovnom centru IaaS poslužitelja. Korisnik ne mora održavati servere sam. Poduzeća koriste IaaS model za izgradnju privatnih oblaka.

Prema istraživanju [S19], prednosti IaaS su:

- Smanjeni troškovi kupnje jer resursi već postoje, a krajnji korisnik plaća samo resurse koje koristi;
- Naplata po korištenju i bilo koje druga kombinacija korištenja usluge;
- Smanjeni utjecaj na okoliš od resursa koji se koriste na lokalnom računalnom centru.

Studija [S45] predlaže klasifikacijski okvir (*eng. Classification framework*) nezavisan od poslužitelja usluga za IaaS, koji se može koristiti u e-Vladi (prema slici 4.15), zbog nedostatka mogućnosti za usporedbu i klasifikaciju poslužitelja usluga u oblaku.



Slika 4.15. Okvir klasifikacije za IaaS [S45]

Predloženi okvir može pomoći tvrtkama u procesu odabira i stvara veću transparentnost u tržištu oblaka. Podijeljen je u šest ciljnih dimenzija: fleksibilnost, pouzdanost i povjerenje, IT sigurnost i sukladnost, opseg i učinkovitost, upravljanje uslugama i oblakom, te troškovi.

PaaS

Izazovi i problemi

Najveći izazovi za PaaS su vezani uz toleranciju grešaka, konfiguraciju, troškove, performanse, sigurnost i interakciju s IaaS slojem. Studija [S54] se bavi izazovima PaaS-a kao što su sigurnost i skalabilnost. Potrebno je napraviti optimiziranje sustava kontrolne točke i poboljšati performanse i sigurnost.

Predložena rješenja

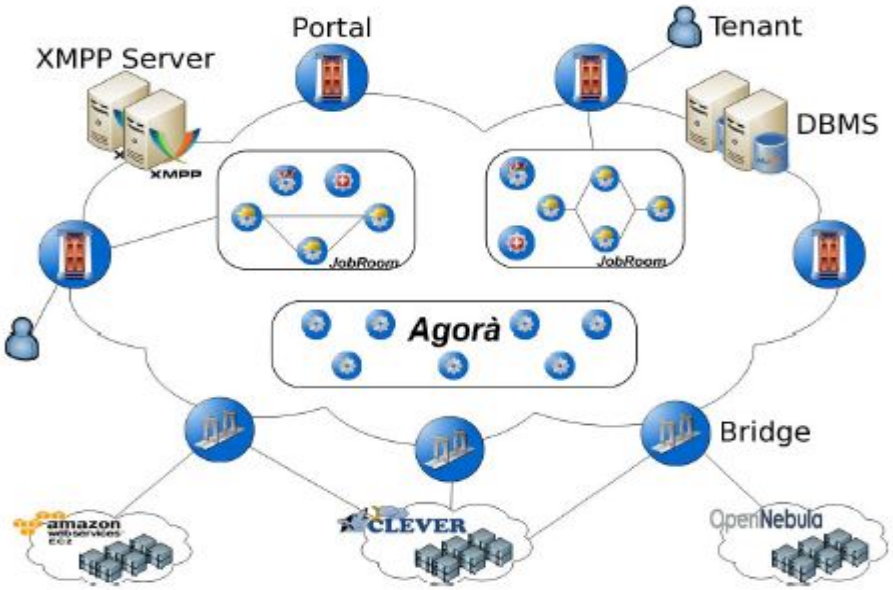
Korisnici mogu brže razviti i testirati svoje aplikacije, jer PaaS pruža alate za razvoj aplikacija.

Prema istraživanju [S19], ključne prednosti PaaS su:

- Niži ukupni trošak vlasništva, jer nema potrebe za posjedovanjem i upravljanjem svim resursima poput hardvera;

- Smanjeno upravljanje i održavanje, jer većina upravljanja je dio poslužiteljevog podatkovnog centra;
- Skalabilan i fleksibilan kapacitet sustava, dinamično skaliranje svih resursa prema potrebi, te skaliranje sustava na višoj razini apstrakcije (okruženje platforme, razvojno okruženje);

Ali, nedostatak uputa i računalnih modela čine izradu PaaS-a prilično teškom. Članak [S52] predlaže distribuirani elastični prilagodljiv usmjereni oblak (DRACO – *Distributed Resilient Adaptable Cloud Oriented*) PaaS (prema slici 4.16), kako bi se osigurala platforma za razvoj složenih algoritama u oblaku, kao i za osmišljavanje međusloja za razvoj PaaS koji može povezati slojeve usluga IaaS i PaaS, te za ostvarenje novog računalnog modela za razvoj PaaS u oblaku [S52].

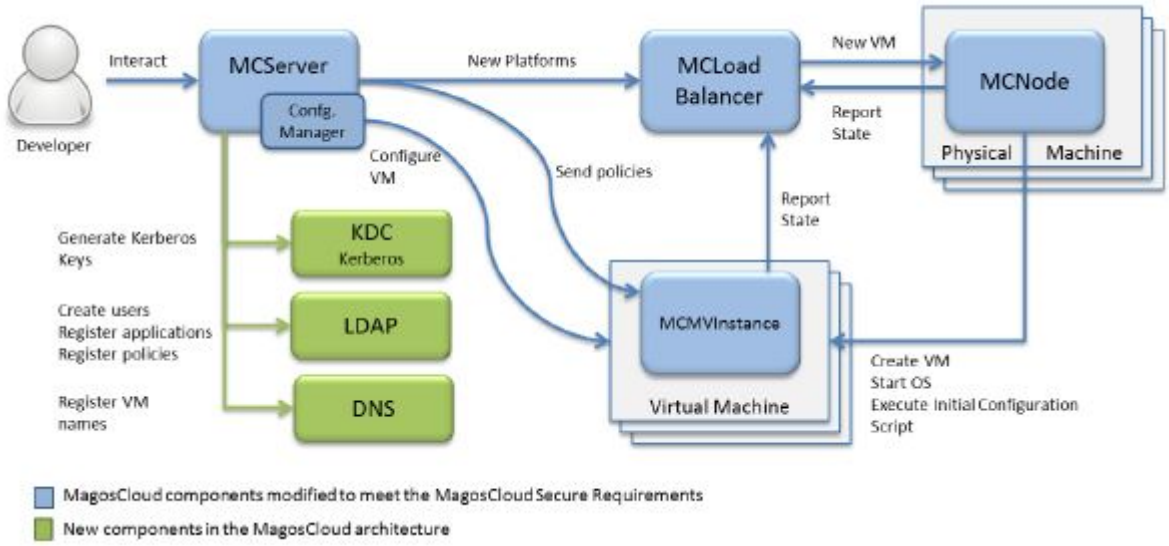


Slika 4.16. Primjer DRACO PaaS okruženja [S52]

Sustav signalizacije XMPP poslužitelja (prema slici 4.16) ima za zadatak omogućiti DRACO PaaS da se brzo oporaviti od kvarova tijekom rada. Nadalje, postoje i tri posebne sobe (*tzv. Chat rooms*): upravljanje oblakom (gdje se odvija interakcija između čvorova Portal i Bridge), Agora (soba za sastanke odabire poslovne čvorove (*eng. Jobnodes*) koji zadovoljavaju zahtjeve radnog procesa u toku), i JobRoom (soba za sastanke dodijeljena radnom procesu i čvoru Portal na kojem je ostvaren radni proces).

Iako neki poslužitelji usluga nude sigurna rješenja u oblaku, korisnici su prisiljeni preoblikovati i prilagoditi svoje programe kako bi bili u skladu s uvjetima okruženja u kojem se nalaze. Neka PaaS

rješenja ne zadovoljavaju sigurnosne zahtjeve programera i krajnjih korisnika. Studija [S54] predstavlja *MagosCloud secure* (prema slici 4.17), izrađen na temelju MagosCloud (PaaS rješenje u oblaku), dizajniran kako bi se zadovoljile potrebe programera tradicionalnih aplikacija i aplikacija e-znanosti.



Slika 4.17. MagosCloud Secure [S54]

Sustav osigurava mehanizme za provjeru autentičnosti korisnika i aplikacija. Raspodjela resursa se temelji na virtualiziranoj shemi koja jamči dostupnost i oporavak od kvarova, ovisno o opterećenju fizičkih strojeva koji održavaju infrastrukture. MagosCloud Secure pruža model koji definira kako se resursi mogu dijeliti među korisnicima.

SaaS

Izazovi i problemi

Najveći izazov SaaS aplikacija predstavlja skalabilnost. Nadalje, studije [S35, S43] definiraju informiranost zakupnika kao jedan od ključnih problema SaaS aplikacija.

Predložena rješenja

SaaS omogućuje korisnicima pristup softveru održavanom u podatkovnom centru SaaS poslužitelja. Korisnici SaaS usluga ne trebaju upravljati nadogradnjama servisa, jer se to čini u podatkovnom centru SaaS poslužitelja. Slika 4.18 prikazuje usporedbu SaaS usluga i korištenje tradicionalnog softvera [S43].

Aspect	SaaS	Traditional Software
Total Cost of Ownership(TCO)	low	high
Use Mode	subscribe, plug in	separate installation
Apply Scope	similar customers	specific customer
Specific Demand	flexible configuration or extension	specific re-development or upgrade
Maintenance	fixing a problem for one customer fixing it for everyone	fixing problem for every customer respectively

Slika 4.18. Usporedba SaaS usluga i tradicionalnog softvera [S43]

Povijesno gledano, prvi uspješni SaaS je bila rana verzija Salesforce.com u kojem je prilagođeni upravitelj aplikacija sa skupom funkcionalnosti bio izložen kao plaćeni servis.

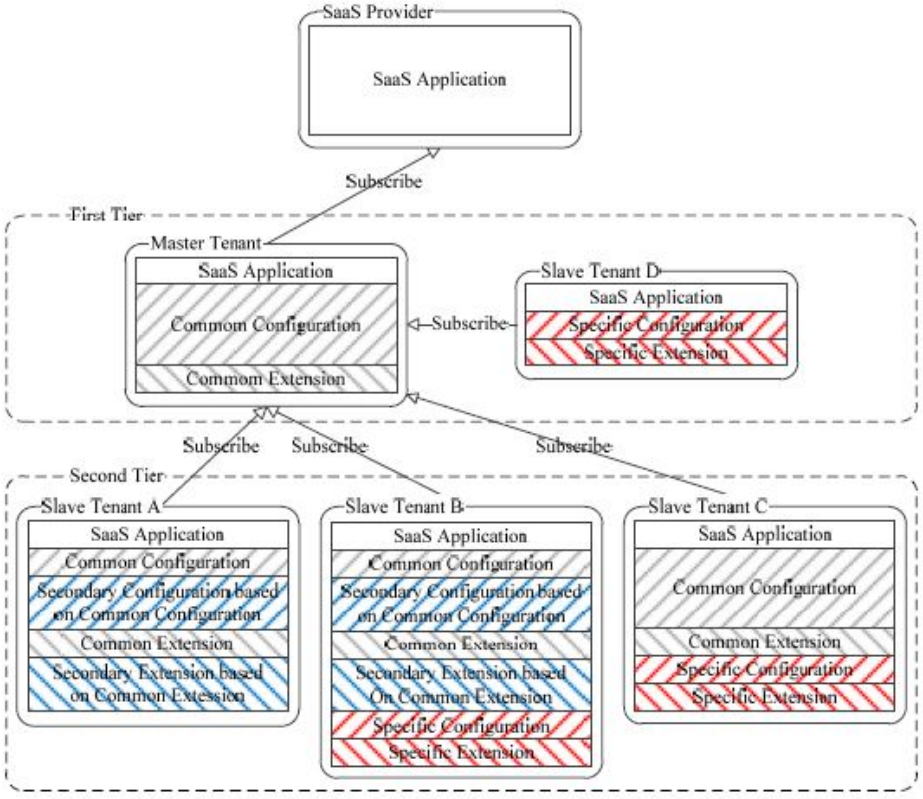
Prema studiji [S19], ključne karakteristike SaaS su:

- Poboljšane interne produktivnosti resursa, jer nema potrebe za upravljanjem vlastitim resursima;
- Brza isporuka novih aplikacija i funkcionalnosti (odmah dostiže sve korisnike spojene na mrežu);
- Bolje državne usluge, sa standardizacijom usluga za sve korisnike;

Glavni izazov SaaS aplikacija je skalabilnost. Učinkovito obrađivanje velikog broja korisničkih zahtjeva je kritično za SaaS aplikacije. Studija [S35] identificira ključne faktore koji utječu na SaaS skalabilnost i to su: stupnjevi mehanizma skalabilnosti, potpore opterećenja, oporavak i otpornost na pogreške, informiranost zakupnika, automatizirana migracija, softverske arhitekture, pristup bazi podataka. Aktualni poslužitelji SaaS usluga odnose se prema kupcima kao samostalnim entitetima i zanemaruju poslovne interakcije i integrirani odnos među korisnicima softvera. Kao rezultat toga usporena je implementacija SaaS aplikacija u malim i srednjim organizacijama.

Studija [S43] predlaže dvoslojnu SaaS arhitekturu na temelju grupe-zakupnika (*eng. Tenant*) (prema slici 4.19), koja opisuje poslovnu interakciju između svakog zakupnika skupine i postiže dvoslojnu konfiguraciju. Nadređeni zakupnik je u prvom sloju, i može proširiti i konfigurirati zajedničke informacije kako bi se zadovoljile potrebe svojih podređenih zakupnika. Također može proširiti i konfigurirati svoj dio. Podređeni zakupnici čiji dio konfiguracije se ne temelji na nadređenom zakupniku, se također nalaze u prvom sloju. Podređeni zakupnici čiji se dio konfiguracije temelji na nadređenom zakupniku, se nalaze u drugom sloju, i mogu si podesiti svoje specifične dijelove.

Zakupnici prvog sloja poznati su kao SaaS zakupnici, a zakupnici drugog sloja poznati su kao SaaS poslužitelji.



Slika 4.19. Predložena dvoslojna SaaS arhitektura [S43]

XaaS

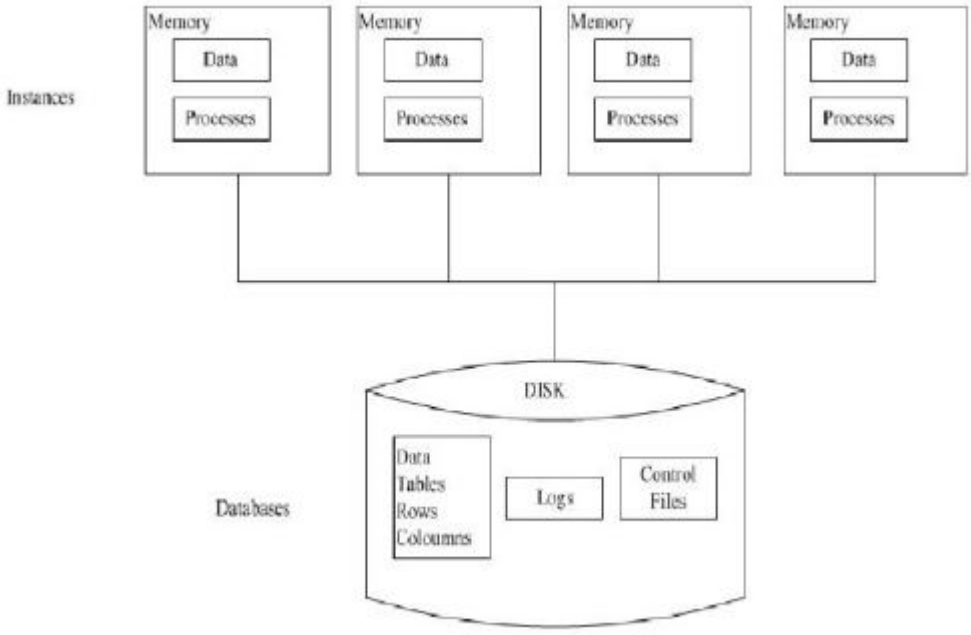
Izazovi i problemi

Arhitektura pohrane u oblaku je aktualna tema, jer se uporaba podataka i kapacitet memorije povećavaju dvostruko svake godine. Pitanja koja rješavaju studije [S6, S11, S8], a koja se odnose na pohranu, su vezani uz sigurnost, i probleme potrošnje vremena.

Predložena rješenja

Računarstvo u oblaku pruža mnoge usluge po modelu „sve kao usluga“ (XaaS). Bilo što može biti XaaS (*eng. Anything as a service*), što znači da sve može biti usluga u oblaku. Studija [S6] predlaže bazu podataka kao usluga (*DbaaS – Database as a service*) koji omogućuje velikim poduzećima: veću dostupnost, uštedu, bolju uslugu putem centraliziranog upravljanja, smanjen rizik (prema slici

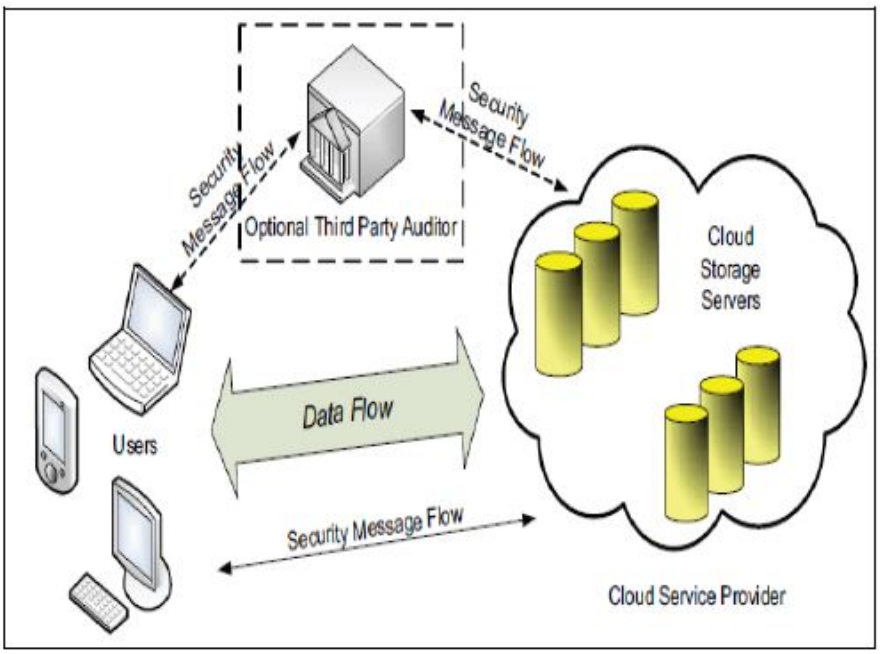
4.20); Alternativno rješenje problema pružanja funkcionalnosti baze podataka programerima aplikacija je ponuda baze podataka kao zajedničku uslugu za poduzeća u cjelini.



Slika 4.20. Arhitektura zajedničko korištenog diska [S6]

Baze podataka na zajedničko korištenom disku omogućuju nakupinama niskobudžetnih servera da koriste jednu kolekciju podataka, objavljenu od strane SAN – Storage Area Network ili NAS (Network Attached Storage). Svi podaci su dostupni svim serverima, što znači da ne postoji podjela podataka. Arhitektura baze podataka na zajedničko korištenom disku podržava elastičnu skalabilnost.

Pohrana kao usluga (*eng. Storage as a Service*) i njena arhitektura (prema slici 4.21) su definirani u članku [S23] kao poslovni model u kojem veliko poduzeće iznajmljuje prostornu infrastrukturu manjim poduzećima ili individuama. To je dobra alternativa za poduzeća male i srednje veličine koje nemaju dovoljno kapitalnog proračuna i/ili tehničko osoblje za provedbu i održavanje vlastite infrastrukture za pohranu.

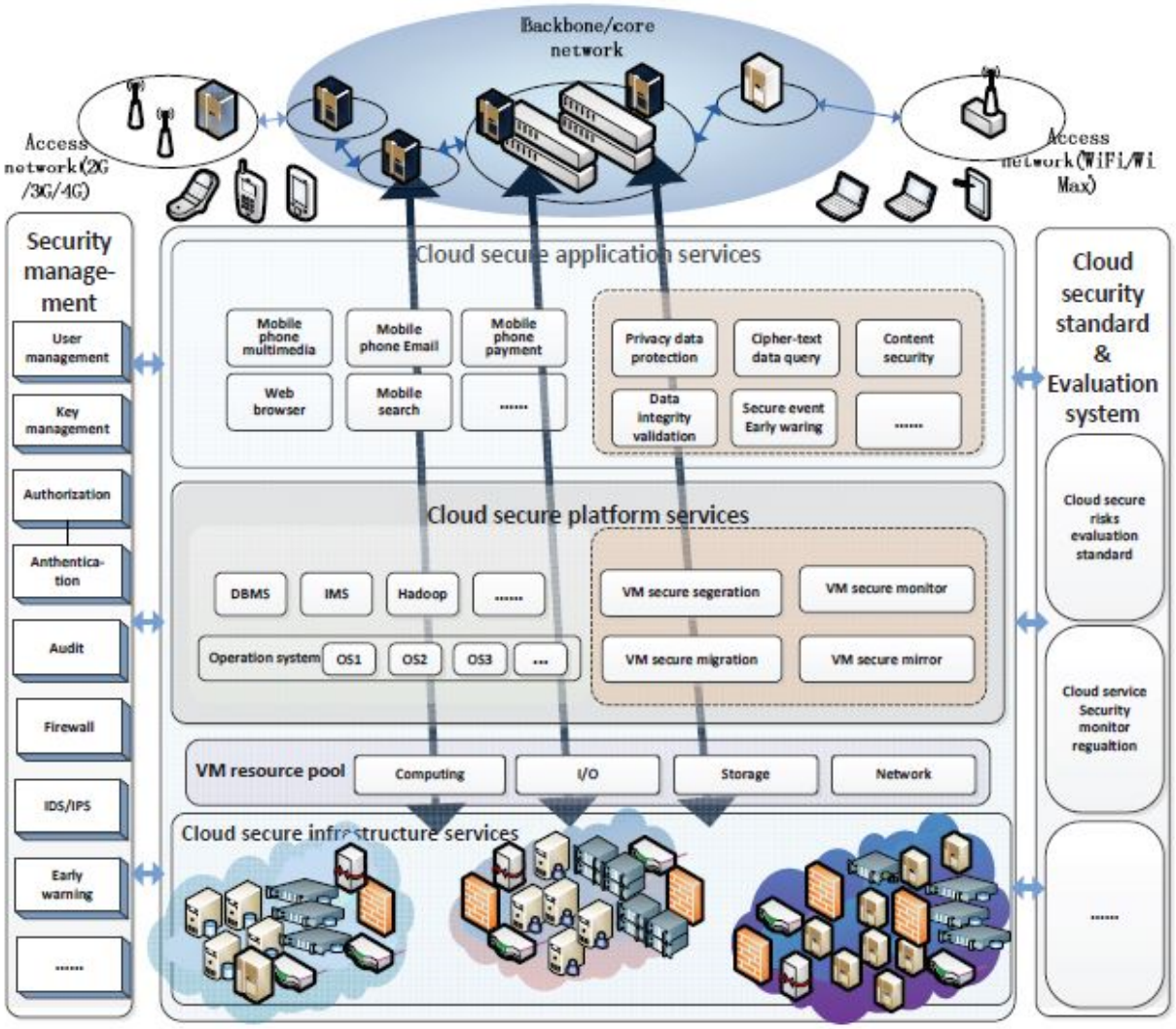


Slika 4.21. Sigurnost pohranjenih podataka u oblaku i način pristupa [S23]

Kao što je prikazano na gore navedenoj slici, postoje tri načina pristupa prostoru pohrane u računalnom oblaku: programiranje temeljeno na bloku, programiranje temeljeno na datoteci i putem web-servisa. Pristupi programiranja temeljeni na bloku i datoteci poboljšavaju performanse, dostupnost i sigurnost. Ovaj model poboljšava sigurnost, dostupnost, zaštitu podataka, agilnost pohrane, učinkovitost i skalabilnost.

Istraživanja u studiji [S8] predstavljaju arhitekturu (BETaaS) za platformu tako što su osmislili *Izgradnju okruženja za stvari kao uslugu* (*Building the Environment for the things as a service*) preko lokalnog oblaka koja omogućuje pohranu i obradu podataka u stvarnom vremenu sa poboljšanom energetskom učinkovitosti, skalabilnosti, sigurnosti i pouzdanosti.

Studija [S11] implementira model *Tehnologija kao usluga* (*eng. Technology as a service*) koji daje korisnicima pristup samo resursima koji su im potrebni za određeni zadatak, u smislu da korisnici ne plaćaju za resurse koje ne koriste. Studija [S53] prikazuje model *Elastičnost kao usluga (eng. Elasticity as a service)*, automatizirana i optimizirana raspodjela resursa u oblaku preko okruženja udružene platforme u oblaku. Analizirajući sigurnosne rizike programiranja u oblaku, studija [S36] osmišljava multi-hijerarhiju, višeslojnu, elastičnu i ujedinjenu sigurnu arhitekturu korisničkog sučelja za mobilni oblak (prema slici 4.22) prema načelu SeaaS (*Sigurnost kao usluga – Security as a service*).



Slika 4.22. Arhitektura mobilnog oblaka temeljena na sigurnosti [S36]

Razina sigurnosti usluge u oblaku za klijente koju pruža arhitektura se može prilagoditi različitim zahtjevima korisnika. Provedba ove arhitekture je fleksibilna za različite skale sustava s različitim zahtjevima. Zamisao SeaaS je [S36]: osigurati sigurnost virtualizacijskog radnog okruženja platforme u oblaku, pružiti prilagođene sigurnosne usluge u skladu s različitim zahtjevima, provesti procjenu rizika i sigurnosni nadzor operacija u oblaku, osigurati sigurnost infrastrukture računalnog oblaka i izgradnju pouzdane usluge u oblaku, očuvati integritet i povjerljivost korisničkih osobnih podataka.

Međutim, prema studiji [S49], današnje *„as- a – service“* karakterizacije poslužitelja oblaka ne uspijevaju prikladno kategorizirati komponente arhitekture.

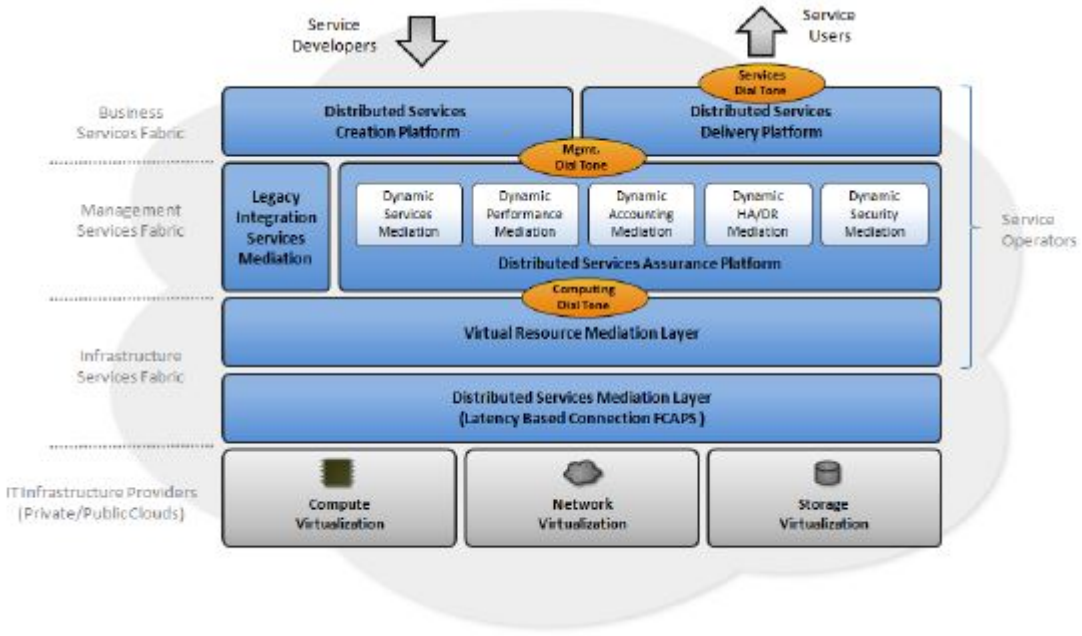
4.8. Arhitektura autonomnog upravljanja oblakom

Izazovi i problemi

Problemi koji se izdvajaju prilikom autonomnog dizajniranja oblaka su pitanja provedbe dinamične infrastrukture računalnog oblaka, kao što su: osjetljivost latencije, performanse, skalabilnost, pouzdanost i sigurnost. Studija [S13] definira nekoliko izazova dizajniranja arhitekture autonomnog računalnog oblaka, a to su: kvaliteta usluge, energetska učinkovitost, sigurnost koja uključuje povjerljivost, dostupnost i pouzdanost. Studija [S7] je riješila problem pouzdanosti i dostupnosti, na način omogućavanja auto-oporavka (oporavak razine komponenta i oporavak razine čvora) i dinamično dimenzioniranje skupine komponenta. Izazovi s kojim se potrebno baviti u budućnosti su: automatsko ažuriranje softvera, automatsko ažuriranja hardvera i profili temeljeni na vremenu.

Predložena rješenja

Upravljanje uslugama u oblaku se sastoji od procesa, aktivnosti i metoda koje stvaraju poslužitelji oblaka, u kojima se perspektiva korisnika oblaka koristi kao mjera osiguranja kvalitete usluge. Upravljanje uslugama u oblaku je podržan od tri usluge potpore: usluga arhitekture, poslovna podrška i operativna podrška [S40]. Aktualni nadzornici nisu adekvatno razdvojili upravljanje aplikacijama i upravljanje fizičkim resursima [S31]. Kao rješenje tog problema, studija [S31] predlaže referentni model (prema slici 4.23) za upravljanje mrežno orijentirane infrastrukture podatkovnog centra koji omogućuje dinamičnost, skalabilnost, pouzdanost i sigurnost u industriji.



Slika 4.23. Predloženi model referentne arhitekture [S31]

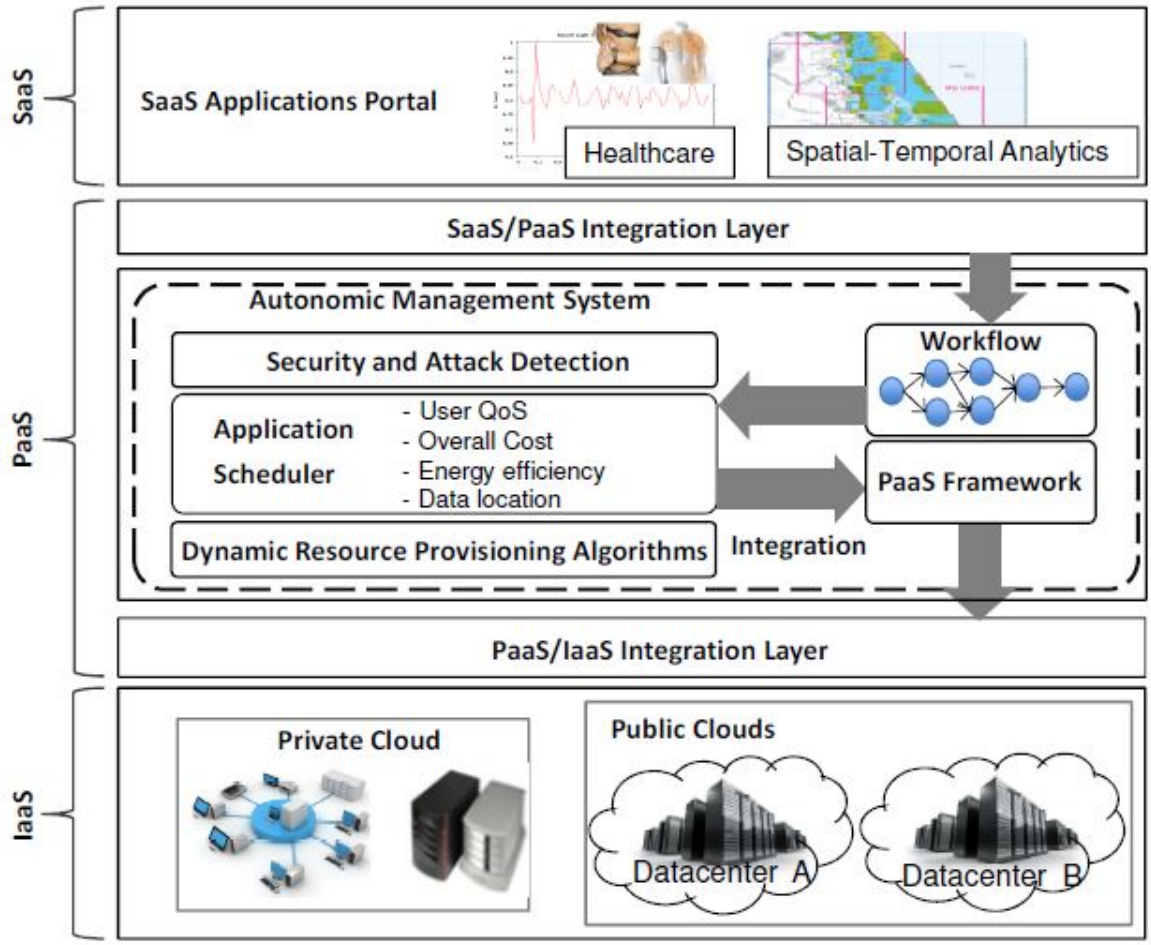
Predloženi model se sastoji od pet različitih slojeva:

- Usluga infrastrukture – uključuje dva dijela koji zajedno čine osnovu za osiguravanje resursa svim aplikacijama u oblaku:
    - Posredovanja distribuiranim uslugama – sloj apstrakcije grešaka, konfiguracije, računovodstva, performanse i sigurnosti (FCAPS). To omogućuje autonomno samoupravljanje svakog resursa u mreži;
    - Sloj posredovanja virtualnih resursa – pruža mogućnost za sastavljanje logičnih virtualnih poslužitelja;
- Platforma osiguranja distribuiranih usluga – omogućuje stvaranje FCAPS-upravljanih virtualnih poslužitelja kako bi se omogućilo izvršenje zahtjeva;
- Platforma dostave distribuiranih usluga – izvršava program i definira usluge odzivnog tona u predloženom modelu;
- Platforma izrade distribuiranih usluga – pruža alate koje koriste programeri za stvaranje aplikacija koje mogu biti sastavljene, rastavljene i distribuirane na virtualnim poslužiteljima koje automatski stvara platforma izrade distribuiranih usluga;

- Posredovanje usluge naslijedne integracije – pruža integraciju i podršku za postojeće aplikacije.

Poboljšanje energetske učinkovitosti oblaka upravljanjem

Predložena referentna arhitektura studije [S15] ima sposobnost upravljanja ekosustavom oblaka koja pruža mogućnost upravljanja životnim ciklusom, evolucijom, interakcijom, te odnosom između uključenih sudionika. Studija [S13] predlaže autonomni upravitelj oblaka (prema slici 4.24), s posebnim ulogama: raspoređivač aplikacija, energetski učinkovit raspoređivač, algoritmi rezerviranja dinamičkih resursa, te sigurnost i otkrivanje napada. Raspoređivač aplikacija dodjeljuje poslove u aplikaciji resursima koji su spremni za izvršenje na temelju parametara kvalitete usluge korisnika i troškova poslužitelja usluga. Energetski učinkovit raspoređivač osigurava korištenje energije, bez ugrožavanja razine ugovora usluga i troškova; aplikacije trebaju rasporediti resurse na način da je njihova ukupna potrošnja energije minimizirana. Algoritmi rezerviranja dinamičkih resursa provode logiku osiguravanja i upravljanja virtualiziranih resursa u okruženju oblaka na temelju raspoređivača aplikacija. Sigurnost i otkrivanje napada rade sve provjere kako bi se procijenila opravdanost zaprimljenih zahtjeva.

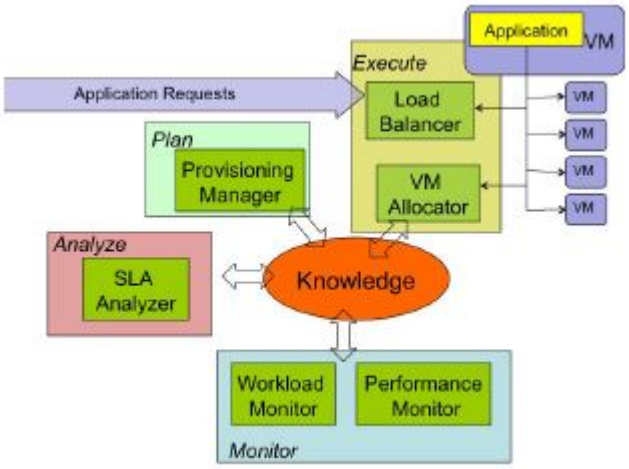


Slika 4.24. Arhitektura sustava za autonomno upravljanje oblakom [S13]

Pošto su oblaci složeni, veliki i heterogeno raspodijeljeni sustavi, upravljanje njihovim resursima je izazovan zadatak. Potrebne su automatizirane i integrirane strategije za osiguravanje sredstava za ponudu usluga koje su sigurne, pouzdane i troškovno-učinkovite. Autonomni sustavi oblaka se samostalno reguliraju, oporavljaju, štite i usavršavaju [S13, S51].

Pružanje autonomnih usluga u oblaku

Studija [S7] predlaže uvijek aktivnu (*eng. Always On*) arhitekturu koja ima sposobnost automatskog oporavka i dinamičnog dimenzioniranja skupine komponenta. Da bi se osmislilo samoprilagodljivo rješenje koje je sposobno reagirati na nepredvidive situacije opterećenja, studija [S12] predlaže model arhitekture (prema slici 4.35) za sustav autonomnih usluga. Također se identificira autonomni životni ciklus koji se sastoji od faze nadgledanja, faze analize, faze planiranja, faze izvršenja i faze znanja.



Slika 4.25. Arhitektura pružanja autonomnih usluga u oblaku [S12]

Predložena arhitektura može biti implementirana na različite načine, ovisno o tome koji su dijelovi raspoređeni na strani IaaS pružatelja i koji na strani autonomnog pružanja usluga (ASP – *eng. Autonomic service provisioning*) [S12].

Predlažu se četiri različite implementacije:

- Ekstremna ASP kontrola – ASP ima potpunu kontrolu nad svim komponentama, uključujući fizički i virtualni stroj za upravljanje;
- Potpuna ASP kontrola– ASP ima potpunu kontrolu nad fazama analize i planiranja autonomnog ciklusa, što umanjuje trošak pokretanja podatkovnog centra, i poboljšava skalabilnost i dostupnost;
- Djelomična ASP kontrola – ova realizacija se nalazi između potrebe za potpunom ASP kontrolom nad komponentama koje obavljaju prilagodbu, i prilike iskorištavanja funkcionalnosti skalabilanog i robusnog uravnoteženja opterećenja i praćenja koje nude IaaS poslužitelji;
- Ograničena ASP kontrola – može se provoditi samo uz IaaS poslužitelje koji nude funkcionalnosti auto-skaliranja. ASP je odgovoran za postavljanje pravila skaliranja na komponente auto-skaliranja IaaS poslužitelja.

U studiji [S51] je prikazan autonomni sustav upravljanja. Definirano je nekoliko kontrolnih točaka: identifikacija kontrolnih parametara, model sustava, sustav identifikacije ulaza, identifikacije modela, model ažuriranja, tip odluke sustava, stvaranje predviđanja, stvaranje koordinatora, mjerni podaci, uprava kontrolnim sustavom, autonomna kontrola sustava [S51]. Studija [S68] daje životni ciklus osmišljavanja rješenja za računalni oblak u kojima je posljednja faza sveukupno upravljanje okruženjem oblaka (upravljanje usluge u oblaku i upravljanje svim aspektima vezane uz korisnika).

4.9. Dvoslojna nasuprot troslojnoj nasuprot višeslojnoj arhitekturi oblaka

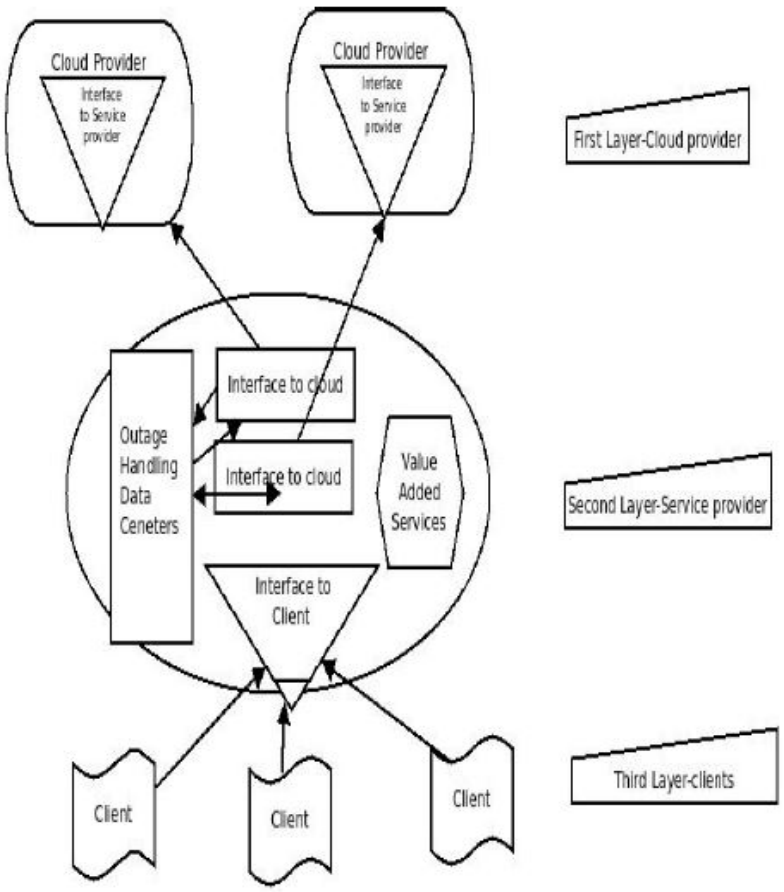
Izazovi i problemi

Većina postojećih arhitektura oblaka temelji se na dvoslojnim arhitekturama oblaka. Studija [S32] je naglasila nedostatke dvoslojnih arhitektura vezane uz pitanja pouzdanosti i interoperabilnosti; te predlaže troslojnu arhitekturu za rješavanje tih problema. Studija [S58] definira nedostatke troslojne arhitekture, te predlaže implementaciju višeslojnih oblaka kako bi se poboljšala elastičnost oblaka.

Predložena rješenja

Studija [S43] predlaže dvoslojnu SaaS arhitekturu za model zakupnika koji predstavlja grupu korisnika koji imaju relativno poslovanje i zajedničke podatke i pretplatili su se na SaaS aplikaciju u cjelini. Studija [S32] prikazuje glavne nedostatke dvoslojne arhitekture i predlaže troslojnu arhitekturu (prema slici 4.26) s poboljšanom pouzdanosti i interoperabilnosti. Prema studiji [S32] nedostaci dvoslojne arhitekture oblaka su:

- Nedostupnost usluga – zastoji ili nedostatak pouzdanosti;
- Zaključavanje dobavljača (*eng. Vendor lock-in*) – oblak treba osigurati interoperabilnost podataka i prenosivost aplikacije;
- Ograničen izbor za korisnike – nedostatak interoperabilnosti. U dvoslojnoj arhitekturi oblaka korisnici ne mogu koristiti različite usluge od različitih pružatelja usluga.



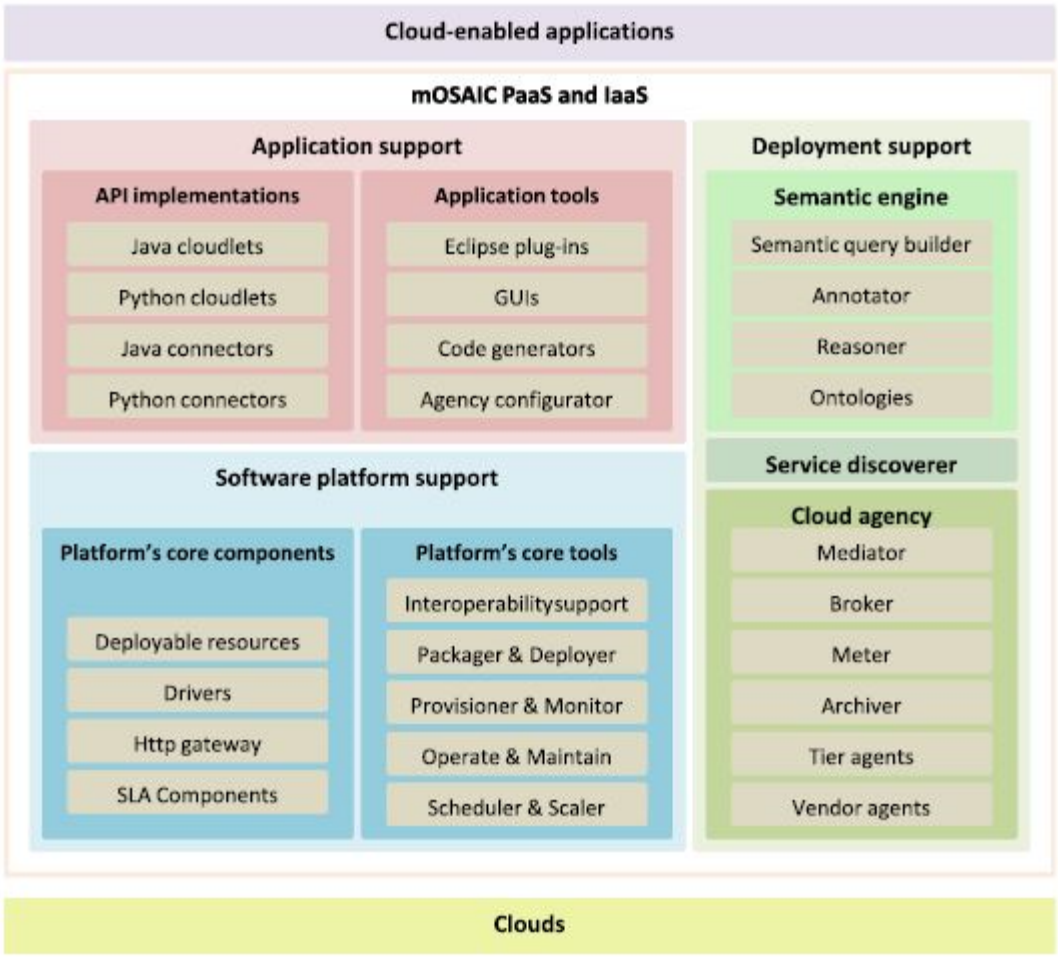
Slika 4.26. Predložena dvoslojna arhitektura [S32]

Predložena arhitektura poboljšava pouzdanost kroz podatkovne centre koji upravljaju ispadima. Kada nastane ispad u glavnim serverima, podatkovni centri preuzimaju vodstvo, bez da korisnici znaju o tome. Standardno sučelje osigurava prenosivost aplikacija, jer su podaci u slučaju ispada pohranjeni u rezervnim serverima. Nadalje, predložena arhitektura omogućuje korisniku pretplatu na više ponuda oblaka kroz više poslužitelja oblaka.

Studija [S6] predlaže troslojnu arhitekturu zajedničkog diska za bazu podataka u oblaku (*eng. 3-tiered Shared Disk Architecture for Cloud Database*) (prema slici 4.20), kako bi se poboljšala dostupnost i skalabilnost.

Međutim, migracija arhitekture iz klijent-server na modernu troslojnu arhitekturu ili web usluge će također značiti da će neke usluge biti znatno sporije. Dizajneri arhitekture za oblak moraju uključivati nove i različite tehnike obrade tog pristupa, na primjer, asinkrone tehnike slanja poruka,

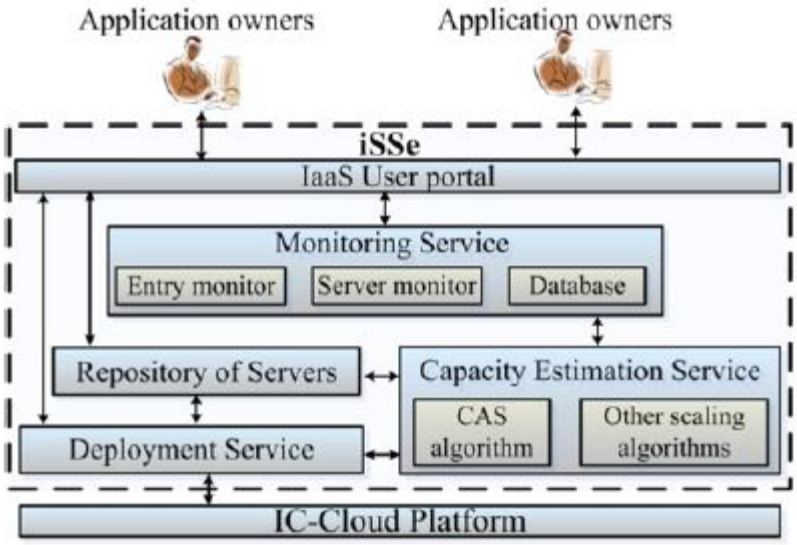
koje sustavu pridonose dodatnu skalabilnost [S19]. Tradicionalne web 2.0 aplikacije se oslanjaju na troslojne arhitekture. To je pogodno uglavnom aplikacijama s predvidivim brojem korisnika, jer slijedi mali broj obrazaca uporabe i smanjuje broj naglog povećanja opterećenja. No, pojavljuju se problemi zbog potrebe za visokom skalabilnosti i elastičnosti modernih web aplikacija. U višeslojnim aplikacijama, serveri su razvrstani u različite kategorije prema svojim funkcionalnostima. Studija [S62] predlaže novi set API za razvoj aplikacija u oblaku (višeslojnu arhitekturu) koji je integriran u open-source *Cloudware*, *mOSAIC* tako da poboljša prenosivost, dostupnost, toleranciju kvarova, te mogućnosti održavanja (prema slici 4.37).



Slika 4.27. Komponente mOSAIC arhitekture [S62]

Komponente grupirane pod podrškom aplikacije i softverske platforme za podršku su odgovorni za prenosivost oblaka. Ostale komponente su odgovorne za osiguranje elastičnosti na razini

komponente aplikacije. Predložena mOSAIC arhitektura daje slobodu razvoja progameru tako da može odabrati programski jezik tokom osmišljavanja aplikacija u oblaku, i također omogućuje izgradnju vlastitog softvera za aplikaciju. Putem mOSAIC arhitekture, programer računalnog oblaka ne mora brinuti o pitanjima specifičnim za određene poslužitelje oblaka. Ovaj istraživački projekt je financirala Europska unija (Europska komisija) [S1]. Predložene API mOSAIC-a su poboljšavaju učinkovitosti na strani aplikacije uklanjajući ograničenja kao uvjet za sinkronu komunikaciju. Studija [S66] definira mrežnu arhitekturu u kojoj su poslužitelji organizirani u višeslojnu klijent-poslužitelj arhitekturu. „Inteligentno rješenje upravljanja pohrane treba se temeljiti na višeslojnoj arhitekturi pohrane [S23]“. Studija [S58] predlaže osjetljivi (prema troškovima) pristup elastičnom skaliranju u višeslojnoj aplikaciji u oblaku (prema slici 4.28).



Slika 4.28. Predložena arhitektura mehanizma Imperial Smart Scaling (iSSe) [S58]

Predloženo rješenje iSSe je međusloj između poslužitelja oblaka i vlasnika aplikacija. Korisnički portal IaaS pomaže vlasnicima aplikacija pružiti usluge krajnjim korisnicima aplikacija. Sve komponente rade zajedno da pomognu poboljšanju elastičnog skaliranja. Nadzorna usluga prati izvršenje aplikacija pomoću dvije vrste monitora (monitora ulaza, i monitor poslužitelja). Usluga procjene kapaciteta procjenjuje skaliranje određenog broja poslužitelja korištenjem informacija u bazi podataka. Usluga implementacije automatski raspoređuje skaliranje poslužitelja. Kada se jedan server doda, iSSe može automatski odabrati poslužitelja oblaka koji nudi najjeftinije cijene.

4.10. Ostale arhitekturalne paradigme i rješenja za dizajniranje oblaka

U ovom dijelu analizirane su ostale paradigme arhitekture razmatrajući konkretna područja primjene programiranja u oblaku, kao što su: proizvodnja oblaka, mobilno računarstvo u oblaku, Internet stvari, oblak u oblacima, arhitekture računalnih oblaka za vladu (*eng. Government Cloud Computing*) , upravljanje oblakom i znanstvene arhitekture računalnog oblaka.

Proizvodnja oblaka

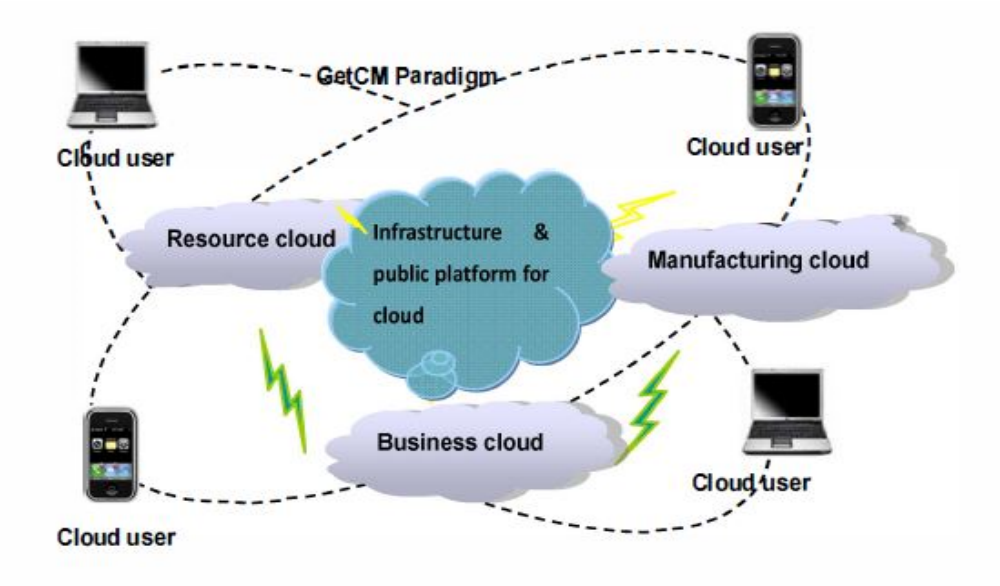
Izazovi i problemi

Problemi umrežene proizvodnje su: zatvoreni heterogeni mehanizmi nagodbe, nedostatak prikladnog mehanizma opskrbe resursa, nedostatak ujedinjenog mehanizma za upravljanje resursima, nedostatak prilagodljivosti promjenama vanjskog okružja i nejasni poslovni modeli.

Predložena rješenja

Za razliku od konvencionalnog proizvodnog pristupa, znanje o programiranju oblaka može uveliko poboljšati opseg, agilnost i sigurnost aplikacija, te smanjiti troškove pružanja novih usluga. Uvođenje računarstva u oblaku kao rješenje problema umrežene proizvodnje može pružiti nove ideje i mogućnosti. Predložena vizija proizvodnje oblaka *GetCM* (prema slici 4.29) ima odgovornost pružiti pouzdanost i fleksibilnost [S21]. Također se definiraju prednosti GET paradigme proizvodnje oblaka iz različitih perspektiva:

- Perspektive poslužitelja – objaviti i dijeliti proizvodne resurse i poslove;
- Perspektive podnositelja zahtjeva – pronaći i koristiti najprikladnije proizvodne resurse i poslove, poboljšanje učinkovitosti i plaćanje samo ono što se koristi;
- Perspektive poduzeća - pozicioniranje lanca vrijednosti, dobiti više tržišnih prilika;
- Perspektive gospodarstva – raspodjela sredstava, uspostava konkurencije u ekosustavu nabavnog lanca;



Slika 4.29. Paradigma Get CM [S21]

Paradigma GetCM se sastoji od pet dijelova: oblak resursa, poslovni oblak, proizvodni oblak, infrastruktura i javna platforma, i korisnik oblaka. Korisnici oblaka su kategorizirani u tri vrste: pružatelj usluga proizvodnje u oblaku, podnositelj zahtjeva, i poslužitelj javnih platformi i infrastrukture. Podnositelj zahtjeva usluge može popraviti fleksibilnost resursa na temelju sinteze zrnatosti (*eng. Granularity*).

Mobilno računarstvo u oblaku

Izazovi i pitanja

Uvođenje programiranja u oblaku na mobilni internet dovodi do mijenjanja arhitekture mobilnog interneta, a javljaju se mnogi novi sigurnosni problemi, kao što su sigurnost podataka i zaštita privatnosti između domena, sigurnost virtualnih okruženja operacije i nadgledanje sigurnosti između domena. Studija [S5] definira izazove računarstva u mobilnom oblaku, kao što su:

- Latencija mreže – računarstvo u mobilnom oblaku se suočava s izazovima latencije mreže, zbog ograničene pojasne širine bežičnih mreža;
- Razni mehanizmi pristupa – računarstvo u mobilnom oblaku zahtijeva bežične veze dostupne na zahtjev sa skalabilnom pojasnom širinom i energetski učinkovitim uređajima;
- Latencija predaje – uzrokovano zbog kretanja između različitih pristupnih tehnologija, npr sa 3G na WiFi. Također se događa zbog prelaska iz jedne mrežne domene na drugu;

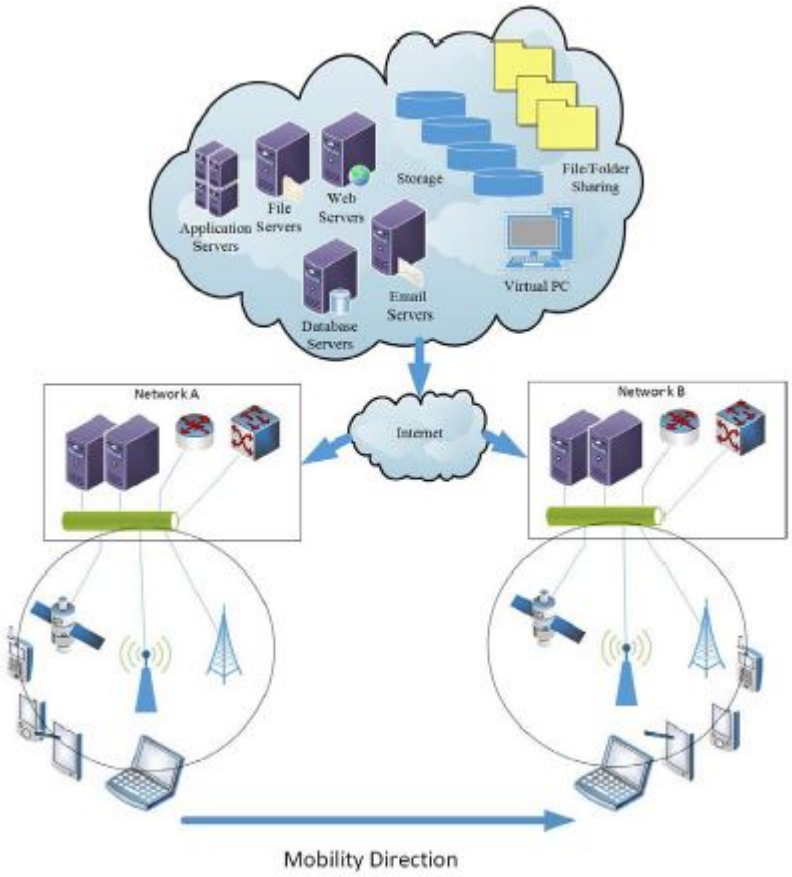
- Elastična mobilna aplikacija – ograničeni resursi na mobilnom uređaju ne dopuštaju ažuriranje izvan mreže;
- Sigurnost i privatnost – računarstvo u mobilnom oblaku može doživjeti različite napade zbog slabe snage i sporog izvođenja CPU. Sigurnost također može biti ugrožena zbog autentičnosti na različitim mrežama tijekom rada. Mobilni korisnik će možda morati provjeriti autentičnost kod različitih poslužitelja usluga ili poslati podatke za provjeru autentičnosti početnom serveru koji može izložiti korisnika napadu [S5].

Studija [S5] definira izazove poduzeća s kojima se suočava dizajniranje arhitekture mobilnog oblaka:

- Sigurnost identiteta – kako osigurati identitet korisnika, ako se mobilni uređaj proda ili daje drugoj osobi;
- Administrativni poslovi – kako održavati korisnička prava na lokalnoj razini i na oblaku jer različiti korisnici mogu trebati različite vrste prava za aplikacije i usluge;
- Revizija i praćenje – kako pratiti aktivnosti kada korisnici mijenjaju mrežne granice. Još uvijek ne postoji rješenje za snimanje, reviziju ili praćenje podataka koje koriste mobilni korisnici;
- Osobni ili službeni podaci – kako osigurati službene podatke na uređaju ili oblaku u slučaju izgubljenog ili ukradenog uređaja;
- Zalihost usluga i balansiranje opterećenja – ako postoji neki kvar na hardveru, usluga mora raditi i biti isporučena korisnicima. Mnogi poslužitelji usluga u oblaku nemaju vlastite podatkovne centre. U slučaju velikog broja mobilnih korisnika koji pristupaju videu ili podacima, vrlo je važno osigurati balansiranje opterećenja u cilju učinkovitog pružanja usluga;
- Oporavak od katastrofe – postoji mogućnost gubitka podataka ili usluga u slučaju propusta, jer mnogi poslužitelji oblaka ne implementiraju zalihe resursa u podatkovnom centru.

Predložena rješenja

Rast bežičnih mreža je dovelo do izuma pametnih mobilnih uređaja kao što su mobilni telefoni, prijenosna računala i tablete. Prema ICT statističkim podacima Cisco Visual Networking indeksa, očekuje se rast mobilnih podataka od 66% godišnje, zbog toga što mobilne mreže i računalni oblaci formiraju novi računalni model, mobilni oblak (*eng. Mobile Cloud Computing*) [S5].



Slika 4.30. Mobilni oblak [S5]

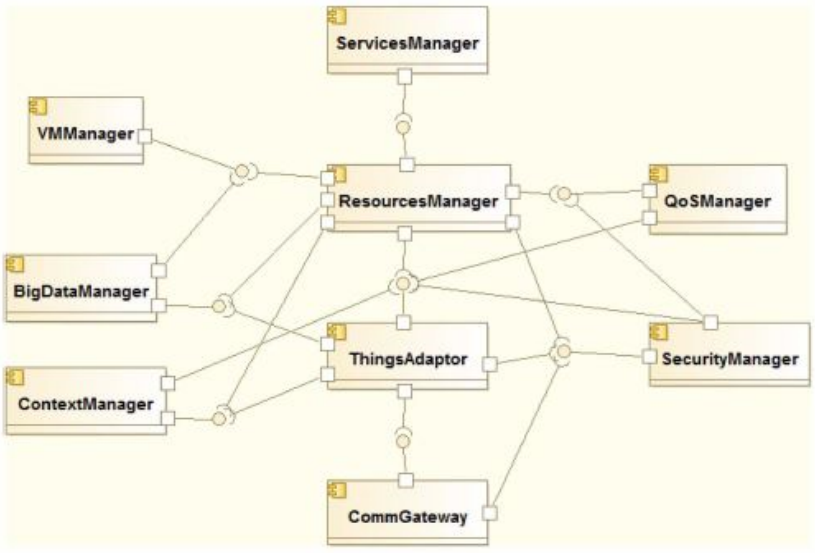
Arhitektura mobilnog oblaka (prema slici 4.30) temelji se na četiri sloja kako bi se osigurala pouzdanost, sigurnost i valjanost za podršku mobilnih korisnika [S5]: pristup, upravljanje, virtualni i fizički sloj. Razvijeni su mnogi različiti programi koji se temelje na računarstvu u mobilnom oblaku kao Google Gmail, Kalendar, Grupe, Karte, iCloud, Live Mesh itd. Pet najvažnijih karakteristika dizajniranja mobilnog oblaka [S5]: samoposluga na zahtjev, pristupne širokopojasne mreže, udruživanje resursa, brza elastičnost, i usluga koja se može mjeriti. Studija [S36] predlaže višeslojnu, elastičnu, sigurnu arhitekturu preko platforme u oblaku na temelju mobilnog interneta (prema slici 4.22).

Izazovi i pitanja

Prema studijima [S69, S8], izazovi integracije programiranja u oblaku s Internetom stvari (*eng. Internet of Things*) su: računalna infrastruktura, upravljanje podacima i umrežavanje. Studija [S8] predlaže BETaaS platformu kako bi se poboljšala fleksibilnost, skalabilnost, sigurnost, pouzdanost.

Predložena rješenja

Integracija računarstva u oblak s Internetom stvari predstavlja održiv pristup kako bi se olakšao razvoj aplikacija za stvari (svi računalni objekti koji djeluju na Internetu) [S24]. Oblak je čvrsto spojen na Internet stvari. Razlika između Interneta stvari u oblaku i konvencionalnog Interneta stvari je sposobnost razvoja, implementacije, pokreta i upravljanja aplikacija stvari u oblaku preko interneta, studija [S24] predlaže CloudThings arhitekturu. Studija [S8] predlaže BETaaS (Izgradnja okruženja za stvari kao uslugu – *eng. Building the Environment for the Things as a Service*), kako bi se poboljšala fleksibilnost, energetska učinkovitost, skalabilnost, sigurnost i pouzdanost (prema slici 4.31).



Slika 4.31. Predložena BETaaS arhitektura [S8]

- Upravitelj uslugama (*eng. Service manager*) se koristi za upravljanje uslugama i aplikacijama u BETaaS platformi. Komunicira sa upraviteljem resursa u svrhu dodjele

sredstava, kao i ostalih komponenata koje imaju utjecaj na aspekte koji mogu utjecati na izvršenje aplikacija.

- Upravitelj resursa (*eng. Resource Manager*) odlučuje kako upravljati raspoloživim resursima. Odgovoran je za izradu lokalnog oblaka stvari u BETaaS instanci, što jamči skalabilnost.
- Adapter stvari (*eng. Things adaptor*) označava most između stvari i BETaaS sposobnosti koje se odnose na stvari. Također pruža podatke onim komponentama koje ovise o tim podacima kako bi ispravno radili.
- Veliki upravitelj podataka (*eng. Big dana manager*) nudi napredne mogućnosti pohrane velike količine podataka koji dolaze iz stvari i njihovo praćenje.
- Upravitelj VM (*eng. VM Manager*) stvara i uklanja lagane VM-ove koji se mogu koristiti za obavljanje obrade ili osiguravanja izolirane pohrane podataka.
- Upravitelj atributima kvalitete (*eng. Quality of Service manager*) dohvaća podatke od BETaaS instance i definira razinu atributa kvalitete. Održava zakone atributa kvalitete koje mogu utjecati na upravljanje resursa u BETaaS instanci.
- Upravitelj konteksta (*eng. Context manager*) pruža stvarima informacije konteksta.
- Upravitelj sigurnosti (*eng. Security manager*) omogućuje izračun povjerenja i ostale osnovne usluge koje se koriste na različitim razinama.
- Komunikacijski prolaz (*eng. CommGateway*) izravno komunicira sa stvarima. Pruža platformi fleksibilnost i sposobnost evolucije.

Studija [S69] razmatra TransCloud infrastrukturu koja se usredotočuje na probleme kao što su: osigurati da korisnici oblaka mogu pokrenuti aplikacije gdje god imaju pristup, osigurati sigurno izvršenje udaljenog upita i za daljinskog korisnika i nositelja podataka, i osmišljavanje jednostavne, arhitekture mreže za prostorno distribuirane heterogene podatke.

Izazovi i problemi

Studija [S39] definira pitanja sigurnosti i pouzdanosti s kojima se suočavaju postojeće arhitektura računalnog oblaka, kao što su:

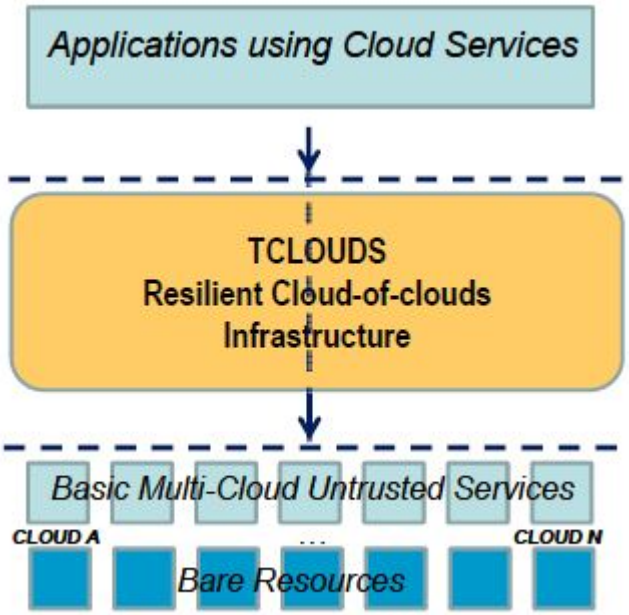
- Pitanja pouzdanosti i sigurnosti se ne mogu riješiti samo u aplikacijskom sloju, trenutno zahtijevaju sigurnosno specifična rješenja koje treba osigurati u nižim slojevima (infrastruktura i platforma) arhitekture oblaka;
- Neki IaaS i PaaS pristupi za postizanje otpornosti mogu uzrokovati tešku i skupu migraciju i interoperabilnost, što stvara *vendor lock-in* (kada korisnik postaje ovisan o jednom poslužitelju oko održavanja i nadogradnji sustava);
- Ciljevi visoke prilagodljivosti mogu biti ugroženi u rješenjima pojedinačnih točaka propusta (npr zajednička ili povezana upravljanja i domene povjerenja).

Predložena rješenja

Veliki poslužitelji usluga oblaka pokušavaju postići fleksibilnost uz razvoj nekoliko, drukčije razmještenih, podsustava u oblaku, ali da i dalje ostaju pod jednom upravom i stranom povjerenja, s obzirom na zajednički rad. Paradigma oblaka u oblacima (*eng. Cloud of Cloud*s) je predložen način studije [S39] kako bi se postigla elastičnost računalnog oblaka. Paradigma oblaka u oblacima proširuje koncept oblaka, iskorištavanjem višestrukog okruženja oblaka za izradu različitih ekosustava, dopuštajući korisnicima da samostalno organiziraju način na koji koriste razne usluge računalnog oblaka. Da bi se riješili navedeni izazovi i pitanja, studija [S39] predlaže TCloud arhitekturu (prema slici 4.32). Princip TCloud arhitekture je izgraditi podatkovne centre s serverima koji pružaju napredna sigurnosna svojstva [S1]. Predložena arhitektura omogućuje automatsku računalnu otpornost protiv napada i nezgoda, a izbjegava pojedinačne točke propusta. Da bi se to osiguralo, imenuju se algoritmi poput bizantske tolerancije kvarova[1] i proaktivan oporavak. Ovi mehanizmi su transparentni i nude apstrakcije oblaka u oblacima korisnicima više razine, dok skrivaju složenost upravljanja zalihosti i raznolikosti. Predložena arhitektura također čuva potrebe

[1]Cilj bizantske tolerancija kvarova je da se može braniti protiv bizantskih propusta koji su proizvoljni propusti koji se javljaju za vrijeme izvođenja algoritma na distribuiranom sustavu [8]

zalihosti dok omogućava raznoliki ekosustav. Ovaj istraživački projekt je financirala Europska unija (Europska komisija) [S1].



Slika 4.32. Predložena TCloud arhitektura [S39]

Arhitekture oblaka za vladu

Izazovi i problemi

Prilike koje nudi programiranje u oblaku državama diljem svijeta su postizanje optimizirane, isplative, informatičke infrastrukture koja podržava agencijsko poslovanje, a pruža pouzdanost i sigurnost usluga. Postoji velika potreba da vlada pretvori svoju informatičku infrastrukturu virtualizacijom podatkovnih centara, da konsolidira podatkovne centre i operacije, te prihvati poslovni model računarstva u oblaku.

Predložena rješenja

Da bi postigle globalnu konkurentnost, vlade zemalja u razvoju provode tehnologije programiranja u oblaku kako bi se omogućilo njihovim zemljama da sudjeluju u trenutnoj ICT revoluciji [S61]. Postoji niz procesa i tehničkih izazova i pitanja koja država treba pažljivo razmotriti prije

osmišljavanja ili korištenja rješenja programiranja u oblaku ili uzimanja u obzir korištenje računalnih resursa u oblaku i kapaciteta za državne potrebe. Microsoftovi Privatni oblaci za vlade (*eng. Microsoft Government Private Clouds*) su dodijeljena okruženja računalnih oblaka, koja se bave sigurnosti podataka, privatnosti i regulatornim zahtjevima specijaliziranim za vlade [S19]. Američka vlada procjenjuje svoju potrošnju tokom migracije u rješenja u oblaku na 20 milijarde dolara [S45]. Studija [S19] definira četiri vrste državnih rješenja u oblaku, ovisno o potrebama organizacije, i razine rizika uzrokovanih u javnim i privatnim oblacima:

- Javni državni oblak – *eng. Government Public Cloud* (za srednja/federalna ministarstva/agencije) – iz prespektive tehnologije, ovo je najjednostavnije rješenje za provedbu računalnog oblaka u vladi, ali postoji najviše rizika privatnosti, kontrole i sigurnosnih napada. Smanjuje ukupne troškove vlasništva uklanjanjem potrebe za upravljanjem IT infrastrukturom, te omogućava brz, fleksibilan i ekonomičan način za proširenje dostupnosti usluga korisnicima;
- Privatni dodijeljen državni oblak – *eng. Government Private Cloud Dedicated* (za srednja/federalna ministarstva/agencije) – tehnologije i usluge su smještene u podatkovnim centrima dobavljača, ali lokacija podatkovnog centra je u državi zbog pitanja sigurnosti i privatnosti;
- Privatni self- hosted državni oblak – *eng. Government Private Cloud Self-Hosted* (za srednje/federalne vlade koje pružaju zajedničke usluge) – vlada može postići sve prednosti programiranja u oblaku zadržavajući poboljšanu razinu kontrole i sigurnosti. Samo entitet vlade može smjestiti državni podatkovni centar. Ovo rješenje omogućuje državama da poboljšaju svoje podatkovne centre, i za operaciju i pružanje usluga u oblaku kao dio svojih zajedničkih službi;
- Privatni smješteni državni oblak - *eng. Government Private Cloud Hosted* (za srednja/federalna ministarstva/agencije i vlade koji koriste vanjske IT resurse) – rješenje gdje sigurnost podataka zahtijeva da privatni oblak mora biti smješten u državi i gdje međunarodna pojasna širina interneta nije dovoljna za smještaj privatnog oblaka u drugoj državi.

Studija [S61] nudi model e-obrazovanja koja daje vladi odgovornost za osiguravanje infrastrukture i e-obrazovnih usluga obrazovnim ustanovama u državi. Usvajanje računalnog oblaka za obrazovanje

nudi velike uštede za vlade putem konsolidacije podatkovnog centra, agregacije zahtjeva, i više-zakupničog modela.

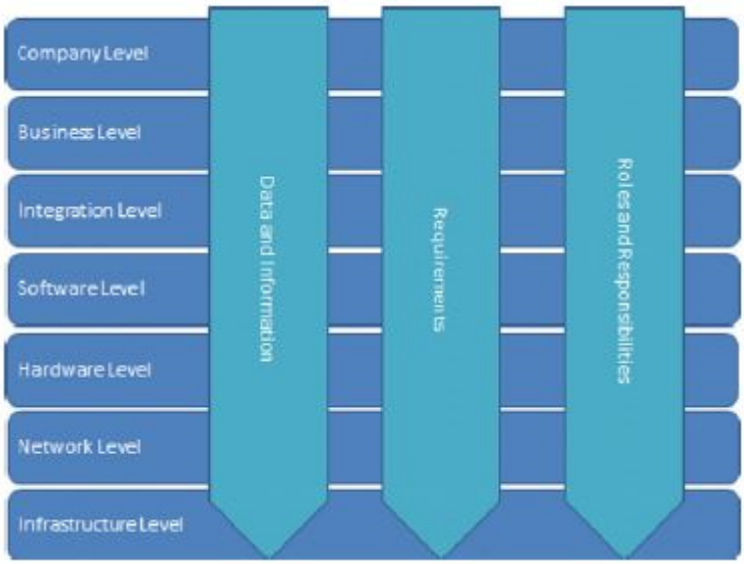
Uprava računalnim oblakom

Izazovi i problemi

Sigurnosna pitanja dostupnosti podataka postoje u okruženju uprave oblaka (*eng. Cloud governance*). Budući izazovi su sljedeći: moguće veze trebaju biti razvijene iz točke gledišta svake relevantne perspektive. Usklađenost paradigma treba biti još više navedena i proširena s mogućnošću pružanja konkretne preporuke koje se temelje na poduzećima ili tehnologiji, i treba biti poboljšan razvoj pratećih alata [S63].

Predložena rješenja

Paradigma proširenja tradicionalne IT uprave na računalne oblake se naziva uprava računalnim oblakom [S63]. Studija [S63] nudi model uprave (prema slici 4.33) za upravljanje arhitekturom poduzeća. Ovaj predloženi model mogu iskoristiti organizacije korisnika oblaka kao i poslužitelji oblaka. Organizacije korisnika oblaka mogu osnivati vlastite procese upravljanja temeljene na alatima modela, a poslužitelji oblaka mogu osigurati bolju uslugu između njihovih ponuda i zahtjeva organizacije njihovih potencijalnih i sadašnjih korisnika.



Slika 4.33. Predloženi model arhitekture [S63]

Predloženi model sastoji se od sedam horizontalnih (tvrtka, posao, integracija, softver, hardver, mreža i infrastruktura) i tri vertikalnih (podaci i informacija, zahtjevi, uloge i odgovornosti) razina.

Znanstvene arhitekture računarstva u oblaku

Izazovi i problemi

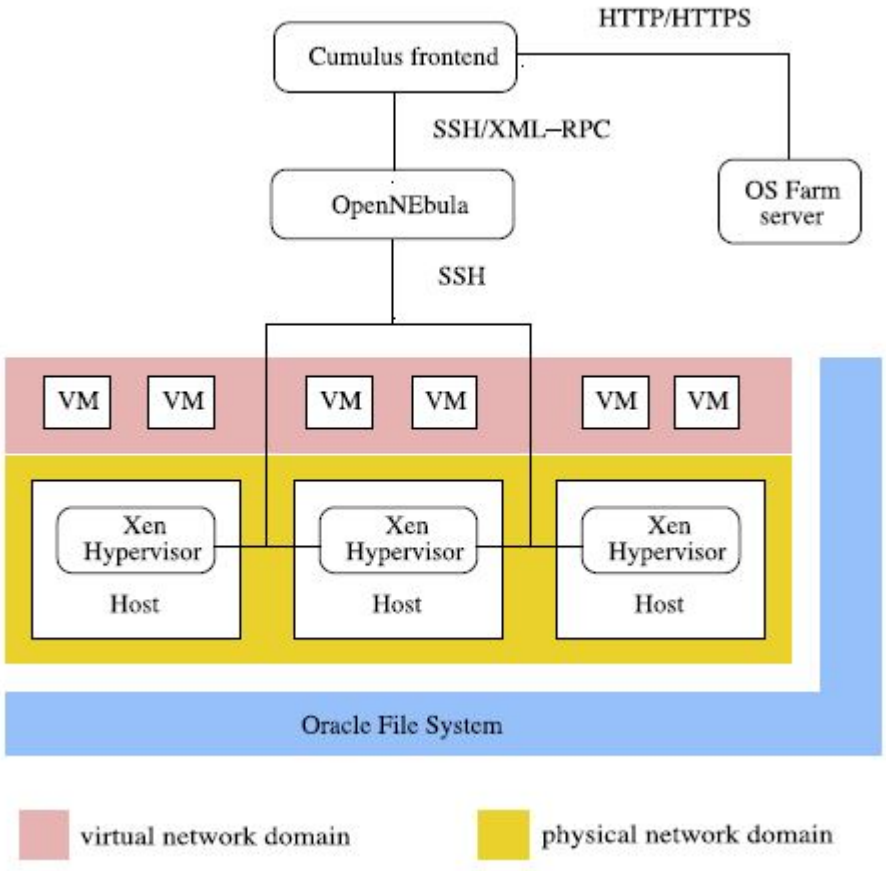
Studije [S64, S50], tokom osmišljavanje arhitektura znanstvenih računalnih oblaka (*eng. Scientific Cloud Computing Architectures*), su naišle na sljedeća pitanja: skalabilnost, fleksibilnost i modularnost. Studija [S48] obraća pažnju na sigurnost kao glavni problem tokom osmišljavanje aplikacije u oblaku. Također jedni od problema su automatizam, skalabilnost i interoperabilna sigurnost.

Predložena rješenja

Povećanjem rasta podatkovnog prometa, također eksponencijalno rastu i znanstvena analiza i kompleksnost izvođenja procesa. Prednosti izvođenja znanstvenih aplikacija u oblaku [S50]:

- Opseg znanstvenih problema može biti znatno povećan s ograničenim dijeljenjem resursa, platforma u oblaku može ponuditi dovoljnu količinu računalnih resursa, kao i prostora za pohranu;
- Implementacija aplikacija može biti fleksibilna i povoljna;
- Znanstvene aplikacije u oblaku mogu dobiti dinamički dodijeljena sredstva;
- Računalni oblak pruža priliku za poboljšanje omjera performansi/troškova za znanstvene probleme većih razmjera.

Međutim, znanstvenici i medicinski istraživači su još uvijek u potrazi za jednostavnom arhitekturom u oblaku koja omogućuje sigurnu suradnju i razmjenu distribuiranih baza podataka [S48]. Studija [S64] predstavlja Cumulus projekt (prema slici 4.34). Cumulus omogućuje virtualne računalne platforme za znanstvene i inženjerske aplikacije.

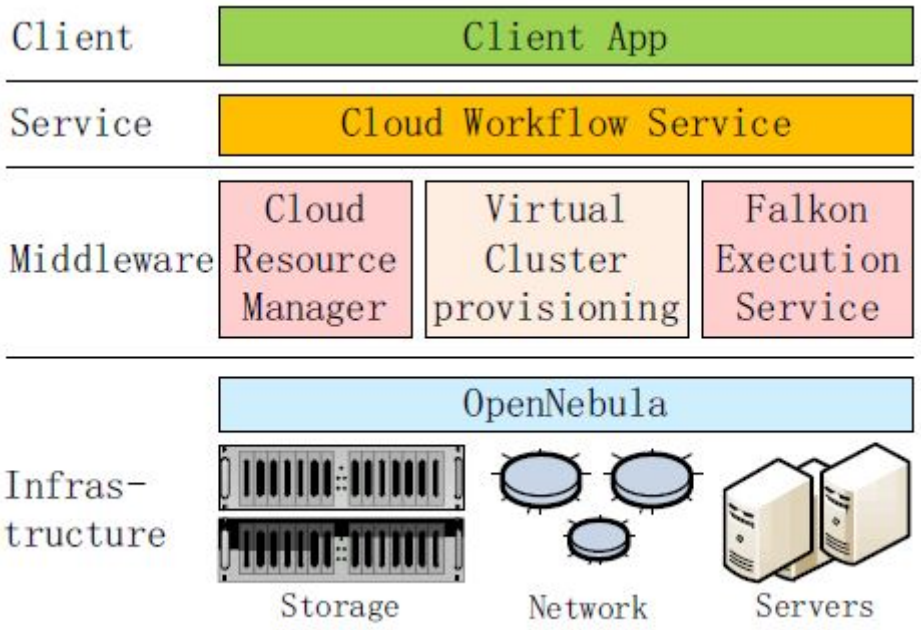


Slika 4.34. Predložena arhitektura Cumulus [S64]

Predložena arhitektura poboljšava skalabilnost i čuva autonomiju računalnih centara:

- Cumulus sučelje ne ovisi o bilo kojem specifičnom lokalnom sustavu upravljanja virtualizacije, nego koristi softver treće strane;
- Računalni centri unutar oblaka definiraju vlastite politike upravljanja resursima;
- Cumulus sučelje može prenijeti korisničke zahtjeve prema drugim oblacima.

Studija [S50] predstavlja platformu znanstvenog računarstva u oblaku CloudDragon (prema slici 4.35).



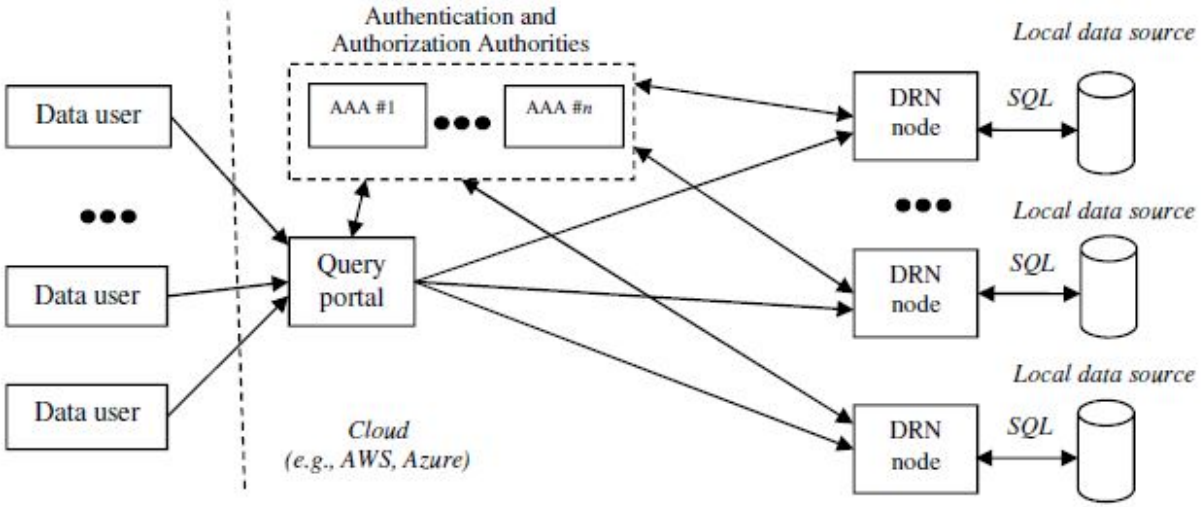
Slika 4.35. Predložena arhitektura CloudDragon [S50]

Arhitektura se sastoji od četiri sloja:

- Sloj klijenta – pruža razvoj sa strane klijenta i daje alate za specifikaciju aplikacija;
- Sloj usluge – usluga tijekom rada u oblaku se temelji na brzom upravljanju sustava i prikazana je kao gateway platformama u oblaku;

- Sloj međusoftvera – nekoliko komponenti (Upravitelj resursima u oblaku, poslužitelj virtualnih nakupina i servis izvršenja zadataka) su integrirane kako bi se premostio jaz između sloja usluga i sloja infrastrukture;
- Sloj infrastrukture – upravlja resursima podatkovnog centra u oblaku, kao što su poslužitelji, mreže i pohrane.

Studija [S48] dizajnira CloudDRN (prema slici 4.36) za sigurno dijeljenje istraživačkih podataka u oblaku, i omogućuje poduzećima da sudjeluju u CloudDRN putem servera pokrenutih u okviru samih poduzeća.



Slika 4.36. Predložena arhitektura *Cloud Distributed Research Network* [S48]

Za poboljšanje učinkovitosti podataka, lakše dostupnosti, jake pouzdanosti i stalnog pristupa, studija [S17] predlaže kao rješenje napraviti šest lokalnih pod-oblaka na temelju šest pod-kontinenata umjesto jednog globalnog oblaka. Ako jedan pod-oblak ispadne, bilo koji klijent s bilo kojeg mjesta u svijetu ima pravo koristiti drugi pod-oblak pod tehnikom „failover“ [S17]. Studija [S46] uvodi i predlaže međusoftver Cloud@home za poboljšanje interoperabilnosti među oblacima.

5. EVOLUCIJA RAČUNALNOG OBLAKA, ŠTO SLIJEDI?

Studija [S29] definira razvoj tržišta računalnog oblaka sa tri faze:

- „Monolitni“ – trenutno stanje;
- „Vertikalni lanac opskrbe“ – neki poslužitelji usluga oblaka će iskoristiti usluge u oblaku od drugih poslužitelja usluga;
- „Horizontalna federacija“ – manji, srednji i veliki poslužitelji će sami izvršiti horizontalnu federaciju da steknu učinkovito korištenje vlastitih sredstava;

OpenNebula, Nimbus i Eucalyptus frameworks su gotovo u drugoj fazi [S29]. Budući da, standardne upute osmišljavanja arhitekture za oblak ne postoje, svaki međusoftver oblaka se može jako razlikovati jedni od drugih, što čini prijelaz na treću fazu vrlo teško postići.

Tijekom sljedećeg desetljeća, rasprostranjena dostupnost masivnog računalnog kapaciteta će dati svakome mogućnost ne samo potražiti ono što netko ne zna, ali i otkriti stvari koje nitko ne zna [S69]. Programeri će imati znanje za osmišljavanje sljedeće generacije sustava koji će biti implementiran u računalni oblak, općenito, naglasak bi trebao biti na horizontalnoj skalabilnosti tisuće virtualnih strojeva na jednom virtualnom stroju [S70].

6. ZAKLJUČAK

Iako postati poslužitelj usluga u oblaku je uistinu skupo i uzima puno resursa, postoje mnoge prednosti, kao što prikazuje ovaj rad, zbog brzog rasta interesa za programiranje u oblaku. U ovom radu su prikazani stavovi i izazovi arhitekture s gledišta različitih sudionika računalnog oblaka kako bi se pomoglo programerima bolje razumjeti njihove potrebe. Sigurnost je na vrhu popisa pitanja; ali za korisnike oblaka, privatnost podataka je ono što ih najviše zabrinjava. Neka aktualna rješenja u oblaku ne dopuštaju korisnicima migraciju na druge poslužitelje servisa u oblaku, također je često onemogućeno povlačenje svojih podataka jednom pohranjenih. Iako se većina postojećih cloud aplikacija temelji na dvoslojnoj arhitekturi, predstavljene studije prikazuju nedostatak njegovih sposobnosti i predlažu nova rješenja za arhitekturu oblaka (troslojne i višeslojne arhitekture). Rezultati i zaključci u ovom radu će pomoći gospodarstvu i akademskim zajednicama, kao i arhitektima računalnih oblaka jer su sažeta pitanja vezana uz postojeće arhitekture oblaka kao što su: nedostatak prenosivosti, skalabilnost, dostupnost, pitanja sigurnosti i privatnosti itd; i odabrane su upute za osmišljavanje novog oblaka koji je u stanju riješiti nekoliko pitanja atributa kvalitete.

Ovaj diplomski rad je nastao kao dio diplomskog rada pisanog u sklopu Erasmus studentske razmjene na sveučilištu Mälardalens högskola u gradu Västerås, Švedska. Diplomski rad su pisala dva studenta i on objedinjuje sistematski pregled literature na pitanje kako dizajnirati aplikaciju za oblak kao i detaljan pregled postojeće arhitekture oblaka vezan uz glavna pitanja i probleme. Diplomski rad [9] je dobio najveću pohvalu te je objavljen u obliku znanstvenog članka na IEEE Xplore. Popis studija dobivenog nakon izvršenog sistematskog pregleda literature je dan u prilogu kao dodatak ovom radu.

Glavni doprinos ovog rada je identifikacija glavnih problema pri osmišljavanju računalnog oblaku i rješenja postojeće arhitekture za suočavanje s tim različitim pitanjima.

LITERATURA

[1] http://www-935.ibm.com/services/us/gbs/thoughtleadership/ibv-power-of-cloud.html

[2] http://www.yourdigitalspace.com/2010/10/the-amount-of-data-generated-and-consumed-on-the-internet/

[3] http://www.idc.com/getdoc.jsp?containerId=prUS24298013

[4] Peeyush Mathur, Nikhil Nishchal, "Cloud Computing: New challenge to the entire computer industry, 1st International Conference on Parallel, Distributed and Grid Computing" (PDGC 2010), 2010

[5] Mohammad Hamdaqa, Ladan Tahvildari, "Cloud computing uncovered – a research landscape", Advances in Computers, Volume 86, 2012

[6] An Oracle White Paper, "Cloud Reference Architecture", Oracle Enterprise Transformation Solutions Series, 2012

[7] http://www.cert.hr/sites/default/files/NCERT-PUBDOC-2010-03-293.pdf

[8] http://en.wikipedia.org/wiki/Byzantine_fault_tolerance

[9] http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7023596

SAŽETAK

Programiranje u oblaku je u konstantnom porastu kao nova paradigma online usluga vezanih uz razna područja primjene i interesa. No, dizajniranje novih usluga u oblaku kao i dizajniranje usluga u već postojeće poslovne koncepte predstavlja veliki proces u koji treba uključiti puno elemenata. Od velike je važnosti za dizajnere aplikacija u oblaku da znaju koja su glavna pitanja i problemi sudionika usluga u oblaku kao što su korisnici oblaka, pružatelji oblaka i posrednici oblaka. U ovom su radu istraženi glavni izazovi i pitanja tijekom izgradnje aplikacije za oblak kao i različiti arhitekturalni pristupi vezani uz rješavanje spomenutih pitanja. Istraživanje je pokazalo da sigurnost i privatnost podataka predstavljaju najveći problem za korisnike oblaka kao i za pružatelje usluga u oblaku. Ovaj rad daje kompletnu kategorizaciju studija koji opisuju razne pristupe i razmatranja arhitekture prilikom dizajniranja aplikacija za oblak.

Ključne riječi: *programiranje u oblaku, problemi, arhitektura oblaka, dizajniranje oblaka, sigurnost*

ABSTRACT

Cloud Computing has emerged as a new paradigm in the field of network-based services within many industrial and application domains. However, building new services in the cloud or designing cloud-based solutions into existing business context is a complex decision process involving many factors. It is important for the cloud designers to know what are the main concerns of cloud stakeholders such as cloud consumers, cloud providers and cloud brokers. In this study were investigated the main challenges and concerns while building cloud-based architectures and different architectural approaches are proposed to meet these specific concerns. The study revealed that security and dana privacy is on the top of the list of concerns for both cloud consumers and cloud providers. This study gives fully categorized studies that describe architectural approaches and design considerations when architecting for the cloud.

Keywords: *Cloud Computing, concerns, cloud architecture, cloud design, security*

ŽIVOTOPIS

Ivan Balatinac rođen je 13.rujna 1990. godine u Osijeku. U Osijeku završava Osnovnu školu Ljudevita Gaja 2005. godine. Te iste godine upisuje Matematičku Gimnaziju koju završava 2009. godine u Osijeku. Nakon gimnazije upisuje preddiplomski studij računarstva na Elektrotehničkom fakultetu u Osijeku. Preddiplomski studij završava 2012. godine. Iste godine upisuje diplomski studij,smjer procesno računarstvo. Nakon prve godine diplomskog studija odlazi na stručnu praksu u Tokyo, Japan gdje boravi tri mjeseca. Na stručnoj praksi se bavi izradom mobilne aplikacije za prepoznavanje zvuka. 2014. godine odlazi na studentsku razmjenu Erasmus u Švedsku, sveučilište Mälardalens högskola u gradu Västerås gdje piše diplomski rad u suradnji s kompanijom ABB. Te iste godine odlazi i na Erasmus stručnu praksu u Švedsku u trajanju od 7 mjeseci u firmi C/Mind u Stockholmu. Tamo se bavi izradom web aplikacija. Trenutno radi u NBES, startupu u Švedskoj, gdje obnaša funkciju CTO-a.

PRILOG – POPIS STUDIJA

[S1] Alysson Bessani, Paolo Romano, Rüdiger Kapitza, Spyridon V. Gogouvitis, Roberto G. Cascella, Dana Petcu, Dimosthenis Kyriazis, "A look to the old-world_sky EU-funded dependability cloud computing research", SIGOPS Oper. Syst. Rev., Volume 46, Issue 2, July 2012.

[S2] Paul Brebner, "Is your cloud elastic enough performance modelling the elasticity of infrastructure as a service (IaaS) cloud applications", Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering, 2012.

[S3] Siani Pearson, "Taking account of privacy when designing cloud computing services", Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, 2009.

[S4] Abdulrahman A. Almutairi, Muhammad I. Sarfraz, "A Distributed Access Control Architecture for Cloud Computing", Software IEEE, Volume 29, Issue 2, 2012.

[S5] Mohammed Arif Amin, Kamalrulnizam Bin Abu Bakar, Haider Al-Hashimi, "A review of mobile cloud computing architecture and challenges to enterprise users", GCC Conference and Exhibition (GCC), 2013 7th IEEE, November 2013.

[S6] V.P.Krishna Anne, Vidya Sagar Ponnam, Gorantla Praveen, "A significant approach for cloud database using shared-disk architecture", Software Engineering (CONSEG), 2012 CSI Sixth International Conference on, September 2012.

[S7] Manu Annd, "Always On Architecture for High Availability Cloud Applications", Cloud Computing in Emerging Markets (CCEM), 2012 IEEE International Conference On, October 2012.

[S8] Francisco Javier Nieto, "An architecture for a platform providing things as a service", Wireless Personal Multimedia Communications (WPMC), 2013 16th International Symposium on, June 2013.

[S9] Mark Gall, Angelika Schneider, Niels Fallenbeck, “An Architecture for Community Clouds Using Concepts of the Intercloud”, Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on, March 2013.

[S10] Mark Shtern, Bradely Simmons, Michael Smit, Marin Litoiu, “An architecture for overlaying private clouds on public providers”, Network and service management (CNSM), 2012 8th international conference and 2012 workshop on system virtualization management (SVM), October 2012.

[S11] Bassam S. Farroha, Deborah L. Farroha, “Architecting security into the clouds An enterprise security model”, Systems Conference (SysCon), 2012 IEEE International, March 2012.

[S12] E. Casalicchio, L. Silvestri, “Architectures for autonomic service management in cloud-based systems”, Computers and Communications (ISCC), 2011 IEEE Symposium on, July 2011.

[S13] Rajkumar Buyya, Rodrigo N. Calherios, Xiaoronog Li, “Autonomic Cloud computing Open challenges and architectural elements”, Emerging Applications of Information Technology (EAIT), 2012 Third International Conference on, December 2012.

[S14] Liang-Jie Zhang, Qun Zhou, “CCOA Cloud Computing Open Architecture”, Web Services, 2009, ICWS 2009, IEEE International Conference on, July 2009.

[S15] Jing Liu, Liang-Jie Zhang, Bo Hu, Keqing He, “CCRA Cloud Computing Reference Architecture”, Services Computing (SCC), 2012 IEEE Ninth International Conference on, June 2012.

[S16] Huimin Zhang, Xiaolong Yang, “Cloud Computing Architecture Based-On SOA”, Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium on, October 2012.

[S17] Nawsher Khan, A. Noraziah, Tutut Herawan, Elrasheed Ismail, Zakira Inayat, “Cloud Computing Architecture for Efficient Provision of Services”, Network- Based Information System (NBiS), 2012 15th International Conference on, September 2012.

[S18] En NiariSaad, Khalil El Mahdi, Mostapha Zbakh, “Cloud computing architectures based IDS”, Security Days (JNS3), 2013 National, November 2012.

[S19] Ratko Mutavdzic,"Cloud computing architectures for national, regional and local government", MIPRO, 2010 Proceedings of the 33rd International Convention, May 2010.

[S20] Huaglory Tianfield, "Cloud computing architectures", Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on, October 2011.

[S21] Mingwei Wang, Jingtao Zhou, Shikai Jing, "Cloud manufacturing Needs, concept and architecture", Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on, May 2012.

[S22] Pooyan Jamshidi, Aakash Ahmad, Claus Pahl, "Cloud migration research a systematic review", IEEE Transactions On Cloud Computing, Volume 1, Issue 2, December 2013.

[S23] Gurudatt Kulkarni, Rani Waghmare, Rajnikant Palwe, Vidya Waykule, Hemant Bankar, Kundlik Koli,"Cloud storage architecture", Telecommunication Systems, Services, and Applications (TSSA), 2012 7th International Conference on, October 2012.

[S24] Jiehan Zhou, Teemu Leppänen, Erkki Harjula, Mika Ylianttila, Timo Ojala, Chen Yu, Hai Jin, Laurence Tianruo Yang, "CloudThings A common architecture for integrating the Internet of Things with Cloud Computing", Computer Supported Cooperative Work in Design (CSCWD) 2013 IEEE 17th International Conference on, June 2013.

[S25] Kapil Bkshi, "Considerations for cloud data centers Framework, architecture and adoption", IEEEAC paper, March 2011.

[S26] Peterson Gunnar, "Don't Trust. And Verify A Security Architecture Stack for the Cloud", Security & Privacy, IEEE, Volume 8, Issue 5, October 2010.

[S27] Vajihe Lohmosavi, Akbar Farhoodi Nejad, Elham Morad Hosseini, "E-learning ecosystem based on service-oriented cloud computing architecture", Information and Knowledge Technology (IKT), 2013 5th Conference on, May 2013.

[S28] Longji Tang, Jing Dong, Yajing Zhao, Liang-Jie Zhang, "Enterprise Cloud Service Architecture", Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, July 2010.

[S29] Antonio Celesti, Francesco Tusa, Massimo Villari, Antonio Puliafito, “How to Enhance Cloud Architectures to Enable Cross-Federation”, Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, July 2010.

[S30] Yuri Demchenko, Marc X. Makkes, Rudolf Strijkers, Cees de Laat, “Intercloud Architecture for interoperability and integration”, Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on, December 2012.

[S31] Vijay Sarathy, Purnendu Narayan, Rao Mikkilineni, “Next Generation Cloud Computing Architecture Enabling Real-Time Dynamism for Shared Distributed Physical Infrastructure”, Enabling Technologies, Infrastructures for Collaborative Enterprises (WETICE), 2010 19th IEEE International Workshop on, June 2010.

[S32] Madhukara Phatak, Kamalesh. V. N., “On cloud computing deployment architecture”, Advances in ICT for Emerging Regions (ICTer), 2010 International Conference on, October 2010.

[S33] Saira Begum, Muhhamad Khalid Khan, “Potential of cloud computing architecture”, Information and Communication Technologies (ICICT), 2011 International Conference on, July 2011.

[S34] Hongli Luo, Aaron Egbert, Timothy Stahlhut, “QoS architecture for cloud-based media computing”, Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on, June 2012.

[S35] Wei-Tek Tsai,Yu Huang, XiaoyingBai, Jerry Gao, “Scalable Architectures for SaaS”, Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW), 2012 15th IEEE International Symposium on, April 2012.

[S36] Qiu Xiu-feng, Liu Jian-wei, Zhao Peng-chuan, “Secure cloud computing architecture on mobile internet”, Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on, August 2011.

[S37] Wei-Tek Tsai, Xin Sun, Janaka Balasooriya, “Service-Oriented Cloud Computing Architecture”, Infromation Technology: New Generations (ITNG), 2010 Seventh International Conference on, April 2010.

[S38] Samah Ahmed Zaki Hassan, “STAR A proposed architecture for cloud computing applications”, Cloud Computing Technologies, Applications and Management (ICCCTAM), 2012 International Conference on, December 2012.

[S39] Paulo Verissimo, Alysson Bessani, Marcelo Pasin, “The TClouds architecture Open and resilient cloud-of-clouds computing”, Dependable Systems and Networks Workshops (DSN-W), 2012 IEEE/IFIP 42nd International Conference on, June 2012.

[S40] Yanuarizki Amanatullah, Charels Lim, Heru Purnomo Ipung, Arkav Juliandri, “Toward cloud computing reference architecture Cloud service management perspective”, ICT for Smart Society (ICISS), 2013 International Conference on, June 2013.

[S41] Nikolaos Loutas, Vassilios Peristeras, Thanassis Bouras, Eleni Kamateri, Dimitrios Zeginis, Konstantinos Tarabanis, “Towards a Reference Architecture for Semantically Interoperable Clouds”, Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on, December 2010.

[S42] Roger Clark, “User Requirements for Cloud Computing Architecture”, Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on, May 2010.

[S43] Hao Yuan, Xiaoping Liu, Chunhui Guo, “A design of two-tier SaaS architecture based on group-tenant”, Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on, December 2012.

[S44] Gerd Breiter, Vijay K. Naik, “A Framework for Controlling and Managing Hybrid Cloud Service Integration”, Cloud Engineering (IC2E), 2013 IEEE International Conference on, March 2013.

[S45] Jonas Repschlaeger, Stefan Wind, Ruediger Zarnekow, Klaus Turowski, “A Reference Guide to Cloud Computing Dimensions Infrastructure as a Service Classification Framework”, System Science (HICSS), 2012 45th Hawaii International Conference on, January 2012.

[S46] V. D. Cunsolo, S. Distefano, A. Puliafito, M. Scarpa, “Applying Software Engineering Principles for Designing Cloud@Home”, Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on, May 2010.

[S47] Santonu Sarkar, Rajeshwari Ganesan, Manish Srivastava, Sharada Dharmasankar, “Cloud Based Next Generation Service and Key Challenges”, Services in Emerging Markets (ICSEM), 2012 Third International Conference on, December 2012.

[S48] Marty Humphrey, Jacob Steele, In Kee Kim, Michael G. Kahn, Jessica Bondy, Michael Ames, “CloudDRN A Lightweight, End-to-End System for Sharing Distributed Research Data in the Cloud”, eScience (eScience), 2013 IEEE 9th International Conference on, October 2013.

[S49] Teresa Tung, “Defining a Cloud Reference Model”, Cluster, Cloud and Grid Computing (CCGrid), 2011 11th IEEE/ACM International Symposium on, May 2011.

[S50] Yong Zhao, Yanzhe Zhang, Wenhong Tian, Ruini Xue, Cui Lin, “Designing and Deploying a Scientific Computing Cloud Platform”, Grid Computing (GRID), 2012 ACM/IEEE 13th International Conference on, September 2012.

[S51] Bogdan Solomon, Dan Ionescu, Marin Litoiu, Gabriel Iszlai, “Designing autonomic management systems for cloud computing”, Computational Cybernetics and Technical Informatics (ICCC-CONTI), 2010 International Joint Conference on, May 2010.

[S52] Antonio Celesti, Nicola Peditto, Fabio Verboso, Massimo Villari, Antonio Puliafito, “DRACO PaaS A Distributed Resilient Adaptable Cloud Oriented Platform”, Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International, May 2013.

[S53] Giuseppe Carella, Thomas Magedanz, Konrad Campowsky, Florian Schreiner, “Elasticity as a service for federated cloud testbeds”, Communications Workshops (ICC), 2013 IEEE International Conference on, June 2013.

[S54] JhanYuler De la Pava Torres, Claudia Jimenenez-Guarin, “MagosCloud Secure A secure, highly scalable platform for services in an opportunistic environment”, High Performance Computing and Simulation (HPCS), 2012 International Conference on, July 2012.

[S55] Ying Chen, QingniShen, Pengfei Sun, Yangwei Li, Zhong Chen, Sihan Qing, “Reliable Migration Module in Trusted Cloud Based on Security Level - Design and Implementation”,

Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International, May 2012.

[S56] Linping Su, Lin Li, Lu Zhang, Xiaoqian Nie, “Research and Design of Electric Power Private Cloud Data Storage Model”, Computational and Information Sciences (ICCIS), 2012 Fourth International Conference on, August 2012.

[S57] Lawrence Chung, Tom Hill, Owolabi Legunsen, Zhenzhou Sun, Adip Dsouza, Sam Supakkul, “A goal-oriented simulation approach for obtaining good private cloud-based system architectures”, Journal of Systems and Software, Volume 86, Issue 9, October 2012.

[S58] Rui Han, Moustafa M. Ghanem, Li Guo, YikeGuo, Michelle Osmond, “Enabling cost-aware and adaptive elasticity of multi-tier cloud applications”, Future Generation Computer Systems, Volume 32, Issue 0, May 2012.

[S59] Bahman Javadi, Jemal Abawajy, Rajkumar Buyya, “Failure-aware resource provisioning for hybrid Cloud infrastructure”, Journal of Parallel and Distributed Computing, Volume 72, Issue 10, June 2012.

[S60] Georg Lackermair, “Hybrid cloud architectures for the online commerce”, Procedia Computer Science, Volume 3, Issue 0, 2010.

[S61] Azubuike Ezenwoke, Nicholas Omoregbe, Charles Korede Ayo, Misra Sanjay, “NIGEDU CLOUD Model of a National e-Education Cloud for Developing Countries”, IERI Procedia, Volume 4, Issue 0, 2013.

[S62] Dan Petcu, Georgiana Macariu, Silviu Panica, Ciprian Craciun, “Portable Cloud applications—From theory to practice”, Future Generation Computer Systems, Volume 29, Issue 6, January 2012.

[S63] Knud Brandis, Srdan Dzombeta, Knut Haufe, “Towards a framework for governance architecture management in cloud environments A semantic perspective”, Future Generation Computer Systems, Volume 32, Issue 0, September 2013.

[S64] Lizhe Wang, Marcel Kunze, Jie Tao, Gregor von Laszewski, “Towards building a cloud for scientific applications”, Advances in Engineering Software, Volume 42, Issue 9, March 2011.

[S65] Christos Kalloniatis, Haralambos Mouratidis, Manousakis Vassilis, Shareeful Islam, Stefanos Gritzalis, Evengelia Kavakli, “Towards the design of secure and privacy-oriented information systems in the cloud Identifying the major concepts”, Computer Standards & Interfaces, Volume 36, Issue 4, December 2013.

[S66] Won Kim, “Cloud architecture a preliminary look”, 9th International Conference on Advances in Mobile Computing and Multimedia, MoMM, 2011.

[S67] Gianmario Motta, Nicola Sfondrini, Daniele Sacco, “Cloud computing An architectural and technological overview”, 2012 International Joint Conference on Service Sciences, Service Innovation in Emerging Economy: Cross-Disciplinary and Cross-Cultural Perspective, IJCSS 2012, May 2012.

[S68] Mahesh H. Dodani, “The practice of architecting cloud solutions”, Journal of Object Technology, Volume 9, Issue 1, February 2010.

[S69] Rick McGeer, “Transcloud Design considerations for a high-performance cloud architecture across multiple administrative domains”, 1st International Conference on Cloud Computing and Services Science, CLOSER 2011, 2011.

[S70] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia, “Above the Clouds A Berkeley View of Cloud Computing”, Technical Report, February 2009.

[S71] Christoph Fehling, Frank Leymann, Ralph Retter, David Schumm, Walter Schupeck, “An architectural pattern language for cloud-based applications”, October 2011.

[S72] Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee Badger, Dawn Leaf, “NIST Cloud computing reference Architecture”, NIST Special Publication 500-292, July 2011.