



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

Justas Šatrauskas

**ĮMONĖS INFORMACINIŲ TECHNOLOGIJŲ INFRASTRUKTŪROS
PERKĖLIMO Į DEBESŲ KOMPIUTERIJĄ IŠŠŪKIAI**

**THE CHALLENGES OF MOVING THE COMPANY'S IT
INFRASTRUCTURE TO CLOUD COMPUTING**

Baigiamasis magistro darbas

Informacinių technologijų valdymo studijų programa, valstybinis kodas 621I13001

Informatikos studijų kryptis

Vilnius, 2017

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

TVIRTINU
Katedros vedėjas

(Parašas)

(Vardas, pavardė)

(Data)

Justas Šatrauskas

**ĮMONĖS INFORMACINIŲ TECHNOLOGIJŲ INFRASTRUKTŪROS
PERKĖLIMO Į DEBESŲ KOMPIUTERIJĄ IŠŠŪKIAI (LT)**

**THE CHALLENGES OF MOVING THE COMPANY'S IT
INFRASTRUCTURE TO CLOUD COMPUTING (EN)**

Baigiamasis magistro darbas

Informacinių technologijų studijų programa, valstybinis kodas 621I13001

Informatikos studijų kryptis

Vadovas

(Pedag. vardas, vardas, pavardė)

(Parašas)

(Data)

Lietuvių kalbos konsultantas

(Pedag. vardas, vardas, pavardė)

(Parašas)

(Data)

Vilnius, 2017

Vilniaus Gedimino technikos universitetas
Fundamentinių mokslų fakultetas
Informacinių technologijų katedra

ISBN ISSN
Egz. sk.
Data-.....-.....

Antrosios pakopos studijų **Informacinių technologijų valdymo** programos magistro baigiamasis darbas

Pavadinimas **Įmonės IT infrastruktūros perkėlimo į debesų kompiuteriją iššūkiai**
Autorius **Justas Šatrauskas**
Vadovas **Egidijus Ostašius**

Kalba: lietuvių

Anotacija

Baigiamajame magistro darbe nagrinėjami įmonės informacinių technologijų infrastruktūros perkėlimo į debesų kompiuteriją iššūkiai, darantys įtaką migravimo į debesį proceso įgyvendinimui, taip pat jų tarpusavio ryšys.

Pirmoje darbo dalyje trumpai pristatoma debesų kompiuterijos samprata, pagrindiniai migravimo į debesų kompiuteriją paslaugų modeliai (IaaS, PaaS ir SaaS), debesų tipai ir nuosavos programinės įrangos adaptavimo debesų kompiuterijos aplinkoje būdai.

Antroje dalyje analizuojami konkretūs debesų kompiuterijos iššūkiai.

Trečioji dalis skirta parengti sprendimo dėl migravimo priėmimo proceso modeliui ir migravimo į IaaS planui, atsižvelgiant į prieš tai nagrinėtų iššūkių analizę. Darbo rezultatas – nustatyta IT infrastruktūros perkėlimo į debesų kompiuteriją visumos faktorių įtaka šiam procesui ir migravimo į IaaS planavimui. Nagrinėjamas hipotetinis privačios įmonės IT infrastruktūros migravimo į debesies teikėjo platformą atvejis. Pateikiamos išvados.

Darbą sudaro šešios dalys: įvadas, trys skyriai (dalys), išvados, literatūros ir kitų šaltinių sąrašas.

Darbo apimtis – 79 p. teksto be priedų, 22 iliustr., 47 bibliografiniai šaltiniai.

Prasminiai žodžiai: Debesų kompiuterija, daugianaudotojiškumas, hibridinis debesis, hypervizorius, infrastruktūra kaip paslauga, išplečiamumas, migravimo planas, virtuali mašina.

Vilnius Gediminas Technical University
Faculty of Fundamental Sciences
Department of Information Technologies

ISBN ISSN
Copies No.
Date-.....-.....

Master Degree Studies **Information Technology Management** study programme Master Graduation Thesis

Title **The Challenges of Moving Company's IT Infrastructure to Cloud Computing**

Author **Justas Šatrauskas**

Academic supervisor **Egidijus Ostašius**

Thesis language: Lithuanian

Annotation

This master thesis deals with the challenges which emerge when a company moves its IT infrastructure to cloud computing and which affect the implementation of that process, also the interrelationship between the said challenges is being explored.

In the first part of the thesis, the concept of cloud computing is introduced, as well as the main models of migration to the cloud (IaaS, PaaS and SaaS), the types of clouds and the methods of adapting the company's owned software to the cloud environment.

In the second part the specific cloud computing challenges are being examined.

The third part is dedicated to the preparation of the process model, which is used for making the decision on migration to the cloud, and it also explores the migration to the IaaS plan, taking into consideration the previous analysis of the challenges. Results of the thesis reveal the impact of entirety of those factors on the process of the migration to the cloud and on the planning of the IaaS migration as such. The thesis examines a hypothetical case study concerning the migration of a private company's IT infrastructure to a cloud platform. Conclusions are provided.

Structure: introduction, 3 parts, conclusions and references.

Thesis consists of: 79 p. text without appendixes, 22 pictures, 47 bibliographical entries.

Appendixes included.

Keywords: Cloud computing, hybrid cloud, hypervisor, infrastructure as a service, migration plan, multi-tenancy, scalability, virtual machine.

(Baigiamojo darbo sąžiningumo deklaracijos forma)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Justas Šatrauskas, 20150939

(Studento vardas ir pavardė, studento pažymėjimo Nr.)

Fundamentinių mokslų fakultetas

(Fakultetas)

Informacinių technologijų valdymas, ITVmit-15

(Studijų programa, akademinė grupė)

BAIGIAMOJO DARBO (PROJEKTO)

SĄŽININGUMO DEKLARACIJA

2017 m. gegužės 24 d.

Patvirtinu, kad mano baigiamasis darbas tema „Įmonės IT infrastruktūros perkėlimo į debesų kompiuteriją iššūkiai“ patvirtintas 2016 m. gruodžio 20 d. dekanų potvarkiu Nr. 418fm, yra savarankiškai parašytas. Šiame darbe pateikta medžiaga nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Prenkant ir įvertinant medžiagą bei rengiant baigiamąjį darbą, mane konsultavo mokslininkai ir specialistai: dr. Egidijus Ostašius. Mano darbo vadovas dr. Egidijus Ostašius.

Kitų asmenų indėlio į parengtą baigiamąjį darbą nėra. Jokių įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs (-usi).

(Parašas)

Justas Šatrauskas
(Vardas ir pavardė)

Turinys

Įvadas	10
1. DEBESŲ KOMPIUTERIJOS SAMPRATA IR PASLAUGŲ MODELIAI	13
1.1. Debesų kompiuterijos samprata	13
1.2. Debesų kompiuterijos paslaugų modeliai (IaaS, PaaS, SaaS)	13
1.3. Debesies diegimo būdai: privatus, viešas, hibridinis	17
1.4. Nuosavos programinės įrangos adaptavimo debesies aplinkoje būdai.....	18
1.4.1. Programinės įrangos perkėlimas į debesies platformą	18
1.4.2. Programinės įrangos perdėrbimas	21
1.4.3. Programinės įrangos peržiūrėjimas	22
1.4.4. Programinės įrangos pakeitimas debesies paslauga	22
1.4.5. Skirtingų aplinkų integravimas	22
1.4.6. Debesies platformos programinės įrangos įdiegimui	23
2. IT INFRASTRUKTŪROS PERKĖLIMO Į DEBESIES APLINKĄ IŠŠŪKIAI	24
2.1. Iššūkių klasifikavimas	24
2.2. Saugumas ir privatumas debesies aplinkoje	26
2.2.1. Saugumo privalumai, kai programos yra perkeliamos į debesį	27
2.2.2. Saugumo trūkumai migravus programinę įrangą į debesį	28
2.2.3. Saugumo priemonių taikymas debesų kompiuterijoje	30
2.2.4. Duomenų apsauga debesų kompiuterijoje	31
2.2.5. Debesies naudotojo ir tiekėjo atsakomybės pasiskirstymas užtikrinant saugumą	32
2.2.6. DDoS atakos debesyje	33
2.3. Organizaciniai iššūkiai	33
2.4. Teisiniai ir reguliaciniai reikalavimai	36
2.5. Finansiniai iššūkiai	36
2.6. Programinės įrangos suderinamumas	40
2.7. Programinės įrangos veikimo veiksmingumas	42
2.8. Resursų panaudojimo efektyvumas	46
2.8.1. Išplečiamumas (angl. scalability)	47
2.8.2. Elastingumas (angl. elasticity)	48
2.8.3. Daugianaudotojiškumas (angl. multi-tenancy)	49
2.9. Paslaugų kokybės užtikrinimas debesyje	49
2.10. Debesies paslaugų tiekėjo pasirinkimas	51
2.11. Programos migravimo metodo pasirinkimas	53
2.12. Vieningo standarto nebuvimas	54
2.13. Perkeliamumas tarp debesies tiekėjų	54
2.14. Rodiklių matavimas	55
2.15. Migravimo trukmė	57
3. MIGRAVIMO PROCESAS IR PLANO PARENGIMAS	58
3.1. Sprendimas dėl migravimo	58
3.2. Migravimo į debesies aplinką planas	60
3.3. Hipotetinės įmonės IT infrastruktūros perkėlimas į debesų kompiuteriją	67
IŠVADOS	73
LITERATŪROS IR KITŲ ŠALTINIŲ SĄRAŠAS	75
PRIEDAI	79

Paveikslų sąrašas

1.1 pav. Palyginimas tarp organizacijos vietinės aplinkos ir skirtingų debesies paslaugų modelių aplinkų atsižvelgiant į IT infrastruktūros resursų valdymą [23]	14
1.2 pav. Architektūrų palyginimas virtualių mašinų ir konteinerių kuriamos virtualizacijos atvejais [47]	16
1.3 pav. Debesies diegimo rūšys ir organizacijos prioritetai [23]	18
1.4 pav. Pagrindiniai programos migravimo būdai priklausomai nuo to, kiek programa turi būti atnaujinama ir kokiuose debesies paslaugų modeliuose įgyvendinta [19].....	19
1.5 pav. Perkėlimo būdu į IaaS migruojamos programos komponentai ir jų pokyčiai [19]	19
2.1 pav. Su įmonės vykdomos veiklos faktoriais susiję IT infrastruktūros migravimo į DK iššūkiai [29]	24
2.2 pav. Su techniniais faktoriais susiję IT infrastruktūros migravimo į DK iššūkiai [29]	25
2.3 pav. Svarbiausi migravimo į debesies aplinką faktoriai. Apibendrinti 2016 metų skirtingo dydžio organizacijų (tiek pagal darbuotojų skaičių, tiek pagal pajamas) apklausos duomenys [11]	26
2.4 pav. Skirtingi saugumo užtikrinimo aspektai debesies aplinkoje	27
2.5 pav. <i>Microsoft Azure</i> resursų skaičiuoklė pateikia sąmatą, įvertinančią bendrą IT infrastruktūros debesyje naudojimo kainą priklausomai nuo tos infrastruktūros savybių ir parinkčių [33]	39
2.6 pav. <i>Microsoft Azure</i> siūlomos <i>Windows Server Datacenter</i> virtualios mašinos, 2017 m. kovas [33]	40
2.7 pav. Veikimo veiksmingumo santykis su galėjimu išplėsti IT infrastruktūrą [10].....	43
2.8 pav. Platformų palyginimas pagal vidutinį 3 puslapių užkrovimo laiką (per 50 kartų)	45
2.9 pav. Platformų palyginimas pagal užklausų perdavimo laiką tarp galinių taškų (3 p., 50 kartų)..	45
2.10 pav. <i>Clouddorado.com</i> portale atliekama debesies teikėjų paieška pagal nustatytus kriterijus debesies serveriui [13].....	52
2.11 pav. Virtualaus serverio disko skaitymo/rašymo ir įeinančio/išeinančio tinklo srauto parametrų rodikliai <i>Microsoft Azure</i> platformoje [33]	56
2.12 pav. Migravimo į IaaS trukmė pagal migravimo stadijas [11]	57
3.1 pav. Sprendimo dėl programinės įrangos migravimo į debesų kompiuteriją priėmimo procesas	59
3.2 pav. Esamos programinės įrangos architektūros modelio rengimas	63
3.3 pav. Potencialaus programinės įrangos DK architektūros modelio rengimas	63
3.4 pav. Programos kodo transformavimas	65
3.5 pav. Programinės įrangos sluoksnių talpinimas vienoje VM [33].....	69

Santrumpų sąrašas

AES	Šifravimo algoritmas, kuriame naudojamas iki 256 bitų raktas
API	Sąsaja tarp programų, ne tarp naudotojo ir programos (angl. <i>application programming interface</i>)
App	Programa, programėlė (angl. <i>application</i>)
Capex	Išlaidos kapitalui
CPU	Kompiuterio procesorius
CRM	Santykių su klientais valdymo sistema
DB	Duomenų bazė
DDoS	Paskirstytas paslaugos teikimo nutrūkimas (angl. <i>distributed denial of service</i>)
DK	Debesų kompiuterija
ERP	Įmonės resursų planavimo sistema
IaaS	Infrastruktūra kaip paslauga (angl. <i>infrastructure as a service</i>)
IOPS	Įvesties/išvesties operacijos per sekundę
ISO/IEC	Tarptautinės standartizavimo organizacijos ir Tarptautinės elektrotechninės komisijos (standartas)
IT	Informacinės technologijos
MSA	Bendrasis susitarimas dėl paslaugų (angl. <i>master service agreement</i>)
QoS	Paslaugų kokybė
Opex	Veiklos išlaidos
OS	Operacinė sistema
PaaS	Platforma kaip paslauga (angl. <i>platform as a service</i>)
PHP	Programavimo kalba, pritaikyta interneto svetainių kūrimui (angl. <i>hypertext preprocessor</i>)
PI	Programinė įranga
RAM	Kompiuterio operatyvioji atmintis
SaaS	Programinė įranga kaip paslauga (angl. <i>software as a service</i>)
SLA	Susitarimas dėl paslaugų kokybės (angl. <i>service level agreement</i>)
SSL	Saugumo technologija, šifruojanti duomenų perdavimą tarp interneto serverio ir interneto naršyklės

TCA Bendrosios įsigijimo išlaidos (angl. *total cost of acquisition*)
TCO Bendrosios nuosavybės turėjimo išlaidos (angl. *total cost of ownership*)
VM Virtuali mašina
VPN Virtualus privatus tinklas
Web Pasaulinis interneto tinklas

Įvadas

Informacinių technologijų srityje debesų kompiuterija jos autorių dažnai yra pavadinama paradigmos posūkiu (angl. paradigm shift) dėl jos lemtingos įtakos visoms su technologijomis susijusioms gyvenimo sritims pradedant elektroninio pašto dėžute ir baigiant įmonės veiklos perkėlimu ir vykdymu debesyje. Tai - posūkis nuo kompiuterijos, kaip produkto turėjimo, į kompiuteriją kaip paslaugą, kuri suteikiama naudotojams internetu iš didelės apimties duomenų centrų – ar „debesų“ (Khajeh-Hosseini *et al.* 2012)

Įmonės informacinių technologijų paslaugų perkėlimas į debesį turi daug privalumų, tokių kaip veiklos efektyvinimas, kaštų mažinimas ar centralizuotas resursų valdymas. Vis dėlto šiame procese išlieka daug netikrumo ir dviprasmiškumo, susijusio su migravimo į debesį realizavimu, bei kartais nepamatuoto entuziazmo ir šio proceso iššūkių ignoravimo (Oliveira *et al.* 2014).

Organizacijos turimos programinės įrangos migravimas į debesų kompiuteriją (toliau – DK) reikalauja ją pritaikyti naujai kompiuterijos terpei. Šiuo metu DK paslaugų išplėtojimas suteikia daugiau lankstumo ir geresnę programinės įrangos migravimo kontrolę, leidžia į debesį perkelti atskirus programinės įrangos sluoksnius ar netgi atskirus architektūros komponentus. Tai lemia, kad IT turi persiorientuoti iš nesusieto programinės įrangos diegimo, esant tik minimaliam suderinamumui skirtingose aplinkose, į programų diegimą iš anksto nustatytose standartizuotose platformose, kur egzistuoja abiejų pusių suderinti susitarimai dėl paslaugų kokybės (SLA).

IT infrastruktūros perkėlimo į debesų kompiuteriją procesu įmonė visiškai ar iš dalies atsisako nuosavos fizinės įrangos (serverių, kitų kompiuterijos resursų), *tačiau išlaiko iki tol turėtą programinę įrangą ir ją siekia paleisti jau iš debesies aplinkos, dažniausiai naudojant Infrastruktūros kaip paslaugos (IaaS) modelį.* Šis modelis pasižymi tuo, kad įmonė – kompiuterijos resursų naudotoja, nevaldo ir nekontroliuoja po jais esančios DK fizinės infrastruktūros, bet išlaiko kontrolę operacinėms sistemoms, duomenų talpinimui, perkeltoms programoms ir dalinę kontrolę pasirinktiems tinklo komponentams. IaaS tikslas yra suteikti lanksčią, standartizuotą ir virtualizuotą aplinką įmonės nuosavų programų įdiegimui ir paleidimui.

Nors IaaS yra naudingas ir patrauklus migravimo modelis, pats IT infrastruktūros migravimo procesas yra gana sudėtingas ir kelia nemažai klausimų: Kaip įmonei sklandžiau perkelti savo IT infrastruktūrą į debesies aplinką? Kokie iššūkiai laukia perkeltiant IT infrastruktūrą į DK ir ar galimas vienas migravimo planas visiems atvejams?

Pasak Pooyan Jamshidi *et al.* (2016) dabartiniai migravimo metodai ir literatūroje siūlomi migravimo planai yra arba paruošti žiūrint iš labai plataus strateginio taško, arba labai detalūs ir techniniai, netinkami suprasti tiems, kurie neturi techninio pasiruošimo. Kiti autoriai, kaip Patricia Bessera *et al.* (2012), nurodo, kad iš prigimties DK sprendimai yra sudėtingi, nes yra nulemti daugybės, galimai prieštaraujančių vieni kitiems faktorių, tokių kaip kaštai, programinės įrangos veikimo veiksmingumas (angl. *performance*), saugumas, išplečiamumas (angl. *scalability*), teisiniai klausimai, techniniai reikalavimai ir pan. Kaip vienas iš tokių faktorių taip pat yra ir platus spektras siūlomų DK paslaugų bei diegimo modelių. Mahdi Gholami *et al.* (2016) migravimo į DK analizėje nurodo, kad trūksta suvienodinto migravimo į DK proceso modelio. Vasilios Andrikopoulou *et al.* (2013) pažymi, kad loginis ir fizinis programos duomenų bei kompiuterijos išskirstymas debesyje iškelia daug ekonominių, teisinių ir veikimo veiksmingumo klausimų.

Atsižvelgiant į literatūros analizę matyti, kad mėginama rasti optimalų migravimo į DK modelį, pritaikomą visais ar daugeliu migravimo atvejų. Tačiau, kaip mini nemažai autorių, rengiantis migravimui iškyla problema, jog pats migravimas neatrodo esanti standartizuota procedūra ta prasme, kad galėtų būti vienodai taikoma visiems atvejams. Darytina prielaida, kad kaskart egzistuoja tam tikros aplinkybės, faktoriai, kurie gali priklausyti nuo pasirinkto migravimo modelio (IaaS, PaaS ar SaaS), programos adaptavimo debesies aplinkoje metodų, nuo įmonės naudojamos programinės įrangos, tvarkomų duomenų jautrumo ir standartų, apibendrintai - nuo migravimo iššūkių. Šie iššūkiai galimai daro įtaką tiek pačiam apsisprendimui, ar pradėti migraciją, tiek planui, kaip tą migravimą atlikti. Dalis iššūkių yra potencialiai keliantys riziką (saugumas, kontrolės praradimas), gi dalis jų yra privalumas (resursų panaudojimo efektyvumas, paslaugų prieinamumas), tačiau visi šie iššūkiai turėtų būti įvertinti ir, kiek įmanoma, suvaldyti.

Darbo tikslas. Šiame darbe analizuojant skirtingus IT infrastruktūros migravimo į debesų kompiuteriją iššūkius ir bus bandoma nustatyti jų įtaką migravimui, ir atsakyti į klausimą, koks turėtų būti sklandaus įmonės informacinių technologijų infrastruktūros migravimo į debesų kompiuteriją planas.

Siekiant šio darbo tikslo, iškelti tokie **uždaviniai**:

- Migravimo į debesų kompiuteriją paslaugų modelių ir jų įgyvendinimo būdų analizė.
- IT infrastruktūros migravimo į debesų kompiuteriją iššūkių analizė.
- Migravimo plano parengimas siekiant nustatyti:
 - Ar į įmonę turėtų migruoti savo programinę įrangą į debesies aplinką?

- Kokius žingsnius atlikti, kad įgyvendinti sklandų migravimą į debesų kompiuteriją?

Darbe naudojami šie **tyrimo metodai**, kurie padeda spręsti išsikeltus uždavinius:

- Informacijos šaltinių analizė (mokslinė literatūra, statistika, paslaugos tiekėjų interneto svetainių ir specializuotų interneto portalų peržiūra);
- Migravimo į DK paslaugų modelių, programos migravimo įgyvendinimo būdų palyginamoji analizė;
- Hipotetinės įmonės IT infrastruktūros migravimo į DK atvejo nagrinėjimas.

Magistro darbas suskirstytas į tris pagrindines dalis. Pirmoje dalyje trumpai pristatoma debesų kompiuterijos samprata, migravimo į debesų kompiuteriją paslaugų modeliai. Čia pateikiama informacija apie galimus debesų tipus, nuosavos programinės įrangos įgyvendinimo (adaptavimo) DK aplinkoje būdus bei tiekėjų platformas.

Antroje dalyje analizuojami debesų kompiuterijos iššūkiai ir nustatomi faktoriai, kurie turi įtakos įmonės IT infrastruktūros migravimo į IaaS procesui ir plano parengimui.

Trečioje dalyje parengiamas IT infrastruktūros migravimo į IaaS planas, kuriame išdėstomos migravimo proceso stadijos ir žingsniai. Nagrinėjamas hipotetinis privačios įmonės IT infrastruktūros migravimo į DK platformą atvejis.

Darbo praktinė vertė. Pateikiama IT paslaugų migravimo į debesų kompiuteriją iššūkių analizė ir migravimo į IaaS planas gali padėti įmonei iš anksto įvertinti ir numatyti ne vien tik techninius, bet ir organizacinius, saugumo, resursų efektyvumo ir kitus faktorius ir į juos tinkamai atsižvelgus, kuo sklandžiau savo IT infrastruktūrą perkelti į debesį. Darbas taip pat galėtų pagelbėti tyrėjams, ieškantiems universalaus migravimo plano.

1. Debesų kompiuterijos samprata ir paslaugų modeliai

1.1. Debesų kompiuterijos samprata

Debesų kompiuterija (angl. *cloud computing*) – terminas, atsiradęs palyginti neseniai dėl poreikio tiekti kompiuterijos išteklius kaip paslaugą ir turintis nemažai apibrėžimų. Pagal vieną iš jų debesų kompiuterija – „modelis, įgalinantis patogią ir pagal pareikalavimą tinklo prieigą prie bendros ir konfigūruojamos kompiuterijos resursų pasiūlos (pvz., tinklų, serverių, duomenų talpyklų, programų ir paslaugų), kurie gali būti nedelsiant pradėti arba nutraukti teikti esant minimaliam valdymo ar paslaugos teikėjo įsikišimui” (Khajeh-Hosseini *et al.* 2012).

NIST (Nacionalinis standartų ir technologijos institutas, JAV) yra pateikęs šias penkias esmines savybes, apibūdinančias debesų kompiuteriją (Loeffler 2011):

- Paslaugos suteikimas pagal naudotojo pareikalavimą (angl. *on-demand self-service*).
- Plati tinklo prieiga (angl. *broad network access*).
- Resursų akumuliavimas (angl. *resource pooling*).
- Staigus elastingumas (angl. *rapid elasticity*).
- Išmatuojamos paslaugos.

Kai kurie autoriai šalia šių savybių mini ir daugianaudotojiškumą (angl. *multi-tenancy*), (priskirtinas resursų akumuliavimui), prieigą iš bet kur (angl. *ubiquitous access*), o taip pat ir atsparumą gedimams (angl. *resiliency*) (Erl *et al.* 2013).

Kiekvienas debesų kompiuterijos paslaugos sprendimas apima tam tikras galimybes, kurios paprastai gali būti grupuojamos į tris pagrindines rūšis: infrastruktūra, platforma ir programinė įranga.

1.2. Debesų kompiuterijos paslaugų modeliai (IaaS, PaaS, SaaS)

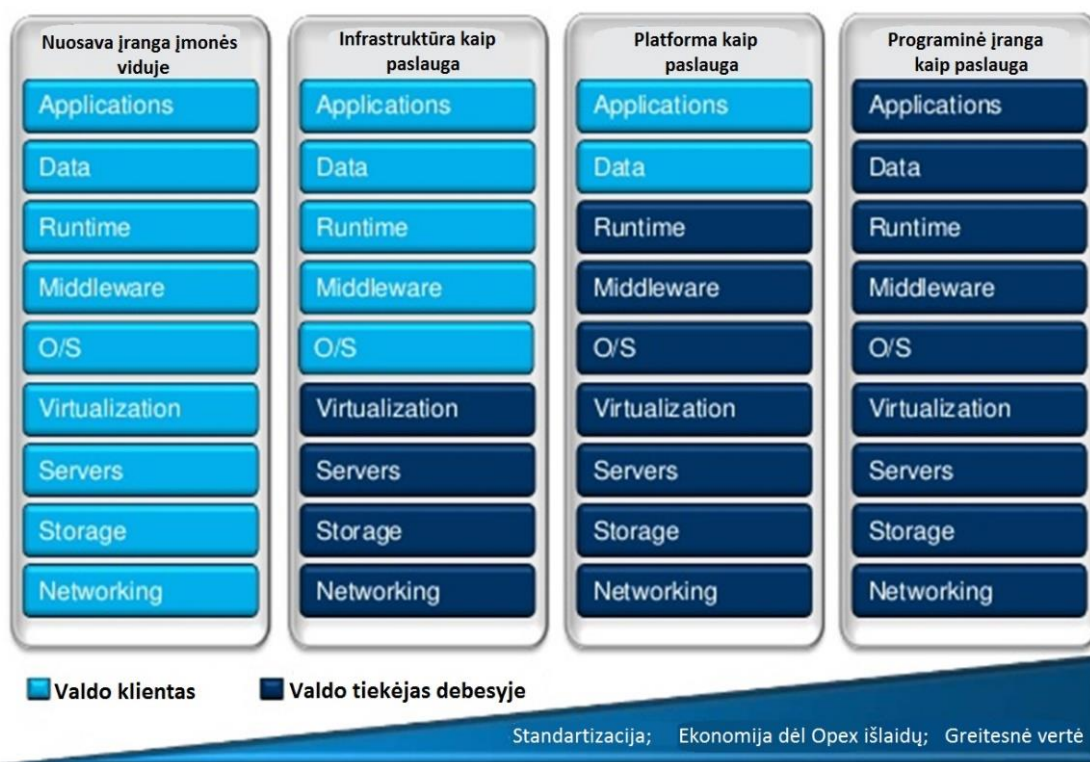
IT **infrastruktūros** perkėlimo galimybė yra susijusi su atitinkama debesų kompiuterijos paslaugos kategorija, siūlančia apdorojimą (angl. *processing*), duomenų talpinimą, tinklo resursus bei kitus esminius kompiuterijos resursus. **Platformos** perkėlimo galimybė siūlo galimybę įdiegti, valdyti ir paleisti naudotojo sukurtą ar naudotojo įsigytą programinę įrangą naudojant programavimo kalbas, o taip pat DK tiekėjo palaikomas įgyvendinimo aplinkas (angl. *execution environments*). **Programinės įrangos** galimybė susijusi su galimybe naudoti programas, palaikomas debesų kompiuterijos paslaugų teikėjo (Erl *et al.* 2013).

IaaS (infrastruktūra kaip paslauga) yra DK modelis, pagrįstas prielaida, kad visa infrastruktūra yra įgyvendinama „pagal pareikalavimą“ (angl. *on demand*) modelį. IaaS dažnai laikoma išteklių

kompiuterija, nes ji traktuoja kompiuterijos resursus panašiai kaip yra traktuojami ištekliai (pvz., elektra). Kai paklausa pajėgumams išauga, tiekėjas pateikia daugiau kompiuterijos resursų. Kai paklausa pajėgumams sumažėja, atitinkamai sumažėja ir kompiuterijos resursai. Šis modelis beveik visada pateikiamas virtualizuotoje formoje, įgalinančioje naudotoją įdiegti kaip komponentus virtualiąsias mašinas, kurios yra valdomos DK konsolės ir kitų įrankių pagalba. Fiziniai resursai – serveriai, talpyklos, tinklas, kartais ir operacinė sistema (OS) yra tiekiami DK tiekėjo, o programinė įranga yra įdiegta ant viršaus minėtų komponentų ir yra valdoma naudotojo (žr. 1.1 pav). Naudotoją šiuo atveju paprastai atstovauja IT specialistų, nusimanančių atskiruose komponentuose, komanda.

IaaS iš DK modelių pasižymi tuo, kad naudotojui suteikia didžiausią debesies aplinkos konfigūravimo ir valdymo kontrolę; jo atsakomybei paprastai priklauso nuosavos operacinės sistemos, programinės įrangos ir duomenų bazių veikimas ir konfigūracija. Pagrindinis IaaS resursas – virtualus serveris yra apibūdinamas jo procesoriaus galia, operatyvine atmintimi ir disko talpa.

IaaS, PaaS, SaaS - kas valdo resursus?



1.1 pav. Palyginimas tarp organizacijos vietinės aplinkos ir skirtingų debesies paslaugų modelių aplinkų, atsižvelgiant į IT infrastruktūros resursų valdymą [23]

Migravimas į debesies aplinką. Lengviausia migruoti programą į IaaS, kai PĮ migruojama kaip visuma: programos kodas kartu su šį kodą palaikančiu kodu – įskaitant ir operacinę sistemą (Cloud Standards Customer Council 2014). Tam turi būti įmanoma supakuoti visą programinės įrangos visumą (angl. *software stack*) kaip vieną ar daugiau virtualios mašinos (toliau – VM) kopijų, tuomet jas perkelti į debesies aplinką ir paleisti. Ar šis migruojamas PĮ vienetas veiks VM aplinkoje gali priklausyti nuo to, ar nuosavai PĮ buvo naudojami specialūs įrenginių valdikliai (angl. *device drivers*) bei fiziniai įrenginiai (angl. *hardware devices*), kurių IaaS tiekėjas nepalaiko. Programa, turinti šių savybių, nebus gera kandidatė migracijai.

IaaS palaiko dinaminį valdymo lygmenį, leidžiantį išplečiamumą pagal pareikalavimą ir automatinį skaičiavimo resursų tiekimą. Virtualizuotos architektūros atveju (kaip minėta, populiariausias IaaS atvejis) tai apima automatinį įdiegimą ir naujų kopijų (angl. *instances*) paleidimą virtualioje mašinoje.

Virtualizacija – procesas, leidžiantis fizinį IT resursą paversti virtualiu IT resursu. Virtualizuoti galima daugelį IT resursų (Erl *et al.* 2013):

- Serveriai – fizinis serveris į virtualų serverį.
- Talpinimas – fizinis talpinimo įrenginys į virtualųjį talpinimo įrenginį ar virtualų diską.
- Tinklas – fiziniai maršrutizatoriai ir perjungėjai į loginę tinklo struktūrą, pvz., VLAN.
- Energijos tiekimas – fizinis UPS ir galios paskirstymo vienetas į virtualų UPS.

Virtualizacijos PĮ sluoksnis leidžia fizinei IT infrastruktūrai patiekti daug virtualių savo kopijų turint tikslą, kad šios infrastruktūros teikiamomis kompiuterijos galimybėmis galėtų pasidalinti daugybė naudotojų. Prieš virtualizacijos technologijų paplitimą programinė įranga būdavo apribota ir tiesiogiai susieta su statiška fizinės infrastruktūros aplinka. Su virtualizacija ši programinės-fizinės įrangos priklausomybė yra nutraukiama, nes dabar fizinės įrangos savybės gali būti imituojamos emuliacijos programinės įrangos pagalba veikiančiose virtualizuotose aplinkose. Ši emuliacijos PĮ vadinama virtualios mašinos valdytoju arba *hypervisoriumi*. Naujieji virtualizuoti IT infrastruktūros resursai tampa suderinami su daugeliu operacinių sistemų, o tų resursų gausinimas daug paprastesnis.

Programą perkeltiant į virtualiąją infrastruktūrą, svarbu migruoti darbą (angl. *workload*) konfigūracijas, o vėliau darbus įdiegti naujoje aplinkoje - debesyje. Konfigūracijos apibūdina darbų komponentų priklausomybes ir apribojimus, o taip pat darbų įdiegimo ant po jais esančios fizinės infrastruktūros reikalavimus. Tad iš esmės infrastruktūros lygio migravimas yra konfigūracijų perkėlimas, kad darbus būtų galima įdiegti ir paleisti naujajame debesyje (Sun, Li 2013).

Konteinerizacija (angl. *containerization*) yra konteinerių pagrindu sukuriama virtualizacija, kuri naudoja branduolį debesies tiekėjo serverio (host) operacinėje sistemoje, tikslu paleisti izoliuotas svečių kopijas (konteinerius). Tam atlikti gali būti naudojamas šiuo metu populiarėjantis *Docker* kompanijos variklis. Izoliuotoji virtuali paleidimo aplinka, kurią sukuria šio tipo virtualizacija, ir yra vadinama konteineriu bei gali būti suvokiama kaip grupė susijusių procesų. Skirtingai nuo VM, konteineriai neapima visos OS – užtenka tik bibliotekų ir konfigūracijos nustatymų, skirtų PĮ paleidimui (žr. 1.2 pav).

Kompiuterija be serverių (angl. *serverless computing*), dar žinoma „funkcija kaip paslauga“ (FaaS) vardu, yra DK įgyvendinimo procesas, kur tiekėjas visiškai valdo tam tikros kliento sistemos funkcijos konteinerio platformos (PaaS) paleidimą/sustabdymą, kai reikia apdoroti konkrečią užklausą (mikro paslaugą). Paslaugos apmokestinamos išmatavus resursų tai užklausai atlikti panaudojimą, o ne už virtualios mašinos naudojimą tam tikrą valandų skaičių. Šio modelio pavadinimas kilęs iš to, kad asmeniui, valdančiam sistemą ir norinčiam paleisti programos kodą, nereikia specialiai įsigyti ar išsinuomoti serverių ar virtualių mašinų (Serverless Computing 2017). Šiame darbe pagrindinis dėmesys bus skirtas VM virtualizacijai, o konteinerizacija išsamiau nebus nagrinėjama.



1.2 pav. Architektūrų palyginimas virtualiųjų mašinų ir konteinerių kuriamos virtualizacijos atvejais [47]

PaaS (platforma kaip paslauga). Naudotojui suteikiama galimybė paleisti debesies aplinkoje savo sukurtą ar įsigytą PĮ, kuri sukurta debesies tiekėjo palaikoma kalba ir įrankiais. Paprastai naudotojas nevaldo debesies infrastruktūros (serverių, tinklo, OS ar talpinimo), bet išlaiko kontrolę debesyje paleistoms savo programoms (Loeffler 2011) (žr. 1.1 pav). PaaS siūloma paruošta naudojimui sukonfigūruota aplinka (pvz., PHP platforma PaaS aplikacijų kūrimui) leidžia DK naudotojams iškart kurti ir paleisti veikti standartinę PĮ, ypač interneto programas (Erl *et al.* 2013). Pastaroji apibrėžiama kaip programa, patalpinta nutolusiame interneto serveryje ir patiekama naudotojui interneto naršyklės sąsajos pagalba (Techterms 2017). Pažymėtina, kad riba skirianti PaaS ir IaaS modelius gali būti labai neryški.

SaaS (programinė įranga kaip paslauga). Šiame modelyje visos IT infrastruktūros valdymo funkcijos yra abstrahuojamos atskirai nuo naudotojo, todėl čia jis paprastai neturi jokios įtakos aplinkos ar programos konfigūracijai, o perka programinę įrangą kaip galutinį produktą iš debesies teikėjo (žr. 1.1 pav). Šios programos paprastai yra pasiekiamos naudotojų galiniais įrenginiais jų sąsajų (pvz., interneto naršyklės) pagalba (*Gmail*, CRM programos ir pan.).

Darytina išvada, kad iš debesies paslaugų modelių didžiausias valdymo ir PĮ konfigūravimo galimybes suteikia IaaS, o mažiausias SaaS, tačiau kontrolės atžvilgiu niekas neprilygsta nuosavos IT infrastruktūros turėjimui.

1.3. Debesies diegimo būdai: privatus, viešas, hibridinis

Atsižvelgiant į IT infrastruktūros migravimo į DK prioritetus organizacijoje, gali būti svarbu įvertinti, kokį debesies diegimo būdą pasirinkti: privatų, viešą ar hibridinį. Prioritetai gali būti įvairūs: informacijos sauga ir privatumas, PĮ svarba įmonės kasdienėje veikloje, IT resursų poreikis ir kt.

Privatus debesis – debesies infrastruktūra skirta naudotis išimtinai vienai organizacijai ir valdoma arba pačios organizacijos, arba trečiojo asmens (debesies tiekėjo). Kaip porūšis egzistuoja bendruomenės debesis – išimtinai kelių konkrečių organizacijų bendrai naudojamas debesis.

Viešas debesis – paslaugų teikėjo valdomas debesis, kurio paslaugomis gali naudotis iš esmės visi norintys ar bent didžioji dalis subjektų.

Hibridinis debesis – debesies infrastruktūra, sudaryta iš dviejų ar daugiau atskirai veikiančių debesų (pvz., viešojo ir privataus), turinčių bendrą sąsają, įgalinančią tiek duomenų, tiek programų perkeliamumą (angl. *portability*) tarp šių debesų.

1.3 pav. pateikiami skirtingų debesies diegimo rūšių privalumai priklausomai nuo įtakos organizacijos veiklos prioritetams (Cloud Standards Customer Council 2014).

Faktoriai	Privatus (nuosavas)	Privatus (debesies tiekėjo)	Viešas
Debesies paslaugų tiekimo svarba	Privatus (nuosavas) debesys tinkamas organizacijos pagrindiniam veiklos tikslui, saugumui jautrioms paslaugoms	Privatus (tiekėjo) debesys tinkamas organizacijos pagrindiniam veiklos tikslui, saugumui jautrioms paslaugoms	Vieši debesys yra labiau skirti paslaugoms, kurios nėra gyvybiškai svarbios organizacijos veikloje ir nereikalaujančioms prieigos prie jautrios informacijos
Vykdomų darbų pobūdis	Privatus (nuosavas) debesys tinkamas programoms, turinčioms griežtus reikalavimus dėl delsos	Privačiam (tiekėjo) debesiui tinkami darbai yra panašūs į viešiams debesims tinkamus	Vieši debesys yra tinkami darbams, reikalaujantiems prieigos prie didelės apimties duomenų (pvz., analizė realiu laiku, vykdoma itin didelėse duomenų talpyklose)

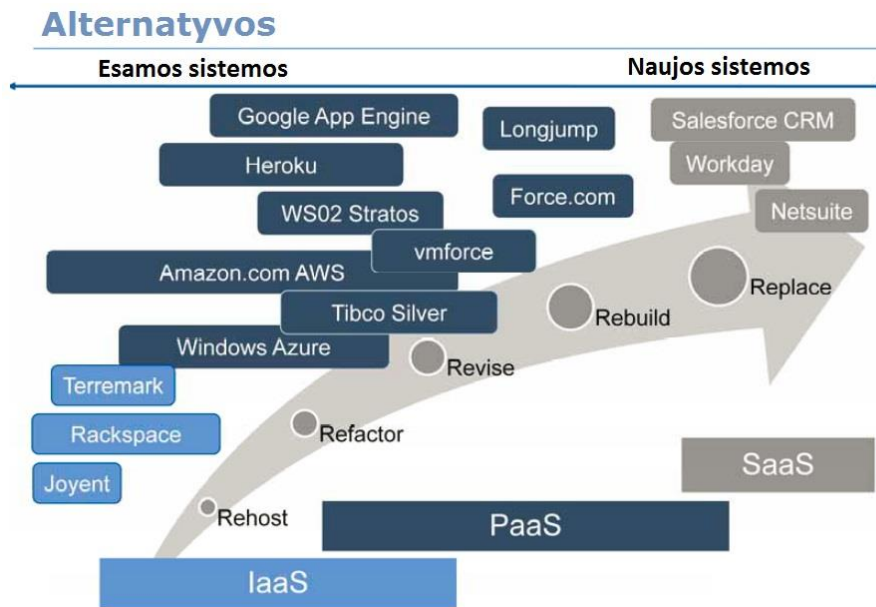
1.3 pav. Debesies diegimo rūšys ir organizacijos prioritetai [23]

1.4. Nuosavos programinės įrangos adaptavimo debesies aplinkoje būdai

Organizacijos, migruojančios į debesų kompiuteriją ir pasirinkusios išlaikyti bei toliau naudoti turimą programinę įrangą, turi keletą galimybių ją adaptuoti debesies aplinkai (Andrikopoulou *et al.* 2013). Įmonė turėtų apsispręsti, kokį **migravimo įgyvendinimo metodą (būdą)** pasirinkti savo perkeliama programai, ypatingai atsižvelgdama į tai, kiek ji pasirošusi modifikuoti turimą programą ir jos komponentus. Šiuo požiūriu išskiriami 3-5 nuosavos PĮ įgyvendinimo debesies aplinkoje būdai (žr. 1.4 pav).

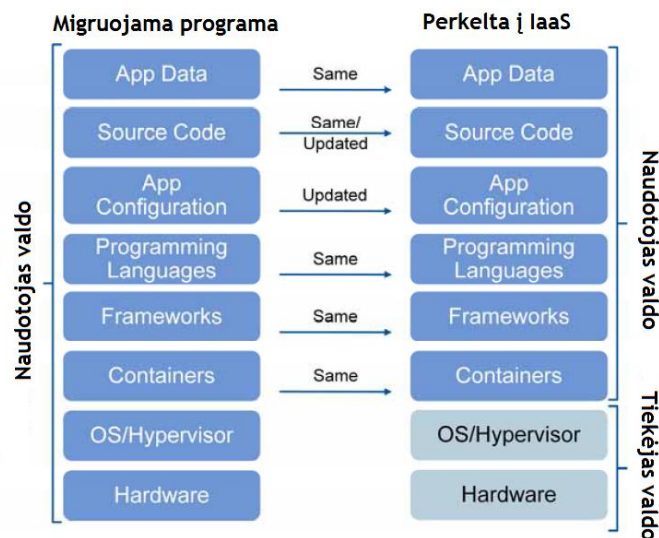
1.4.1. Programinės įrangos perkėlimas į debesies platformą

Perkėlimas (angl. *rehosting, redeployment*, kartais vartojami terminai *emulation*, taip pat *lift and shift*) – programa (jos komponentas) į debesies platformos virtualiąją mašiną yra perkeliama tokia, kokia yra, pakeičiant tik programos infrastruktūros konfigūravimą (Jamshidi *et al.* 2016) (žr. 1.5 pav). Padidėja apsauga nuo gedimų, atsiranda palaikymas, pats įdiegimas paprastesnis, nes vykdomas dideliais programos gabalais arba kaip vienetas. Privalumas, kad tinka PĮ, kur programos kodo



1.4 pav. Pagrindiniai programos migravimo būdai priklausomai nuo to, kiek programa turi būti atnaujinama ir kokiuose debesies paslaugų modeliuose įgyvendinta [19]

pakeitimas neįmanomas. Perkeliama programa kaip visuma (angl. *application stack*) arba susijusių programų visuma. Išlaikoma buvusi architektūra, tačiau tai riboja programos perkeliamumą tarp platformų, taip pat išlieka nevisas išplečiamumas, kol programos architektūra nėra pakeičiama. Perkėlimas automatinis arba rankinis.



1.5 pav. Perkėlimo būdu į IaaS migruojamos programos komponentai ir jų pokyčiai [19]

a) **Emuliacija**, kaip perkėlimas į DK. Po migracijos duomenys ir programos debesyje atrodo taip pat, kaip ir iki perkėlimo. Iš esmės funkcijos ir veiksmai emuliuojami į debesį ir paleidžiami kaip virtualios mašinos darbai.

Naudodama virtualizacijos kūrimo programinę įrangą kaip *VMware* ar *VirtualBox*, įmonė gali sukurti nebrangią ir lanksčią sistemą, kuri, be to, dar leidžia perkelti programas iš vieno tiekėjo aplinkos į kito aplinką arba net ir atgal į nuosavą įmonės sistemą (su sąlyga, kad yra palaikomos pasirinktos VM). Nors ir paprastas būdas, tačiau vienas jo trūkumų yra būtinybė turėti įdiegtas virtualias mašinas. Be to, programa nebus „kilusi iš debesies“ (angl. *cloud-native*), nes debesis yra abstrahuotas atskirai nuo programos, tad yra tikimybė, kad to paties PĮ administravimo ir veikimo veiksmingumo, kaip laikant ją įmonėje, neturėsime.

b) **Lift and shift** perkėlimas, dar vadinamas „kėlimas šakute“ (angl. *forklifting*). Šiuo būdu turimos virtualios mašinos yra perkeliamos į debesį arba 1) rinkmenų lygmens pagrindu, arba 2) atliekant turinio atvaizdo nuotrauką (angl. *volume snapshot*) ir pristatant dalelėmis. Naujoje platformoje virtualios mašinos rekonstruojamos iš failų ir atvaizdų nuotraukų. Šis metodas taupo laiką ir išlaidas. Galima naudotis nemokamais ir/ar atviro kodo įrankiais kaip AWS migravimo įrankis arba *Xeserver*, tokiu būdu automatizuojant migracijas ir nesukeliant PĮ veikimo pertrūkių (nors jų visiškai išvengti gali nepavykti, naudojantis šiuo metodu pertrūkių daug mažiau nei *re-instaliuojant*, *re-architektūrinant* ar *orkestruojant*). Šis metodas sukuria galimybę gyvai migracijai (angl. *live migration* arba *streaming*), kai migravimo procesas nepertraukia programos veikimo.

Trūkumai: perkeliant tiek failus, tiek atvaizdus po dalelę užtrunkama. Duomenų konversijos procesas ne visada einasi sklandžiai ir kai kurie failai – ypač duomenų bazės ir užrakinti failai – nekonvertuojami. Jeigu vyksta sinchronizacija failų pagrindu, gali atsirasti nesuderinamumas dėl negalėjimo dirbti su duomenų baze ar užrakintu failu. Palaikymo ir replikavimo įrankiai gali būti nepakankami, kad pasiekti norėtą funkcionalumą ir išspręsti duomenų skirtumus.

c) **Orkestravimas**. Šio metodo privalumas tas, kad perkeliama programa įdiegiama greitai, galima naudotis nemokamais įrankiais. Kaip trūkumą galima paminėti daug užklausų (angl. *scripts*) rašymo, PĮ veikimo pertrūkių (Cloudendure 2016). Rizikingesnis būdas, nes žalos kontrolė gali būti itin brangi.

Sukurama infrastruktūra, kuri automatiškai norimu būdu sutvarko informaciją. Debesies *orkestravimas* įgalina lankstumą ir suderinamumą bei yra greitai paleidžiamas (ypač, jeigu lyginti šį

metodą su *re-instaliavimu* ar *re-architektūra*, arba su *lift and shift* metodu). Nemažą dalį viso proceso galima automatizuoti.

Šį metodą verslas galėtų rinktis, jeigu perkeliama duomenys yra nesudėtingi, o užklausų dažnai nereikia keisti. Pagrindinis sunkumas yra proceso pradžioje, kol visos užklausos testuojamos, užbaigiamos ir paleidžiamos, o toliau vykdomas tik informacijos teikimas sistemai.

1.4.2. Programinės įrangos perdirbimas

Re-architektūrinimas (angl. *rearchitecting*) reiškia visišką naują PĮ pritaikymą ir galimybę ją modernizuoti. Šis metodas geriausiai veikia su maža sistema. Tačiau visiškai naujos sistemos sukūrimas ne tik reiklus darbui ir brangus, bet sukelia ir verslo veiklos pertrūkius (sistemų išjungimas, personalo trūkumas). Tai iš esmės naujos programinės įrangos kūrimas pasinaudojant PaaS modeliu; procesas pareikalaus testavimo, klaidų taisymo. Be to, šis būdas ilgam laikui pririš įmonę prie susikurto galutinio produkto, kadangi jį kuriant bus investuota daug išteklių. Privalumas tas, kad bus sukurta programa, pasižyminti itin geru išplečiamumu ir elastingumu.

Literatūroje išskiriami tokie porūšiai :

a) **Re-faktoringas** (angl. *refactoring*) - įprastai suprantamas kaip semantika, išsauganti programos transformaciją, kuri atliekama kontroliuojant programuotojui. Kitais žodžiais, *re-faktoringas* yra automatiškas: programuotojas nusprendžia, ar jis privalo būti atliktas ir tada įjungia *re-faktoringo* variklį, kuris automatiškai pakeičia programos kodą. DK kontekste perkeliama programą, *re-faktoringas* išsaugo jos semantiką tuo būdu, kad bendras funkcionalumas nepasikeičia, t.y. vykdant tam tikrą programos funkciją debesyje, tai nekeičia reikšmės (semantikos) žiūrint iš galutinio naudotojo pusės (Kwon, Tilevich 2014).

Re-faktoringas debesyje atliekamas dviem pagrindiniais būdais: ištraukiant paslaugą (angl. *extract service*) ir adaptuojant paslaugos sąsają (angl. *adapt service interface*). *Ištraukti paslaugą re-faktoringas* pateikia porciją centralizuotos aplikacijos funkcionalumo, kaip nuotolinę debesies paslaugą, perrašydamas kliento kodą ir patobulindamas jį reikiamomis atsparumo gedimams (angl. *fault-tolerance*) strategijomis. *Paslaugos adaptavimo sąsajos re-faktoringas* automatizuoja transformacijas, reikalingas perjungti paslaugos klientą į panašią alternatyvią debesimi pagrįstą paslaugą.

b) **Grupės programų modernizavimas** (angl. *multi-application modernization*). Keliose programose yra pakeičiama architektūra ir visas programų paketas įdiegiamas debesyje. Kai kelių programų *re-architektūrinimas* vyksta izoliuotai, prisireikia spręsti nesuderinamumą tarp duomenų

arba tarp dubliuojamų funkcijų, padidėja išlaidos dėl atskirų programų priežiūros ir veikimo. Tad pagal šį metodą programos analizuojamos kartu, ir nustačius konsolidacijos, bendro naudojimo galimybes, siekiama viename sprendime sutalpinti kelis komponentus.

1.4.3. Programinės įrangos peržiūrėjimas

Peržiūra (angl. *revision*) reiškia esamo programos kodo pakeitimą ar papildymą, leidžiantį atitikti programinės įrangos modernizavimo reikalavimus. Po to programos kodas įdiegiamas debesyje panaudojant perkėlimą ar *re-factoringą*.

Pakeitimai gali pridėti naujo funkcionalumo arba pritaikyti programą debesiui. Šiuo atveju naudojamos šios paslaugos: VM kaip paslauga (IaaS) arba platformos ir jų valdymo įrankiai, leidžiantys vystytojams pasinaudoti tiekėjo debesies siūloma nauda (PaaS).

1.4.4. Programinės įrangos pakeitimas debesies paslauga

Pakeitimas (angl. *replacing*) – atsisakoma senosios aplikacijos ir pradedama naudoti tiekėjo programa kaip paslauga interneto sąsajos pagrindu. Programos funkcionalumas įgyvendinamas kaip dalis debesyje veikiančios paslaugos. Tai SaaS modelis, kartais vadinamas PĮ debesifikavimu (angl. *cloudification*). Paprastai dar reikia migruoti duomenis, jų importas/eksportas atliekamas naudojant API (angl. *application programming interface*) arba konfigūravimo/valdymo konsolę (Gartner 2011). Šis modelis gali būti taikomas kombinuotai su PaaS (Jamshidi *et al.* 2016). Individualios pakeistos PĮ sprendimo savybės yra pateikiamos naujai, o ne perdirbamos, nes kai kurie senosios programos komponentai nebeatitinka verslo reikalavimų. Prieš pakeitimą yra išanalizuojamos tos PĮ savybės, kurios bus keičiamos naujomis DK paslaugomis, įvertinus to naudą lyginant su kitu metodu – *re-architektūrinimu*.

1.4.5 Skirtingų aplinkų integravimas

Hibridinis debesis – dviejų ar daugiau skirtingų debesų aplinkų integravimas. Trečiosios šalies teikiamų debesies paslaugų ir įmonės nuosavo duomenų centro paslaugų integravimas vyksta dėl kelių priežasčių:

- Išplėtimas į debesį (angl. *cloud bursting*) – priklausomai nuo situacijos vykstantis nuosavos sistemos pajėgumų išplėtimas pasinaudojant debesies aplinka, pagrįstas iš anksto nustatyta politika, darbų apkrovos stebėsena, paslaugų kokybės reikalavimais ir pan.). Gali būti naudojamas kaip būdas įmonėms išbandyti debesies paslaugas, jei jos turi abejonių dėl debesies aplinkos saugumo, stabilumo ar veiksmingumo.

- Tai taip pat būdas išvengti investicijų perviršio neišnaudojamiems įmonėje pajėgumams, skirtiems suvaldyti paklausos šuolius. Tai sprendimas įmonėms, kurioms pasitaiko sistemų apkrovos šuolių, tačiau nenorinčioms turėti infrastruktūros perviršio.

- Įmonės kartais turi poreikį migruoti darbus iš vienos debesies aplinkos į kitą, atsižvelgdamos į resursų naudojimą ir veiksmingumą.

- Suteikia atsparumą gedimams. Pakeistos programos architektūra yra iš dalies įdiegiama keliose debesies aplinkose, įgalinant ją tęsti veikimą naudojantis atsarginėmis kopijomis, kai ima neveikti pirmoji platforma (pvz., dėl programos kodo ar konfigūracijos klaidos). Šiame metode naudojama atsparių sistemų architektūra, kuri nukreipia naudotojus į artimiausią duomenų centrą.

1.4.6. Debesies platformos programinės įrangos įdiegimui

Kalbant apie populiarius paslaugų teikėjų sprendimus aukščiau nurodytiems nuosavos PĮ įgyvendinimo debesų kompiuterijoje būdams, galima trumpai apžvelgti keletą iš jų.

Amazon EC2 yra viename spektro gale. VM kopija *Amazon* aplinkoje atrodo lyg fizinė įranga, o naudotojai gali kontroliuoti beveik visą programinės įrangos vienetą, nuo OS branduolio ir toliau. Šis žemas lygmuo natūraliai apsunkina *Amazon* galimybę pasiūlyti automatinį išplečiamumą ir perkeliamumą, nes programavimo kalbos semantika, susijusi su kopijų darymu ir kitais būsenos valdymo klausimais, yra itin priklausoma nuo programos (Armbrust *et al.* 2010)

Priešingame spektro gale yra nuo programos specifinio tipo priklausančios platformos kaip *Google AppEngine*, kurios nukreiptos išimtinai tradicinėms interneto programoms, įgyvendinančioms tokią aplikacijos struktūrą, kurioje aiškiai atskiriami neturintis palaikymo atminčiai (angl. *stateless*) skaičiavimo lygmuo ir turintis palaikymą atminčiai (angl. *stateful*) talpinimo lygmuo. *AppEngine* turi įspūdingą automatinį išplečiamumą ir aukšto prieinamumo (angl. *availability*) mechanizmus, o nuosava *Megastore* duomenų saugykla yra skirta *AppEngine* aplikacijoms, kas reiškia apribojimus migracijai.

Programos *Microsoft Azure* platformoje paprastai parašytos naudojant .NET bibliotekas ir sudėtos į *Common Language Runtime*, t.y. nuo programavimo kalbos nepriklausomą valdymo aplinką. Ši struktūra yra daug lankstesnė nei *AppEngine* aplinkoje, bet vis tiek apriboja naudotojo pasirinkimą dėl talpinimo modelio ir programos struktūros. Taigi *Azure* yra per vidurį tarp tokių programinės įrangos struktūrų kaip *AppEngine* ir fizinės įrangos virtualių mašinų kaip *EC2*.

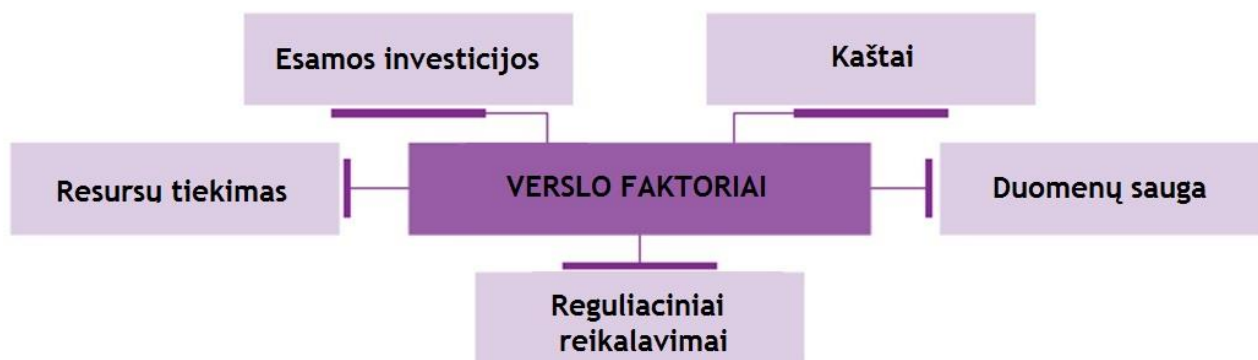
2. IT infrastruktūros perkėlimo į debesies aplinką iššūkiai

2.1. Iššūkių klasifikavimas

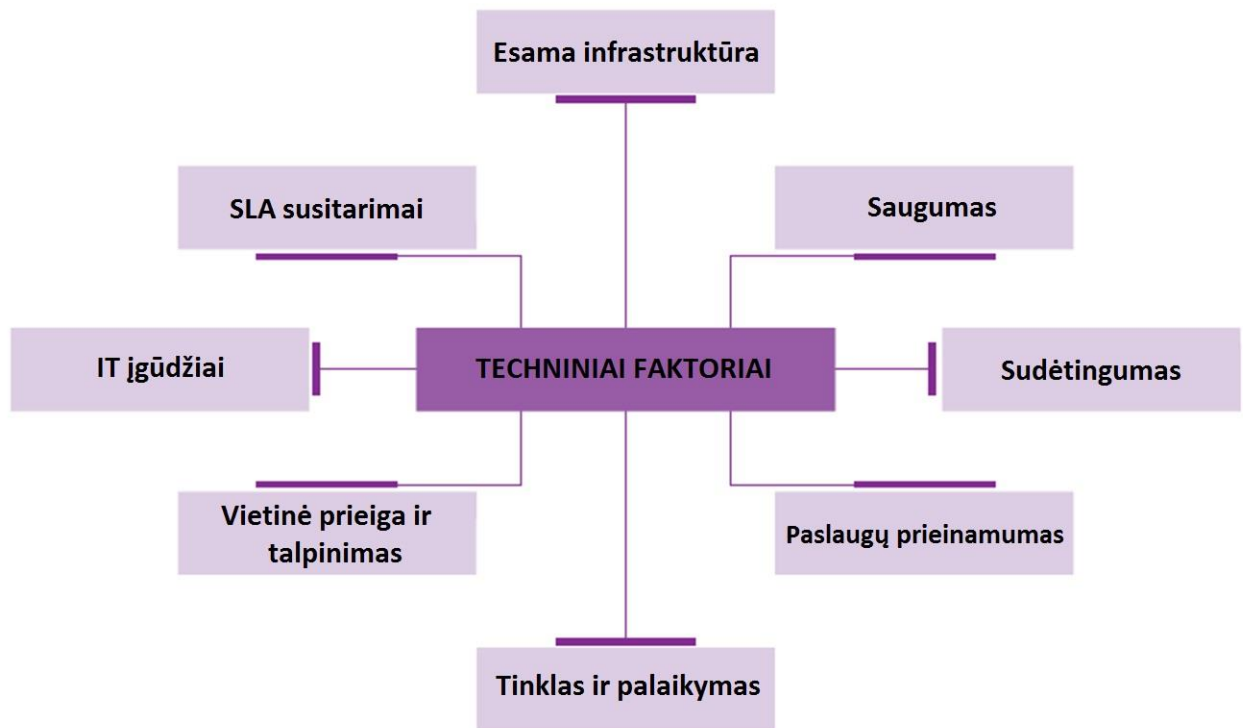
Kaip minėta šio darbo pradžioje, iššūkiais bus laikomi tie faktoriai, kurie daro įtaką migravimo į debesų kompiuteriją pasirinkimui, tiek ir pačiam migravimo procesui (Arbrust *et al.* 2010). Prieš pereinant prie konkrečių įmonės IT infrastruktūros migravimo į debesį iššūkių analizės, naudinga apžvelgti jų grupes. Įvairūs autoriai literatūroje naudoja skirtingas iššūkių klasifikacijas. Pavyzdžiui, kaip iššūkiai gali būti išskiriami funkciniai ir nefunkciniai reikalavimai, kuriuos nauja informacinė sistema turi atitikti (Cripps 2015):

- Funkciniai – pagrindiniai dalykai, kurių naudotojas tikisi iš programinės įrangos, pvz., programa įgalina sukurti paskyrą, ją pakeisti, ištrinti, palaikyti duomenų migravimą ir pan.
- Nefunkciniai – tai nėra tiesioginiai reikalavimai, jie labiau susiję su PĮ naudojimu, tam tikromis sistemos savybėmis ir esamais apribojimais, nulemiančiais PĮ naudojimą: pvz., prieinamumas 24/7, patikimumas ir pan. Nefunkciniai reikalavimai turi tiesioginę įtaką funkciniams reikalavimams ir kuriamos sistemos funkcionalumui.

Taip pat galimas skirstymas į veiklos iššūkius ir techninius iššūkius (žr. 2.1 ir 2.2 pav):



2.1 pav. Su įmonės vykdomos veiklos faktoriais susiję IT infrastruktūros migravimo į DK iššūkiai [29]



2.2 pav. Su techniniais faktoriais susiję IT infrastruktūros migravimo į DK iššūkiai [29]

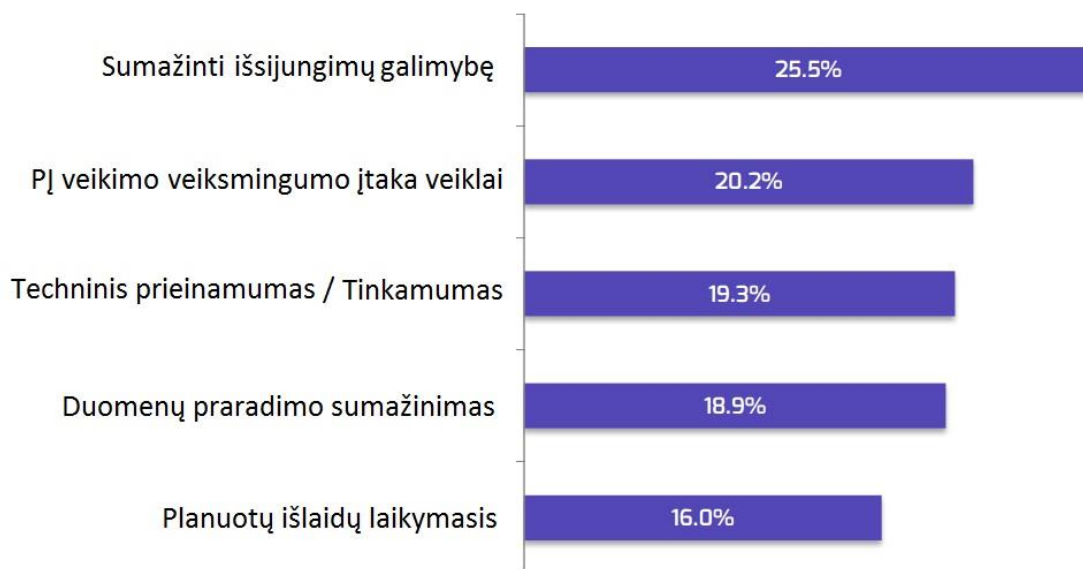
Iššūkiai pagal atsakomybės pasiskirstymą:

- Priklausantys ir kontroliuojami paslaugų naudotojo.
- Kontroliuojami išimtinai paslaugų teikėjo ar nekontroliuojami iššūkiai.

Nagrinėjamų iššūkių grupavimas galimas ir pagal migravimo į debesų kompiuteriją stadijas (Schneider, Sunyaev 2015):

- Reikalavimų nustatymas.
- Įsigijimas.
- Vystymas.
- Integravimas.
- Sutarties įvykdymas.
- Užbaigimas.

Iššūkiai gali būti klasifikuojami pagal svarbą remiantis atliktomis apklausomis. Naujausiuose tyrimuose nustatyti šie pagrindiniai migravimo į debesį faktoriai (žr. 2.3 pav):

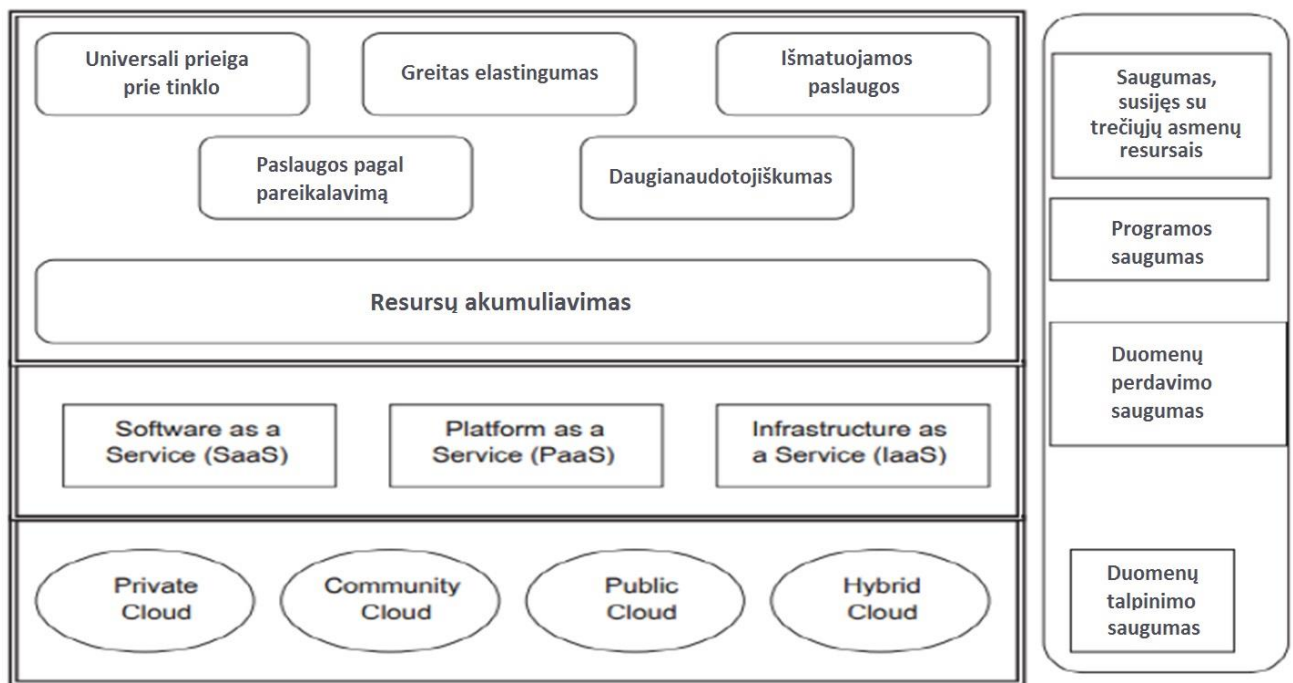


2.3 pav. Svarbiausi migravimo į debesies aplinką faktoriai. Apibendrinti 2016 metų skirtingo dydžio organizacijų (tiek pagal darbuotojų skaičių, tiek pagal pajamas) apklausos duomenys [11]

2.2. Saugumas ir privatumas debesies aplinkoje

Saugumą debesų kompiuterijoje galima nagrinėti įvairiais aspektais. Jį galima laikyti tiek DK pranašumu, tiek tam tikrai atvejais ir trūkumu. Taip pat galima jį vertinti pagal tai, kas ir už ką atsako užtikrinant saugumą: paslaugų teikėjas ar paslaugų naudotojas. Dar kitas pjūvis būtų saugumą nagrinėti pagal saugomą objektą – duomenys ar programinė įranga. Saugumo užtikrinimo DK iššūkis paprastai priskiriamas prie techninių, nors dėl jo svarbumo (ypač kas susiję su duomenų sauga ir privatumu) gali būti laikomas ir vienu veiklos iššūkių.

Saugumo iššūkių suvaldymas priklauso nuo to, koks DK paslaugų modelis ir diegimo būdas buvo pasirinktas bei kaip sprendžiami DK aplinkos klausimai, tokie kaip prieiga prie tinklo, duomenų migravimas į/iš debesies, elastingumas ar *daugianaudotojiškumas* (žr. 2.4 pav).



2.4 pav. Skirtingi saugumo užtikrinimo aspektai debesies aplinkoje [20]

DK yra nebūtinai saugesnė ar mažiau saugi aplinka, palyginus ją su organizacijos vidumi. Saugumo požiūriu, kaip bet kokia nauja technologija, DK suteikia naujų galimybių, pavyzdžiui, PĮ perkėlimas į debesį suteikia galimybę perdaryti senesnes programas ir infrastruktūrą tam, kad jos atitiktų dabartinius saugumo reikalavimus. Kitais atvejais, jautrių duomenų ir programų perkėlimas į naują infrastruktūrą gali viršyti reikalaujamą tolerancijos ribą.

2.2.1. Saugumo privalumai, kai programos yra perkeliamos į debesį

- Masto ekonomija: visos saugumo priemonės yra pigesnės, jei įgyvendinamos didesne apimtimi, dėl didesnio IT resursų sukoncentravimo.
- Duomenų centralizavimas. Nors tokiu atveju didelis duomenų kiekis vilioja įsilaužėlius, tačiau DK aplinkose naudojami apsaugos mechanizmai būna labiau patikimi, išplečiami ir efektyvūs kainos požiūriu. Tai apima įvairias apsaugos priemones, tokias kaip filtravimas, plėtinių (angl. *patches*) valdymas, virtualių mašinų kopijų ir *hypervizorių* saugumo stiprinimas ir pan. DK tiekėjų personalas turi galimybę specializuotis saugumo ir duomenų privatumo srityse.
- Saugumas kaip rinkos diferenciatorius: saugumas yra prioritetas daugeliui DK naudotojų; jie priima sprendimą pirkti paslaugas remdamiesi tiekėjo reputacija dėl konfidencialumo išsaugojimo, tiekėjo patikimumu ir platformos atsparumu gedimams.

- Standartizuotos sąsajos valdomoms saugumo paslaugoms: didieji DK teikėjai gali pasiūlyti atvirą, standartizuotą sąsają su valdomomis saugumo paslaugomis. Tai sukuria atviresnę ir iškart prieinamą saugumo paslaugų rinką.
- Greitas ir protingas resursų *išplečiamumas*: debesų teikėjas gali dinamiškai perskirstyti resursus, skirtus filtravimui, srauto formavimui, autentifikacijai, šifravimui, apsauginėms priemonėms (pvz., prieš DDoS atakas), kas yra akivaizdus privalumas atsparumo gedimams požiūriu.
- Savalaikis ir efektyvus atnaujinimų ir įsipareigojimų (angl. *defaults*) turėjimas.
- Debesyje kompiuterijos platformos paprastai yra labiau vienalytės nei daugumoje tradicinių kompiuterijos centrų. Didesnis vienalytiškumas palengvina platformos stiprinimą ir įgalina geresnę automatizaciją vykdant saugos valdymo veiklas, tokias kaip konfigūravimą, testavimą dėl pažeidžiamumo, saugumo auditus, diegiant platformose saugumo plėtinius; atsarginių kopijų kūrimą ir atstatymą.

2.2.2. Saugumo trūkumai migravus programinę įrangą į debesį

„The Top Threats“ darbo grupė atliko tyrimą ir parengė 2016 ataskaitą, kurioje buvo identifiкуotos *12 esminių probleminių dalykų debesies saugumui* (pagal tyrime nurodytą svarbą) (Top Threats Working Group 2016):

1. Duomenų pažeidimai, pavyzdžiui, dėl to, kad duomenys perduodami laikyti į valstybes, kuriose nesuteikiama pakankama asmens duomenų apsauga.
2. Silpnas tapatybės, ją patvirtinančių dokumentų ir prieigos valdymas.
3. Nesaugios API (angl. *application programming interface*).
4. Sistemos ir PĮ pažeidžiamumas.
5. Paskyros užvaldymas.
6. Piktavaliai naudotojai organizacijos viduje.
7. Sudėtingos nuolatinės grėsmės.
8. Duomenų praradimas.
9. Nepakankamai išsamus tiekėjo ištyrimas (angl. *due diligence*), pavyzdžiui, tiekėjo primestos naudotojui sąlygos, leidžiančios tiekėjui apdoroti asmens duomenis neatsižvelgiant į naudotojo nurodymus.
10. Piktnaudžiavimas ar aplaidus DK paslaugų naudojimas.
11. Paslaugos teikimo nutrūkimas (angl. *denial of service*).
12. Bendrai naudojamos technologijos klausimai.

Daugelį šių saugumo problemų galima suskirstyti pagal bendrus aspektus į šias grupes (Andrikopoulos *et al.* 2013):

- Duomenų kontrolės praradimas.
- Patikimumo mechanizmų trūkumas.
- Daug naudotojų (angl. *multi-tenancy*).

DK aplinkoje visų pirma yra prarandama fizinė **duomenų apsaugos** kontrolė dėl tos priežasties, kad yra dalinamasi duomenų talpinimo ir skaičiavimo resursais su kitais subjektais (viešajame debesyje), be to, fizinę ir kitokią duomenų kontrolę įgyja tiekėjas.

Debesyje, be talpinimo, **virtualizacija ir tinklai** yra kiti didžiausi saugumo iššūkiai. Virtualizacija leidžia keletui ar keliasdešimt naudotojų dalintis vieną fizinį serverį. Egzistuoja skirtingi virtualizavimo technologijų tipai ir kiekvienas jų gali skirtingu būdu taikyti saugumo mechanizmus. Virtualūs tinklai yra geras taikinyss atakoms, ypač kai jie komunikuoja su nutolusiomis virtualiomis mašinomis.

Kai kurie saugumo klausimai susiję su **daugianaudotojiškumu** yra šie:

- Nepakankama loginė saugumo kontrolė. Čia saugumas yra labiau priklausantis nuo loginio atskyrimo (keliuose lygmenyse) nei nuo fizinio resursų atskyrimo užtikrinant, kad naudotojas tyčia ar netyčia nepažeistų kitų naudotojų saugumo.
- Nežinantys apie pavojus arba piktavaliai naudotojai: dėl šių naudotojų gali atsirasti grėsmė saugumui, jeigu tiekėjas turi silpnesnę loginę kontrolę tarp naudotojų.
- Resursai, kuriuos naudotojai dalijasi, gali tapti vieninteliu gedimo tašku dėl naudotojo piktnaudžiavimo ar netinkamo naudojimosi.
- Nekontrliuojamas pakeitimų valdymas ir blogas konfigūravimas: debesies infrastruktūroje visi pakeitimai turi būti gerai koordinuojami ir testuojami.
- Sumaišyti naudotojų duomenys: tiekėjai linkę talpinti toje pačioje duomenų bazėje ir atsarginėse juostose duomenis iš kelių naudotojų, taip mažindami savo sąnaudas. Dėl to duomenys gali tapti pažeidžiami ar sugadinami, ypatingai jei talpinami bendrai naudojamose talpyklose.
- Rizika efektyvumui. Dėl daugelio naudotojų, vieno iš jų intensyvus paslaugos naudojimas gali paveikti kitų naudotojų gaunamos paslaugos kokybę.

2.2.3. Saugumo priemonių taikymas debesų kompiuterijoje

- Naudotojų ir prieigos valdymas – organizacijose paprastai būna nemažai darbuotojų arba jų komandų debesyje paleidžiančių savo programų grupes. Svarbu, kad būtų galima apibrėžti naudotojų prieigos taisykles, reguliuojančias prieigos suteikimą (Internap 2016).
- Ugniasienės ir kita tinklo prieigos įranga – daugumoje debesies aplinkų naudojamos fizinės ir programinės įrangos ugniasienės ir VPN įranga. Taip siekiama apsaugoti prieigą prie serverio resursų arba apibrėžti tinklo prieigos politikas tarp visumos programos komponentų.
- Resursų atskyrimas (tinklo, skaičiavimo, talpinimo) – dažnai programos, kurioms taikomi specialūs atitikimo (angl. *compliance*) reikalavimai, privalo naudoti tik joms vienoms priskirtąją fizinę įrangą arba vienam naudotojui skirtą aplinką.
- Šifravimo valdymo įrankiai - dauguma įmonių reikalauja, kad *daugianaudotojiškose* aplinkose būtų užtikrintas reikiamas šifravimo lygis – tiek duomenims jų perkėlimo į debesį metu, tiek talpinamiems duomenims. Įgalinantys įrankiai yra, pavyzdžiui, raktų ir sertifikatų valdymas.
- Sertifikatų laikymasis ir tiekėjo/naudotojo atsakomybės – įmonės su specifiniais sertifikatų poreikiais gali pareikalausti sertifikatų arba ataskaitų apie jų laikymąsi, siekdamos nustatyti, kokius konkrečius reikalavimus turi atitikti tiekėjas, o kurie lieka naudotojo atsakomybe. Naudotojo atsakomybė priklausys ir nuo DK paslaugų modelio: IaaS, PaaS ar SaaS.
- Atvirasis kodas – atvirojo kodo debesys gali turėti naudos iš griežtų standartų ir aktyvios jų bendruomenės pagalbos, kas reiškia, kad saugumo pažeidžiamumo papildiniai bus paprastai įdiegiami greičiau.
- Sistemos monitoringas ir įrašai (angl. *logs*) – jiems poreikis priklauso nuo migruojamų į debesį programų svarbumo. Monitoringo ir *log*ų įrašai gali turėti jautrios informacijos apie IT infrastruktūrą, tad šie duomenys paprastai skirti tik pačių tiekėjų vidiniam naudojimui, kurie jų nelinkę atskleisti klientams arba tretiesiems asmenims.
- Saugantis nuo rizikų aktualus informacinės saugos standartų laikymasis, populiariausias jų – ISO/IEC 27001 ir ISO/IEC 27002 (ETSI 2016).

2.2.4. Duomenų apsauga debesų kompiuterijoje

Duomenų perkėlimas. Perkeliant duomenis gali būti naudojamas jų filtravimas, pseudonimų suteikimas ar anonimizavimas. Kaip vienas iš saugumo užtikinimo būdų naudojamas šifravimas. Kai kurie autoriai, pavyzdžiui, kaip Carlo Curino, siūlo svogūno lukšto metodą, kai prieš keliant duomenis į debesį taikomos viena po kitos skirtingos šifravimo technikos. Siekiant duomenis debesyje apdoroti, užšifruoti sluoksniai yra dešifruojami vienas po kito, kol bus pasiektas duomenų užšifravimo sluoksnis su kuriuo jau galima būtų atlikti operacijas.

Su užšifruotais duomenimis apdorojimo operacijos paprastai negalimos, nors kai kurie autoriai teigia, kad homomorfinis užšifravimas tam netrukdo (Andrikopoulos *et al.* 2013).

Duomenų apdorojimas. Duomenis apdorojant pačiame debesyje, tampa aktuali atakų rizika iš debesies aplinkos vidaus. Duomenų ištrynimo klausimas: siekiama, kad visos duomenų kopijos (angl. *replicas*), atsarginės kopijos ir archyvai būtų ištrinami nustojus teikti paslaugą (pvz., tai gali būti itin svarbu, kai debesies tiekėjas yra nuperkamas jo kliento konkurento). Tai svarbu ir išsiplėtimo į debesį atveju (angl. *cloud bursting*), kai duomenys tik laikinai perkeliami į debesį, taip siekiant susidoroti su piko apkrovomis.

Duomenų privatumas ir vientisumas debesyje. Naudotojas talpina savo duomenis tiekėjo infrastruktūroje, tad svarbus saugumo iššūkis - privatumas ir naudotojo duomenų konfidencialumas (Rong *et al.* 2013). Naudotojai siekia garantijų, kad svarbios informacijos niekas nepasiektų ir nenaudotų neteisėtai, net ir pats tiekėjas.

- Resursų talpinimo vieta: naudotojai nežino, kurioje fizinėje vietoje yra laikomi jų naudojami DK resursai; ji gali būti ir kitose teisinėse jurisdikcijose. Tai sukelia potencialią problemą iškilus ginčams ir kartais būna už paties tiekėjo kontrolės ribų.

Duomenims, talpinamiems pas tiekėją, taikomi ne tik tiekėjo reikalavimai, bet ir tiekėjo valstybės įstatymai. Naudodami paslaugas naudotojai sutinka su Paslaugos sąlygomis, kurios paprastai visgi suteikia tiekėjams teisę tam tikrais atvejais atskleisti kliento informaciją, vadovaujantis taikomais įstatymais ir teisėsaugos institucijų prašymais.

Europos Sąjungos 2016 m. balandžio 27 d. priimtas reglamentas Nr. 2016/679 „Dėl fizinių asmenų apsaugos tvarkant asmens duomenis ir dėl laisvo tokių duomenų judėjimo ir kuriuo panaikinama Direktyva 95/46/EB“ nustato taisykles dėl asmens duomenų privatumo apsaugos. Šis reglamentas, kaip ir prieš tai galiojusi Direktyva 95/46/EB, uždraudžia asmens duomenų perdavimą į valstybes, kurios neužtikrina tinkamo apsaugos lygio. Asmens duomenų perdavimas už ES valstybių

ribų yra teisėtas, kai: a) tai atliekama su duomenų savininko sutikimu arba duomenys perduodami į valstybę, atitinkančią „saugaus uosto“ (angl. *safe harbour*) principus, b) tai atliekama vadovaujantis ES ir trečiosios valstybės susitarimais, arba, c) kitais iš anksto nustatytais atvejais. Vis dėlto, šio reglamento įgyvendinimas už ES ribų išlieka atviru klausimu.

- Organizacijos veiklos sritis. Priklausomai nuo veiklos srities, kai kurios programos ar sistemos turi atitikti griežtus specialiuosius reikalavimus. Tokios sritys kaip finansai ir sveikatos apsauga yra ypatingai apriboti griežtų reikalavimų, reguliuojančių duomenų ir asmenį identifikuojančios informacijos tvarkymą.

- Autentifikacija ir pasitikėjimas parsiųsta informacija: svarbiems duomenims esant tiekėjo infrastruktūroje, šie gali būti pakeisti be jų savininko sutikimo. Pasikeitę duomenys vėliau gali būti naudotojo parsisiunčiami ir apdorojami darant svarbius sprendimus. Duomenų autentiškumas šiuo atveju itin svarbus, tačiau vieningo standarto duomenų vientisumui užtikrinti nėra.

- Galėjimas taikyti DK standartus: standartais, kuriama įvairių organizacijų, siekiama užtikrinti duomenų perkeliamumą tarp skirtingų debesų ir jų stabilumą bei saugumą. Pavyzdžiui, dabar klientui siūlomos talpinimo paslaugos gali būti nesuderinamos su kito tiekėjo siūlomomis. Siekdami išlaikyti savo naudotojus, tiekėjai gali pradėti siūlyti vadinamas „prilimpančias paslaugas“, kurios sukels sunkumų naudotojams, norintiems migruoti iš vieno tiekėjo pas kitą.

2.2.5. Debesies naudotojo ir tiekėjo atsakomybės pasiskirstymas užtikrinant saugumą

Debesies naudotojas yra atsakingas už programos lygmens saugumą (Varia 2010). Debesis neatleidžia nuo pareigos užtikrinti nuosavos PĮ saugumo. Pavyzdžiui, *Amazon* nurodo, kad kiekvienoje migravimo proceso stadijoje naudotojas turėtų įgyvendinti tinkamą saugos geriausią praktiką:

1. Saugoti savo *Amazon* debesies (AWS) prisijungimo duomenis (keisti periodiškai arba įtariant pažeidimą; palaikyti „multi-factor“ autentifikaciją).

2. Apriboti naudotojų prieigą prie AWS resursų (sukurti skirtingus naudotojų vaidmenis ir jų grupes su skirtingomis prieigos teisėmis prie skirtingų resursų; nuolat peržiūrėti ir prižiūrėti naudotojo politikas; palaikyti saugos grupių įgaliojimus *Amazon EC2*).

3. Apsaugoti duomenis juos šifruojant (stacionariai (AES) ir perkeliant (SSL); automatizuoti saugumo politikas).

4. Įdiegti atstatymo strategiją (sukurti periodines virtualių mašinų atvaizdų kopijas (angl. *snapshots*) ir atsargines kopijas; kartais testuoti atsargines kopijas iki atsirandant jų poreikiui).

Debesies teikėjas yra atsakingas už fizinį IT resursų saugumą ir, tikėtina, už išorinės ugniasienės politikos įgyvendinimą.

Tarpinių lygmenų programinės įrangos vienetė saugumas yra pasidalijamas tarp naudotojo ir tiekėjo; kuo žemesnis abstrakcijos lygmuo ties naudotoju, tuo daugiau atsakomybės jam tenka. Pvz., *Amazon EC2* naudotojai turi daugiau techninės atsakomybės dėl saugumo (t.y. turi jį įgyvendinti ar pasirūpinti patys mainais už daugiau funkcionalumo) nei *Azure* naudotojai, kurie savo ruožtu turi mažiau atsakomybės nei *AppEngine* naudotojai.

2.2.6. DDoS atakos debesyje

Įmonės, kurios savo IT infrastruktūrą yra perkėlusios į debesį, taip pat susiduria su DDoS (paskirstyto paslaugos teikimo nutūkimo) (angl. *distributed denial of service*) atakomis, tačiau netiesiogiai, o per DK teikėjo sistemas.

Grėsmės naudotojui: aukos serverio ekonominiai praradimai, paslaugos nutūkimas, problemos susijusioms paslaugoms, žala veiklai ir reputacijai. Didelis ryšio srauto užkimšimas, didesnis energijos suvartojimas. Kartu talpinamų VM veikimo pažeidimas, resursų lenktynės ir papildomos migracijos, atsiradusios dėl fizinio serverio resursų išsekvojimo. Papildomų migracijų atveju gali būti neįmanoma įgyvendinti kartu talpinamų VM reikalavimų dėl to, kad auka VM išnaudojo resursus.

Grėsmės debesies tiekėjui ir kitiems naudotojams: svarbiausia grėsmė yra DDoS įtaka debesies infrastruktūrai, o taip pat ir kitoms kartu egzistuojančioms paslaugoms. Kadangi debesies infrastruktūra yra daugianaudotojiška, daug VM egzistuoja kartu. Atsižvelgiant į dėl DDoS atakos paveiktos VM atsiradusį nenutrūkstamą resursų poreikį, kitos toje pačioje fizinėje mašinoje esančios VM patirs veikimo sutrikimus ir nesuderinamumą tarp skirtingų resursų panaudojimo, kadangi debesies resursų paskirstymo mechanizmas priims sprendimą pagal tai, kokie yra faktiniai poreikiai tuo momentu (Somani *et al.* 2015).

2.3. Organizaciniai iššūkiai

Įmonės, pereidamos prie IaaS modelio, privalo suprasti pasikeitimų apimtį ir pastangas, reikalingas tiems pasikeitimams pasiekti, kad įvertinti šio modelio privalumus, trūkumus ir pasekmes. DK įdiegimo sėkmė yra tiek pat priklausoma nuo organizacinių ir kultūrinių procesų, kiek ir nuo pačios technologijos.

Įmonės susidurs su šiais organizaciniais pokyčiais (Khajeh-Hosseini *et al.* 2012):

- Apskaitos pasikeitimas, nes virtuali IT infrastruktūra nėra įgyjama iš anksto, tad ją bus naudojama kaip paslauga ir apmokama kaip už resursą.
- Projektų valdymo pasikeitimas, nes dėl perėjimo prie debesų kompiuterijos IT padalinys neteks savo įtakos (tačiau IaaS atveju vis tiek jos išlaikys daugiau nei, pavyzdžiui, SaaS atveju).
- Pasikeis sistemų palaikymas, nes IT administratoriai jau nebeturės visos infrastruktūros kontrolės. Jų darbas vis labiau apims bendravimą su debesies tiekėju tikintis, kad šis laiku imsis atsiradusių problemų sprendimo.
- Galutiniai naudotojai galės dirbti labiau bendradarbiaudami, bet migruotai PĮ neveikiant dėl gedimo, nebegalės tiekėjui nurodyti išspręsti problemos pirmenybės tvarka.
- Pasikeis IT padalinio paskirtis ir funkcijos, o tai gali įtakoti ir įmonės struktūros pasikeitimus. Be to, organizacijos turėtų nustatyti naujas planavimo taisykles ir IT paslaugų tiekimo visos organizacijos mastu valdymą.

Organizacinių pasikeitimų, kai migruojama į DK, kliūtys:

- Sudėtingas techninių sistemų ir organizacijos suderinamumas. Tiek senoji infrastruktūra, tiek IT pagrindu vykstanti organizacijos transformacija yra tarpusavyje susiję. Ypač didelėse organizacijose yra skaitlingas kompiuterines sistemas apjungiančios infrastruktūros, kurios buvo įdiegtos per ilgą laikotarpį, tad apima skirtingas technologijas, gali turėti skirtingus „savininkus“ pačios įmonės viduje. Be to, ir verslo procesai buvo parengti atsižvelgiant į specifines tų programų savybes. Tarptautinėms įmonėms, veikiančioms skirtingose jurisdikcijose, skirtos programos veikia ir yra palaikomos skirtingais būdais.
- Darbuotojų priešinimasis pasikeitimams. Darbuotojai iš inertiškumo priešinasi pakeitimams, kurių iniciatyva kyla kituose organizacijos padaliniuose. Tad tam, kad DK suteiktų tikrą vertę, ji turi būti priderinta prie įmonės, o ne tapti platforma, skirta elementarių užduočių paleidimui. Tad visi procesu suinteresuoti asmenys turi prisidėti: techniniai, projektų valdymo, finansų valdymo darbuotojai.
- Kai yra galimybė - išlaikoma senoji organizacijos forma, derinant įvairiapuses kompiuterijos aplinkas, kurios paremtos nuosavais organizacijos serveriais, privačiais įmonės debesimis ir, tikėtina, daugiau nei vienu viešojo debesies tiekėju.
- Mažas įmonės dydis. Didelės įmonės turi privalumą prieš mažąsias, nes turi daugiau resursų ir gali prisiimti didesnę riziką, susijusią su inovacijų diegimu. Mažos įmonės yra labiau

prisitaikančios, tačiau tai nereiškia, kad jos nedelsdamos diegia naujas technologijas. Taigi mažas dydis šiuo atveju negatyviai įtakos IaaS diegimą (Oliveira *et al.* 2014).

- Reguliaciniai apribojimai įmonės veiklai. Reguluodamos teisės aktais valstybės institucijos gali paskatinti IT inovacijų diegimą įmonėse, taip pat jį ir apriboti. Pavyzdžiui, teisės aktų leidėjai JAV ir ES įpareigoja apsaugoti organizacijų duomenis. Valstybės institucijai reikalaujant, kad įmonė atitiktų privalomus DK standartus ir protokolus, įmonės turės paskatų migruoti į DK, - tokiu būdu reguliacinė aplinka turės teigiamą poveikį.

Debesų kompiuterija ir IT *outsourcingas*. IT infrastruktūros migravimas į debesį yra IT resursų kaip paslaugos naudojimas, apmokant už sunaudojamą jų kiekį, t.y. iš esmės IT paslaugų *outsourcing'as*. Šiuo atveju galime palyginti įprastą IT *outsourcingą* su IT *outsourcingu* į debesį:

- Tradiciniame IT *outsourcinge* įmonė perka paslaugą iš kitos įmonės ir pasirašo sutartį su paslaugą tiekiančiu tiekėju. Įmonė gauna paslaugą, teikiamą iš nustatytos vietos, kurioje yra įmonei priskirti įrenginiai. Mokamas mėnesio ar metinis mokestis, skirtas šių IT resursų išlaikymui, tokiu būdu užtikrinant jų funkcionavimą.

- IT *outsourcingo* debesies atveju įmonės programos gali būti įdiegtos nežinomoje vietoje, o kaina mokama tik už sunaudotus resursus. Šiuo atveju modelis radikalesnis, nes naudojamas procesų automatizavimas, vietoje paprasto proceso perkėlimo į išorinę vietą.

Resursų valdymo pasikeitimas. Debesyje IaaS suteikia savarankišką resursų naudojimą, priklausomai nuo poreikio. Automatizavimas ir *orkestravimas* – debesyje suteikiama savarankiško naudojimosi sąsaja, skirta greitai apsirūpinti skaičiavimo ar talpinimo resursais. Rinkoje egzistuojantys IaaS platformų įrankiai automatizuoja infrastruktūros pritaikymą prie programos naudojimo pasikeitimo, pvz., procesoriaus ribos, srauto pralaidumas ar I/O operacijos. Pavyzdžiui, *DevOps* skirti įrankiai, kaip debesies API (aplikacijos programos sąsaja) integravimo ir konfigūravimo valdymas, gali padėti automatizuoti didelio skaičiaus serverių tiekimą ir greitai instaliuoti reikiamą programinę įrangą siekiant iš karto pasiekti didelį mastą – potencialiai tūkstančius serverių. Taip pat naudotojai gali apsirūpinti reikiamu kiekiu „infrastruktūros“ naudodami interneto pagrindu veikiančią grafinę naudotojo sąsają, veikiančią kaip IT operacijų valdymo skydas.

Teisinės sutartys, kartu su SLA (angl. *service level agreement*), techniniais patikrinimais ir monitoringu, yra tos priemonės, kurios sumažina resursų valdymo riziką. Debesies naudotojas turėtų stebėti, ar realybėje yra vykdomas sutartas paslaugų lygis ir kiti debesies tiekėjo įsipareigojimai.

2.4. Teisiniai ir reguliaciniai reikalavimai

Debesies kompiuterijos paslaugų naudotojai turėtų žinoti, kokia yra jų atsakomybė, susijusi su atitinkamų nacionalinių ir tarptautinių reguliacinių reikalavimų įgyvendinimu, ir užtikrinti, kad tie įsipareigojimai būtų tinkamai įgyvendinti. Kai kurie teisiniai/reguliaciniai klausimai elektroninės informacijos talpinimui (Cloud Standards Customer Council 2014):

- fizinė duomenų talpinimo vieta;
- duomenų pažeidimai;
- asmens duomenų privatumas;
- duomenų sunaikinimas, kai jų nebereikia ar jie perkeliami kitur;
- intelektinė nuosavybė;
- teisėsaugos institucijų prieiga;
- paslaugų prieinamumas;
- planai dėl prieigos suteikimo užtikrinimo (pvz., daugiau nei 150 valstybių yra ratifikavusios Jungtinių Tautų konvenciją dėl Neįgaliųjų asmenų teisių);
- specifiniai debesies paslaugų įdiegimo standartai ir reikalavimai skirtingose pramonės srityse.

Kyla klausimas, kas teisiškai atsakingas už debesies paslaugų naudojimo atitikimą teisiniams reikalavimams? Įprastai daugumoje reguliacinių reikalavimų ir sutarčių yra nustatyta, kad debesies naudotojas yra galiausiai atsakingas už debesies paslaugų naudojimą bei savo duomenų saugumą, vientisumą ir talpinimą, net jeigu šie duomenys yra patalpinti pas išorinį DK tiekėją (Erl *et al.* 2013).

2.5. Finansiniai iššūkiai

Kalbant apie migravimo į debesį finansinių sąnaudų analizę reikia atskirti dvi sąvokas:

- Bendrosios įsigijimo išlaidos, arba TCA (angl. *total cost of acquisition*).
- Bendrosios nuosavybės turėjimo išlaidos, arba TCO (angl. *total cost of ownership*); ši sąvoka apima visus įmanomus, įskaitant ir netiesioginius, investicijų darymo kaštus: įsigijimo kaina, nuvertėjimas, migravimas, patalpų ploto, energijos tiekimo, aušinimo, valdymo, įrangos aptarnavimo ir kiti kaštai (IMB Corporation 2010).

TCO požiūriu finansinės naudos dėl migravimo į debesį įvertinimas yra sudėtingesnis nei TCA, tačiau rengiant migravimo strategiją būtent TCO padės daug geriau atvaizduoti laukiamus pokyčius, nei vien tik paprastas palyginimas tarp DK resursų nuomos ir nuosavų resursų įsigijimo kainos. Pavyzdžiui, finansinės išlaidos susijusios su dinaminiu resursų išplečiamumu debesyje, laiko

sąnaudomis, resursų švaistymu (nepanaudojimu) arba, atvirkščiai, resursų nepakankamumu, taip pat turėtų būti vertinamos (Erl *et al.* 2013).

Kaina nebėra pagrindine paskata perėjimui į debesį (Internap 2016). Anksčiau buvo laikoma, kad DK - kainos požiūriu efektyvesnis sprendimas nei fizinės įrangos sprendimai, ir nors tai paprastai tiesa SaaS aplikacijoms, IaaS modelyje pagrindiniai migravimo privalumai yra ne tiek kaina, kiek išplečiamumas ir naudojimosi paprastumas.

Nuolat patiriamos veiklos išlaidos dėl fizinės IT infrastruktūros turėjimo (Erl *et al.* 2013):

- Techniniai darbuotojai, palaikantys sistemų veikimą.
- PĮ patobulinimai ir atnaujinimai, reikalaujantys papildomo testavimo ir diegimo.
- Einamosios išlaidos ir investicijos kapitalui dėl energijos tiekimo ir aušinimo.
- Saugumo ir prieigos prie resursų kontrolės priemonės.
- Buhalterijos ir administracijos darbuotojai, prižiūrintys licencijas ir kitą dokumentaciją.

Išlaidų skaičiavimas. Veiklos sąnaudų viešajame debesyje įvertinimas yra sudėtingas nes debesies resursų apmokestinimo modelis reiškia perėjimą nuo išlaidų kapitalui (CAPEX) į išlaidas veiklai (OPEX), kas nulemia jų apskaičiavimo specifiką. OPEX išlaidas infrastruktūrai padaro labiau nuspėjamas ir suteikia įmonei didesnę jų kontrolę (Khajeh-Hosseini *et al.* 2012).

Išliekantis neaiškumas apskaičiuojant debesies išlaidas susijęs su: a) realiai sistemos suvartotais resursais, kurie priklauso nuo apkrovos; b) pasirinktu PĮ įgyvendinimo debesies aplinkoje būdu, galinčiu įtakoti jos kainą, nes tokie resursai kaip pralaidumas yra labiau apmokestinami tarp skirtingų debesų, nei viduje debesies, ir c) debesies tiekėjo taikoma įkainių tvarka, kuri gali bet kada pasikeisti.

Kai kurių tyrimų rezultatai visgi patvirtina, kad priimant sprendimą dėl migravimo, nederėtų remtis tik finansiniais duomenimis. Pavyzdžiui, literatūroje aprašytas dujų ir naftos sektoriaus įmonės IT sistemos migravimo į DK tyrimas parodė, kad sistemos infrastruktūra *Amazon EC2* debesyje turėtų kainuoti apie 37% mažiau penkerių metų laikotarpiui lyginant su nuosavu duomenų centru. Kitas tyrimas parodė priešingai, - kad nėra didelio kainų skirtumo tarp fizinių serverių įsigijimo išlaidų ir išlaidų perkeltiant vidutinio dydžio mokyklos IT sistemas į debesį. Abejų tirtų atvejų skirtumą galima paaiškinti tuo, kad mokyklos atveju visiškai nauja tinklo infrastruktūra turėjo būti įsigyta. Bet gali būti ir taip, kad DK yra pigesnis variantas įmonei, kuriai reikia virš 30 serverių, dėl papildomai patiriamų išlaidų specialioms spintoms, aušinimui, žmogiškiesiems ištekliams ir tinklo infrastruktūrai tarp minėtų fizinių serverių įrengti (Khajeh-Hosseini *et al.* 2012).

Literatūroje ir praktikoje minimi šie išlaidų patiriamų dėl PĮ migravimo į debesį aspektai (Andrikopoulos *et al.* 2013):

- Debesies tiekėjo imamas mokestis: įkainis už resursą per laiko vienetą (pvz., 1 *micro image* - 0.25 \$/para). Kokį apmokestinimą pasirinkti – už valandą ar už mėnesį – priklauso nuo įmonės poreikių. Jeigu reikia tik išsiplėtimo į debesį (angl. *cloud bursting*) ar testavimo tikslais – valandinis apmokestinimas atrodo ekonomiškesnis. Labiau nuspėjamiems poreikiams – mėnesio apmokestinimas galimai geresnis.
- Kainos modelis ar jų kombinacijos – pvz., *Microsoft Azure* taiko nuolaidą už *per-use* modelį tam tikrai resursų kvotai, kurią viršijus įprastas *per-use* įkainis taikomas. Kitas pavyzdys – AWS leidžia naudotojams pateikti aukcionui savo nepanaudotą *EC2* debesies išteklių kiekį.
- Programinės įrangos licencijavimas paprastai pagrįstas konkrečiu skaičiumi kompiuterių vienetų, kas nedera su debesies sprendimais, pagrįstais resursų dinamika. Pavyzdžiui, sistemos VM išsiplėtimas gali reikšti netyčinį licencijos sutarties pažeidimą. Praktikoje, kai kurie teikėjai siūlo skirtingus licencijavimo variantus, pvz., *Amazon Relation Database Service* naudotojai gali naudoti nuosavas licencijas *MySQL*, *Oracle* ar *IBM DB2*; mokėti už valandinę „per-hour“ licenciją naudoti *Oracle DB*, ar sumokėti vienkartinį mokestį už kiekvieną *Amazon RDS* kopiją, kad gauti mažesnius valandinius įkainius. Kiti tiekėjai kiekvienai paskyrai suteikia licencijas be užmokesčio.
- Išlaidų naudojantis vietine ir debesies IT infrastruktūra palyginimas. Kai kurie autoriai pažymi, kad iš esmės reikšmingai sutaupyti pereinant nuo CAPEX prie OPEX galima tik esant didesnėms variacijoms tarp piko, vidutinio ir mažo sistemos apkrovimo.
- Išlaidos, patiriamos dėl paties migravimo proceso, priklauso nuo pasirinkto PĮ adaptavimo debesyje būdo: programos kodo *re-architektūrinimas*; dalinis PĮ funkcionalumo migravimas; migravimas programinės įrangos kaip visumos. Pavyzdžiui, kai kurie autoriai nurodo, kad migravimas, jeigu migruojama visa programa, apsimoka tik mažo dydžio ir stabilioms įmonėms, kai tuo tarpu dalinis programos komponentų migravimas joms yra per daug brangus.

Atsižvelgiant į išdėstytą, siūlomi migravimo į debesį plano **finansinio įvertinimo** žingsniai (Chan 2017):

1. Atlikti dabartinių organizacijos IT infrastruktūros išlaidų auditą:
 - Tiesioginės išlaidos : a) programinės ir fizinės įrangos įsigijimo išlaidos, b) veiklos išlaidos (aptarnavimas ir priežiūra). Iš esmės atitinka TCA ir TCO.

- Netiesioginės išlaidos : potencialios išlaidos ir veiklos nuostoliai dėl IT infrastruktūros gedimų.

2. Apskaičiuoti numatomas nuolatinės išlaidas, jeigu infrastruktūra perkelta į debesį.
3. Įvertinti migravimo į debesį įgyvendinimo kaštus:
 - Duomenų migravimas (duomenų konversija, sinchronizavimas, šifravimas).
 - PĮ integravimas su debesies aplinka ir testavimas.
 - Išlaidos konsultantams.
4. Pomigracinės išlaidos.

Kai kurie debesies paslaugų teikėjai siūlo savo išlaidų skaičiuokles (žr. 2.5 pav), dalis jų pritaikytos apskaičiuoti ir palyginti programų paleidimo išlaidas tiekėjo debesyje ir išlaidas, patiriamas paleidžiant tas pačias programas nuosavoje infrastruktūroje. Pavyzdžiui, į *AWS Total Cost of Ownership* (TCO) skaičiuoklę suvedus nuosavos infrastruktūros konfigūraciją, galima gauti detalų nuosavos infrastruktūros išlaidų apskaičiavimą ir palyginimą su AWS.

The screenshot displays the Microsoft Azure Storage cost calculator. The main configuration area on the left includes dropdowns for Region (North Europe), Type (Block blob), Pricing Tier (Standard - General Purpose), and Data Redundancy (GRS). Under Capacity, a value of 1 TB is entered, leading to a calculated cost of \$49.15/MO. Under Storage transactions, a value of 1 is entered, leading to a calculated cost of \$0.01/MO. The Sub-total is \$49.16/MO. On the right, the 'Your estimate' sidebar provides a detailed breakdown of costs: SQL Database (\$15.03), Storage (\$49.16), App Service (\$55.80), Application Insights (\$0.00), and Support Options (\$0.00), resulting in an Estimated Monthly Cost of \$119.98. A green 'Purchase options' button and a blue 'Export estimate' link are also visible.

2.5 pav. Microsoft Azure resursų skaičiuoklė pateikia sąmatą, įvertinančią bendrą IT infrastruktūros debesyje naudojimo kainą priklausomai nuo tos infrastruktūros savybių ir parinkčių [33]

Tiekėjų virtualios mašinos dažnai siūlomos kaip rinkiniai: kartu su talpykla duomenims, apkrovos skirstytoju ir kitais resursais:

DS1 Standard		DS2 Standard		DS11 Standard	
1	Core	2	Cores	2	Cores
3.5	GB	7	GB	14	GB
	2 Data disks		4 Data disks		4 Data disks
	3200 Max IOPS		6400 Max IOPS		6400 Max IOPS
	7 GB Local SSD		14 GB Local SSD		28 GB Local SSD
	Load balancing		Load balancing		Load balancing
	Premium disk support		Premium disk support		Premium disk support
80,94 EUR/MONTH (ESTIMATED)		161,87 EUR/MONTH (ESTIMATED)		188,22 EUR/MONTH (ESTIMATED)	

2.6 pav. Microsoft Azure siūlomos Windows Server Datacenter virtualios mašinos, 2017 m. kovas[33]

2.6. Programinės įrangos suderinamumas

Diegiant IaaS modelį turi būti atsižvelgta, ar naujoji debesies aplinka tiks paleisti įmonės programinę įrangą ir, jeigu debesies aplinka nesuderinama, kiek sudėtinga PĮ adaptuoti nepamirštant suderinamumo su įmonės vykdomos veiklos procesais.

Sudėtingumas yra „laipsnis, pagal kurį inovacija yra vertinama kaip esanti gana sudėtinga suprasti ir naudotis“ (Oliveira *et al.* 2014). Lengvesnis technologijos integravimas į verslo operacijas reiškia didesnius šansus, kad ji bus įdiegta. Tačiau DK paremtų sprendimų diegimas gali tapti rimtu iššūkiu įmonėms, neturinčioms techninio išmanymo ir IT specialistų. Pavyzdžiui, esamų programų integravimas su specifine DK infrastruktūra gali pareikalauti įmonėje neturimų ekspertinių žinių. Nemaži iššūkiai taip pat laukia naudojant DK paremtus sprendimus nusistovėjusiems verslo procesams ir duomenų saugumui užtikrinti padalintoje, daugianaudotojiškoje aplinkoje. Tad sudėtingumas neigiamai įtakoja migravimo į DK įgyvendinimą.

Suderinamumas - tai „laipsnis, pagal kurį inovacija tinka potencialaus įgyvendintojo esamoms vertybėms, ankstesnei praktikai ir dabarties poreikiams“. Suderinamumas yra vienas iš faktorių, apsprendžiančių, ar migruoti įmonei savo IT (Oliveira *et al.* 2014).

Prieš apsisprendžiant dėl migravimo į IaaS įmonei svarbu įvertinti turimus ir galimus paskirti šiam procesui įmonės išteklius. Tokie programos perkėlimo aspektai kaip jos papildymas daugianaudotojiškumu, architektūros ir programos kodo pakeitimas, saugumo užtikrinimas, sistemos konfigūravimas, testavimas, darbuotojų mokymai sunaudoja daug finansinių, organizacinių resursų ir IT padalinio darbuotojų laiko, kurio matui žemiau aprašytuose pavyzdžiuose naudojamas terminas

žmogaus mėnesiai (angl. *man-months*) – vienetas, lygus vieno specialisto darbo kiekiui per mėnesį (Maenhaut *et al.* 2015):

1-asis atvejis. Pieter-Jean Maenhaut *et al.* aprašytame medicininės komunikavimo įrangos migravime į *Microsoft Azure* aplinką programos pakeitimų įgyvendinimas užtruko 6 žmogaus mėnesius. Iki galo parengtai migravimui programai paleisti dar papildomi 14 žmogaus mėnesių būtų reikalingi. Papildomos užduotys būtų skirtos adaptuoti debesies programą atsižvelgiant į turimus SLA, ištirti atsarginių kopijų ir atstatymo galimybes, apmokyti įdiegėjus ir atlikti visą testavimą.

2-asis atvejis. Migruojant į IaaS mažiau sudėtingą aplikaciją – Darbų planuotoją, buvo išbandyti du variantai: *Google AppEngine*, kaip PaaS tiekėjas, nes jis suteikia PHP paleidimo aplinką (angl. *runtime environment*); ir *Amazon EC2*, kaip IaaS tiekėjas. *Google AppEngine* platformoje buvo reikalinga atlikti daugiau pakeitimų su senąja programa, nes ši platforma turi daugiau apribojimų programoms, kai tuo tarpu *Amazon EC2* aplinkoje reikia daugiau priežiūros, nes čia suteikiama visa virtualios mašinos kontrolė. Pastarojo migravimo į IaaS atveju atlikti reikiamus programos migravimui pakeitimus buvo reikalingi tik 2 žmogaus mėnesiai. Dar kiek mažiau nei 2 žmogaus mėnesių pareikalavo daugianaudotojiškumo pridėjimas. PĮ migravimo veiksmų atlikimui *EC2* platformoje buvo pakankama tik 1 diena, kai *Google AppEngine* – keletas dienų. Iki galo parengtai migravimui programai paleisti (įskaitant ir testavimą) dar papildomai reikėtų apie 4 žmogaus mėnesių.

Programų tarpusavio suderinamumas gali būti skirstomas į (Cloud Standards Customer Council 2013):

- *Proceso (ar kontrolės) integracija*, kai siekdama įvykdyti užklausą programa aktyvuoja kitą programą.
- *Duomenų integravimas*, kuomet programos turi bendrų duomenų arba vienos programos išvesties duomenys tampa kitos programos įėjimo duomenimis.
- *Atvaizdavimo integracija*, kai kelių programų rezultatai atvaizduojami tuo pat metu valdymo pulto ar mikserio (angl. *mashup*) pagalba.

Suderinamumo problemų sprendimui rekomenduojami šie veiksmai:

- Nustatyti sujungimo vietas („integravimo taškus“) tarp atskirų programų.
- Išlaikyti lankstų požiūrį, naudoti kelis metodus priklausomai nuo situacijos.
- Remtis standartais, siekiant geresnės sistemos aptarnavimo kokybės ir atsparumo galimiems debesies tiekėjo daromiems pokyčiams.

- Įvertinti, kad ateityje gali būti daugiau migracijų, tad migravimas į debesį gali būti proga modernizuoti infrastruktūrą ir padaryti ją atsparesnę pokyčiams.

Atviras kodas. Nors paprastai atviras kodas yra tinkamesnis vystytojams, kuriantiems privatų debesį, renkantis tiekėją su atviro kodo DK paslaugomis gali būti pasiektas didesnis suderinamumas. Kita vertus, įmonės, turinčios atviro kodo sistemas, turėtų apsvarstyti suderinamumą su šiuolaikine programine įranga, kaip *VMware* ir *Hyper-V*, ir pasirinkti tiekėją turintį technologiją, kuri dera prie įmonės vizijos.

2.7. Programinės įrangos veikimo veiksmingumas

Debesies atveju, veikimo veiksmingumas (angl. *performance*) susijęs su tuo, kaip greitai debesies infrastruktūros aplinka gali apdoroti operaciją ar užklausą (Internap 2016). Čia verta atsižvelgti į šiuos parametrus:

- Apdorojimas (angl. *processing*) – atsižvelgiant į poreikius, reikėtų nusistatyti CPU ir RAM santykį, reikalingą nuosavos PĮ klasteriui ir įvertinti, ar tą gali suteikti debesies tiekėjas.

- Talpinimas (angl. *storage*) – debesyje siūlomas vietinis turinio (angl. *local volume*) talpinimas arba SAN pagrįstas talpinimas (angl. *storage area network*, t.y. didelio greičio talpinimo įrenginių tinklas). Tiekėjas gali būti nustatęs IOPS (įvesties/išvesties operacijos per sekundę, angl. *input/output operations per second*) reikalavimus disko prieigai ir duomenų bazės naudojimui. Kai kurie tiekėjai siūlo didelės talpos, žemo veiksmingumo SATA diskus arba aukšto veiksmingumo mažesnės talpos SSD diskus. Atsižvelgiant į poreikius turėtų būti įvertinta, ar IOPS ir pralaidumas iš SAN talpyklos yra pakankami, ar IOPS iš vietinės talpyklos tinka geriau. Aplikacijų dalims, reikalaujančioms aukšto veiksmingumo disko prieigos, pvz., duomenų bazėms, kai kurie tiekėjai gali siūlyti garantuotus IOPS, taip siekiant atitikti specifinius programos komponentų reikalavimus.

- Tinklas – PĮ veiksmingumą itin veikia delsa (angl. *latency*) ir tinklo greitis, tad optimizuotas interneto protokolas (IP) gali pagerinti atsakymo laiką ir užtikrinti patikimesnį PĮ veiksmingumą:

- a) Vieša ir privati tinklo prieiga. Vieša prieiga siūloma per internetą. Jei pageidaujama privačios prieigos, tiekėjas turi galėti suteikti privatų sujungimą arba VPN paslaugą.

- b) Geografinė vieta – įvertinti, kada naudotojui svarbu būti netoli debesies. Dažnai klaidingai manoma, kad debesies serverių vieta gali reikšmingai pakeisti greitį ir delką įmonės tinkle. Kitas atvejis - renkantis duomenų centrą dėl fizinės įrangos, t.y. bendrai jos nuomai iš tiekėjo (angl. *colocation*), vieta netoli biuro yra pagrįstas pasirinkimas, nes IT darbuotojai ir inžinieriai, tikėtina, turės ten nuvykti,

kad atliktų su infrastruktūros diegimu susijusias užduotis. Tačiau DK paslaugos yra paprastai numatytos valdyti per API ar portalų sąsajas, kas sumažina poreikį būti fiziškai šalia.

Iš kitos pusės, įmonei gali būti svarbi debesies fizinė vieta dėl geografinės rinkos ir naudotojų. Pavyzdžiui, įmonei, kurios prioritetinė rinka yra JAV šiaurės rytų pakrantė, gali būti paranku įgyvendinti migravimą į toje vietoje esančią debesies aplinką, kad sumažinti delsą ir pagerinti sąlygas naudotojams.

Veikimo veiksmingumo padidinimas.

Programoms ir darbams, kurie yra jautrūs veikimo veiksmingumui arba apdorojantys daug duomenų, gali reikėti nuosekliai aukštesnio veiksmingumo lygio nei IaaS virtualus debesies gali suteikti už priimtina kainą. Tad į įmonės aplinką gali būti paranku įterpti *baremetal* (virtuali mašina diegiama tiesiai ant serverio fizinės įrangos, o ne talpinama serverio OS) debesies alternatyvas. *Baremetal* debesies yra siūlomas kaip resursas ir naudojamas panašiai kaip viešasis debesies.

Virtualus viešasis debesies suteikia daugiau galimybių išplečiamumui, bet esant mažiau galios darbo vienetui. Tad naudojant *baremetal* ir virtualaus debesies kombinaciją gali būti pasiektas reikiamas išplečiamumo ir veiksmingumo balansas (žr. 2.7 pav).

Kaip matyti iš žemiau pateikto grafiko, nuo DK modelio pasirinkimo priklauso veikimo veiksmingumo ir išplečiamumo (angl. *scalability*) santykis, tad akivaizdu, kad programos išplečiamumas IaaS yra susijęs su veiksmingumu.



2.7 pav. Veikimo veiksmingumo santykis su galėjimu išplėsti IT infrastruktūrą [10]

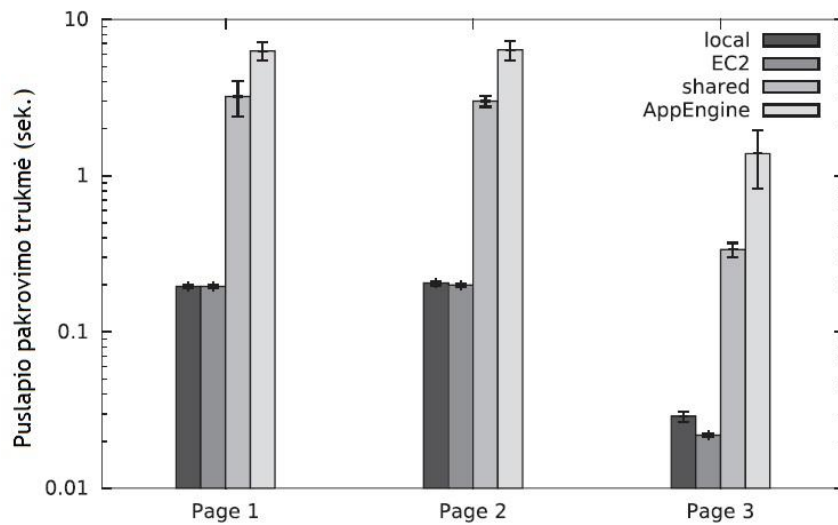
Debesyje paleistos programos veikimo veiksmingumas priklauso tiek nuo pasirinkto DK teikėjo, tiek nuo pasirinkto IaaS kopijos tipo. Anksčiau nagrinėtu atveju P. Maenhaut *et al.* pasirinko ištirti Darbų planuotojo veiksmingumą, kadangi ši programa buvo visiškai migruota į viešąjį debesį, įskaitant ir duomenų bazę (sudarytą iš 3-jų naudotojų įrašų), kai tuo tarpu Medicininio komunikavimo įranga buvo nevisiškai migruota, todėl jos veikimo veiksmingumas priklausė ir nuo daugiau faktorių, tokių kaip įmonėje likusio tinklo topologijos (sandaros) ir galimybių bei įmonės turimos fizinės įrangos. Buvo vertinami šie rodikliai:

- Vidutinis puslapio generavimo laikas (*PHP interpreter* pagalba); į tinklo delsą šiuo atveju neatsižvelgiama, nes jis generuojamas serverio pusėje (žr. 2.8 pav).
- Perdavimo tarp galinių taškų (įrenginių) laikas – bendras pakrovimo laikas žiūrint iš naudotojo pusės; atsižvelgiant į delsą tinkle (žr. 2.9 pav).

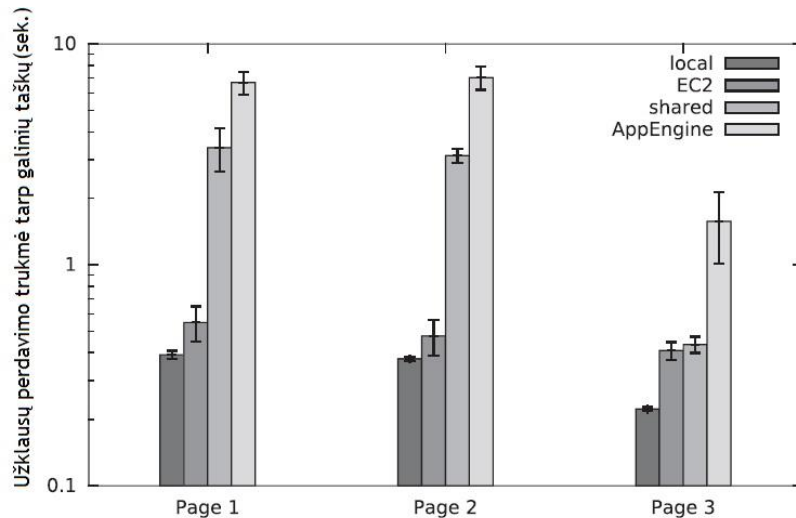
Tyrimai buvo atlikti 2015 m. sausio mėnesį, žemiau pateikiamos abiejų rodiklių palyginimo diagramos. Naudoti serveriai:

- Vietinis (angl. *local*) – priskirtoji VM su 1 GB RAM ir 1vCPU (paleista fiziniame Linux serveryje su *Intel Core i7* CPU (2.80 GHz) ir 8 GB atminties).
- Bendras (angl. *shared*) – bendrojo naudojimo interneto serveris.
- *EC2* – VM *t2.micro instance* su 1vCPU ir 1 GB RAM.
- *AppEngine* – VM kopija *Google AppEngine*.

Kaip matyti, tuo metu *Amazon EC2* parodė gerą veikimo veiksmingumą būdamas netgi kiek greitesnis nei vietinė VM, nors buvo panaudota mažiausia EC 2 kopija, *t2.micro instance*. *Google AppEngine* tuo tarpu buvo ganėtinai lėtas, nes iki 7 sekundžių užtrukdavo sudėtingo puslapio generavimas. Tai aiškinama eksperimentiniu ir neoptimizuotu PHP palaikymu *Google AppEngine* platformoje (autorius pastebėjo šios platformos padarytą pažangą nuo 2014 rugsėjo, kai tas pats puslapis būdavo generuojamas iki 45 sekundžių) (Maenhaut *et al.* 2015):



2.8 pav. Platformų palyginimas pagal vidutinį 3 puslapių užkrovimo laiką (per 50 kartų)



2.9 pav. Platformų palyginimas pagal užklausų perdavimo laiką tarp galinių taškų (3 p., 50 kartų)

Laikyti duomenis ir skaičiavimo išteklius šalia vienas kito yra laikoma gera praktika (Andrikopoulos *et al.* 2013). Tačiau dabartinės tendencijos tokios, kad procesoriaus galia pinga greičiau nei talpinimas diske, kuris savo ruožtu, tampa pigesnis greičiau nei tinklo pralaidumo kaina. Tai skatina iš naujo įvertinti programinės įrangos poreikius, kiek tai liečia skaičiavimo galią ir duomenų talpinimo apimtį, nes pastaroji gali kainuoti daugiau.

Programinės įrangos išplečiamumo ir perkeliamumo požiūriu svarbus programų skirstymas į:

- Be palaikymo atminčiai programa (angl. *stateless application*) yra tokia programa, kuri neįrašo sesijos metu sukurtų duomenų (pvz., informacijos apie naudotojo nustatymus ir įvykius), kad juos panaudoti sekančios sesijos su tuo pačiu naudotoju metu. Debesies kompiuterijai šis programų

tipas tinka geriau dėl išplečiamumo ir perkeliamumo, paprastesnio migravimo į debesį *forklift* būdu, o taip pat dėl to, kad debesies technologijos dažnai nebenaudoja būseną apibūdinančios informacijos.

- Su palaikymu atminčiai programa (angl. *stateful application*) – įrašo informaciją apie pokyčius sesijos metu. Tai reiškia, kad programa galės išsikviesti informaciją, perteikiančią pokyčius, iš vienos sesijos į kitą. Tačiau siekiant šią būsenos informaciją, esančią nutolusiame serveryje ar naudotojo galiniame įrenginyje, gauti, padidėja tinklo apkrovimas. Pavyzdžiui, interneto programos, kaip ir dauguma jas jungiančių API, yra programos su palaikymu atminčiai.

Kai bebūtų paradoksalu, bet resursai, kurių reikia pačios virtualizacijos programinės įrangos paleidimui ir tiekėjo serverio OS, neigiamai įtakoja bendrą sistemos veiksmingumą (Erl *et al.* 2013):

- Tiekėjo serverio OS pati naudoja procesoriaus galią, atmintį ir kitus IT resursus.
- Užklauso fizinei įrangai iš debesies klientų operacinių sistemų turi pereiti keletą sluoksnių į ir iš fizinės įrangos, kas sumažina bendrą veikimo veiksmingumą.

2.8. Resursų panaudojimo efektyvumas

Virtualizacijos modeliai ir pastoviai auganti fizinių įrenginių talpa bei veikimo veiksmingumas teigiamai įtakoja IT resursų konsolidavimą, nes sudėtingoms sistemų konfigūracijoms dabar reikia mažiau fizinių komponentų. Šie konsoliduoti IT resursai gali tarnauti skirtingoms sistemoms ir būti bendrai pasidalijami tarp skirtingų naudotojų (Erl *et al.* 2013).

Resursų panaudojimas susijęs su infrastruktūros **pajėgumais** (angl. *capacity*) aptarnauti organizacijos veiklos procesus. Pajėgumai suprantami kaip maksimalus darbo kiekis, kurį IT resursas gali atlikti per nustatytą laiko vienetą. Neatitikimas tarp IT resurso pajėgumo ir jo paklausos gali sistemą paversti arba neefektyvia (angl. *overprovisioning*) arba negalinčia patenkinti naudotojo poreikių (angl. *underprovisioning*). Pajėgumų planavimas skirtas sumažinti šiam neatitikimui, kad pasiekti planuojamą efektyvumą ir veikimo veiksmingumą.

IT resursas yra fizinis ar virtualus su IT susijęs produktas, kuris veikia arba programinės įrangos pagrindu (pvz., virtualus serveris) arba fizinės įrangos pagrindu (pvz., fizinis serveris arba tinklo įranga) (Erl *et al.* 2013).

Būtent resursų panaudojimo efektyvumu ir pasižymi IT infrastruktūros perkėlimas į DK. Reikia įvertinti ir gebėti suvaldyti šį iššūkį, kuriam apibūdinti toliau panagrinėsime debesies išplečiamumo, elastingumo ir daugianaudotojiškumo savybes.

2.8.1. Išplečiamumas (angl. scalability)

Svarbus tuo atveju, kai yra didėjantis duomenų nuskaitymo ir įrašymo apkrovimas. Šiai situacijai yra du pasirinkimai: vertikalus ir horizontalus išplečiamumas.

Vertikalus išplečiamumas (angl. *scale up/down*) – pavyzdžiui, 4-ių procesorių serverio pasirinkimas vietoj 2-ų procesorių serverio. Pradinėse programos gyvavimo ciklo stadijose, kai duomenų bazės apkrovimas yra nedidelis, įmanoma efektyviai naudoti mažo dydžio VM kopijas tiek *master*, tiek *slave* duomenų bazėms. Padidėjus apkrovai, galima *master* duomenų bazę migruoti į didesnio dydžio VM kopiją, kas leis pasinaudoti papildoma skaičiavimo galia, I/O apdorojimo sparta (angl. *throughput*) ir atmintimi.

Horizontalus išplečiamumas (angl. *scale out/in*), pavyzdžiui – 5 VM kopijos vietoje 2 VM kopijų. Tai yra tos pačios rūšies IT resursų užsisakymas/atsisakymas.

Automatinis išplečiamumas. Tipinėje (pvz., *Amazon*) IaaS platformoje elastingumo infrastruktūra įgalina VM kopijų automatinį išplečiamumą. Egzistuoja tam tikra automatinio išplečiamumo paleidimo veiksmų seka, pavyzdžiui, paleisti auto išplečiamumo algoritmą kas 5 minutes, arba jį aktyvuoti priklausomai nuo rodiklių reikšmės ir nustatytos ribos (pvz., serverio užimtumas > 80%) ir trukmės (pvz., vidurkio per paskutines 10 minučių) (Brebner 2012). Paskui paleidžiama resursų užsakymo užklausa (pvz., padidinti VM kopijų skaičių 1 vnt.)

Debesies infrastruktūra negali iš karto įvykdyti šios užklauso, nes turi:

- pirmiausia rasti ir rezervuoti prieinamą serverį;
- sukurti jame naują virtualios mašinos kopiją;
- įdiegti programos kodą ir kitus duomenis naujoje virtualioje mašinoje;
- paleisti programą;
- ir įtraukti naująją kopiją į apkrovos skirstytoją, kad ji galėtų būti pasiekama iš išorės.

Laikas, skirtas atlikti visiems šiems žingsniams atlikti dažnai vadinamas *spinup* trukme.

Išplečiamumo resursų taupymas. Patariama išjungti nebenaudojamas VM. Periodiškai peržiūrėti sistemos įrašų rinkmenas, kad suprasti kiekvienos VM naudojimo ir gyvavimo ciklo principus. Taip pat įvertinti migruotas programas veikiančias dėl galimai joms tekusių per didelio dydžio VM kopijų, pvz., vietoj *m1.large instance* (1X \$0.40/h) paskirstyti tą pačią apkrovą dviem mažesnėms *m1.small instances* (2 X \$0.10/h) (Varia 2010).

Be to, padidinus debesyje veikiančios programos efektyvumą, galima sutaupyti. Pavyzdžiui, jeigu programa nuolat perduoda dideles duomenų apimtis tarp *Amazon EC2* ir įmonės privataus

duomenų centro, gali apsimokėti suspausti duomenis prieš juos perduodant ryšio kabeliu. Tai gali leisti sutaupyti tiek duomenų perdavimo, tiek saugojimo kaštų.

Siūloma atsižvelgti į tai, kad programa sudaryta iš komponentų:

- Rekomenduojama, kad PĮ išplečiamumas būtų modulinis. Vietoj to, kad paprastai išplėsti virtualių mašinų kopijas visoms programoms, programos turėtų būti kuriamos taip, kad leistų pasirinktinai išplėsti tik tam tikrus komponentus (pvz., sudėtingiems skaičiavimams gali reikėti tik skaičiavimo resursų, bet ne duomenų bazių ar kietojo disko išplečiamumo).
- Užtikrinti, kad VM kopijose būtų įmanomas pritaikyti visų komponentų licencijavimas ir, kad galima būtų išmatuoti resursus, naudojančius specifinius komponentus, jeigu šie apmokestinami „per-use“ pagrindu (Schneider, Sunyaev 2015).

2.8.2. *Elastingumas (angl. elasticity)*

Tiesiogiai įtakojamas išplečiamumo, tačiau tai kiek platesnė sąvoka. Svarbūs šie faktoriai:

- Neatidėliotinas elastingumas – galėjimas iškart pateikti reikiamus resursus, kad aptarnauti papildomą paslaugos paklausą, esant staigiam srauto arba duomenų kiekio pakilimui. Patenkinus poreikį, resursai gali būti grąžinami į pradinę būseną.
- Pakankama talpa – jei išplečiamumas tam tikruose geografiniuose regionuose įmonei svarbus, turi būti užtikrinama, kad tiekėjas reikiamą talpą tuose regionuose garantuos (Erl *et al.* 2013).

Elastingumas gali būti vienas iš didesnių DK privalumų, bet tuo pačiu išlieka ir iššūkiu, jeigu atliekant duomenų ir programų migravimą reikalingas sistemos perdarymas ir pritaikymas elastingumui. Naudojant elastingumą yra išjungiamos/ijungiamos virtualios mašinos priklausomai nuo poreikio. Tačiau elastingumą naudoti gali būti sudėtinga ir brangu tarpusavyje sujungtose įmonės sistemose, kurios paremtos kitų sistemų nuolatiniu prieinamumu, ar sistemose naudojančiose *relational* duomenų bazes (lentelių rinkinys, kuriame duomenys yra sudėlioti į iš anksto nustatytas kategorijas), nes šios nėra lengvai išplečiamos.

Papildomai kyla ir saugumo bei privatumo problemos, bet šiuo atveju to galima išvengti naudojant duomenų šifravimą prieš migravimą ir talpinimą debesyje.

Elastingumas gali būti įgyvendinamas skirtinguose programos architektūros lygiuose. Kad įgyvendinti elastingumą gali reikėti perdirbti (angl. *refactor*) ar išskaidyti programą į komponentus, kad ši taptų labiau išplečiama. Kuo labiau pavyksta automatizuoti elastingumą programoje, tuo lengviau bus programą išplėsti horizontaliai ir pajusti DK naudą.

Duomenų nuskaitymo elastingumas paprastai pasiekiamas naudojant duomenų kopijas (angl. *replicas*) (Andrikopoulos, Strauch, Leymann 2013).

2.8.3. *Daugianaudotojiškumas (angl. multi-tenancy)*

Dėka šios technologijos keli ar daugiau naudotojų gali būti aprūpinti iš bendro kompiuterijos resursų katilo, įskaitant talpinimą, apdorojimą, atmintį ir tinklo pralaidumą; ji turi du pagrindinius aspektus (Strauch *et al.* 2014):

1) Komunikacijos kiekvienam naudotojui yra izoliuotos, t.y. leidžiant kiekvienam individualiai konfigūruoti ir valdyti jų komunikacijos galinius taškus programos arba paslaugų lygmenyje.

2) Toje pačioje sistemoje esančių naudotojų tarpusavio izoliacija kiek tai liečia duomenų ir veikimo veiksmingumo izoliavimą. Trijuose pagrindiniuose DK modeliuose, kaip minėta, stipriai skiriasi funkcionalumas, kiek tai liečia naudotojo galėjimą kontroliuoti debesies infrastruktūrą (Maenhaut *et al.* 2015).

2.9. Paslaugų kokybės užtikrinimas debesyje

Migravus IT infrastruktūrą į debesų kompiuteriją paslaugų kokybei apibūdinti dažniausiai naudojama prieinamumo (angl. *availability*) sąvoka, t.y. DK paslaugų tiekimo klientui nepertraukiamumas. IaaS klientams svarbūs tokie aspektai kaip susitarimas dėl paslaugų lygio, arba SLA (angl. *service level agreement*) ir jo sąlygos (pvz., dėl 99,97% paslaugų tiekimo nepertraukiamumo), debesies tiekėjo duomenų centro (naudotojui priklausančių duomenų laikymo) geografinės vietos, teisinio reguliavimo duomenų ir jų privatumo apsaugai ir pan.

SLA. Susitarimas dėl paslaugų lygio yra sutartis tarp paslaugos tiekėjo ir kliento, apibrėžianti paslaugas ir paslaugų lygio siekiamus rezultatus. Siūloma paslauga gali būti arba standartinė DK tiekėjo paslauga (angl. *off the shelf*) arba pasiūlymas, paremtas specifiniais DK paslaugų naudotojo reikalavimais. Debesies SLA apibrėžia parametrus ir matavimo rodiklius reikalingus paslaugų kokybės (angl. *Quality of Service*, QoS) monitoringui, o taip pat veiksmus, būtinus užtikrinti, kad ir besisiejantys aspektai (nefunkciniai reikalavimai) būtų valdomi ir suteikiami pagal kriterijus, apibrėžtus tam tikrai paslaugai (ETSI 2016).

Nefunkciniai reikalavimai apima, pavyzdžiui, duomenų apsaugą, saugumo, privatumo, intelektinės nuosavybės apsaugos, atsparumo gedimams atvejus, nenumatytų atvejų procedūras, duomenų ištrynimo procedūras, paslaugų nutraukimo procedūras ir kt. SLA yra išdėstomi QoS

parametrai ir jų nustatytos vertės, pvz., paslaugų nepertraukiamumo laikas, operacijų skaičius per laiko vienetą (duomenų perdavimui) ir naudotojų skaičius, kuris gali vienu metu pasiekti paslaugą.

Debesies SLA naudojimas jau visuotinai pripažįstamas, tačiau jo sąlygų (parametrų) ir rodiklių standartizavimas tebevyksta. 2015 metų viduryje daugiausia standartizavime pasistūmėjo ISO/IEC JTC1 SC38 WG3 grupės darbai dėl ISO/IEC 19086 standarto ir Debesies saugos aljanso (angl. *Cloud Security Alliance*, (CSA)) dėl saugumo ir pasitikėjimo rodiklių.

ISO/IEC 19086 standarte debesies SLA tikslas yra papildyti standartinį SLA reglamentavimą tais elementais, kurie būdingi DK. Debesies SLA skirsis priklausomai nuo paslaugos tipo, atsižvelgiant į siūlomos paslaugos sudėtingumą, svarbumą ir paslaugos valdomų duomenų tipą bei daugelį kitų faktorių.

Debesies SLA taip pat gali būti vienu iš kelių elementų, sudarančių **Bendrajį susitarimą dėl paslaugų** (angl. *Master Service Agreement*, MSA). Kiti MSA elementai gali būti:

- Tinkamo naudojimo politika (angl. *Acceptable Use Policy*, AUP) - nustato ir apibrėžia DK paslaugos naudotojo naudojamą DK paslaugą.
- Saugumo politika - apibrėžia tiekėjo ir naudotojo atsakomybes, siekiant užtikrinti tam tikrus saugos rezultatus.
- Privatumo politika - ji yra arba dalis saugumo politikos arba kaip atskira politika. Privatumo politika apima duomenų apsaugos priemones, kurių reikia užsibrėžtų privatumo politikos tikslų išlaikymui ir siekiant atitikti privatumo standartus ir sertifikatus, naudojamus tų tikslų įgyvendinimui.

QoS. Įmonei migravus savo programas į debesį, kokybės kriterijų valdymo kontrolė bus prarasta mainais į QoS, siūlomą iš tiekėjo pusės.

Pats akivaizdžiausias šios priklausomybės atvejis yra DK paslaugų teikimo nutrūkimo pasekmės įmonėms talpinančioms savo programas tame debesyje. Tiekėjo nesugebėjimas užtikrinti prieinamumo paprastai klientui yra kompensuojamas tik suteikiant kreditą būsimų paslaugų naudojimui. Be to, migruotos programos bendras prieinamumas gali būti stipriai įtakojamas ir pačios programos elgsenos, tad prognozuoti prieinamumą gali būti pakankamai sunku.

Be prieinamumo, programos QoS lygis debesyje yra įtakojamas dviejų pagrindinių faktorių: DK tiekėjo siūlomo PĮ veikimo veiksmingumo variacijų ir tinklo delsos tarp paslaugos naudotojų bei DK tiekėjo (Andrikopoulos *et al.* 2013). Veiksmingumo klausimą galima spręsti palyginant tarpusavyje DK paslaugų tiekėjus ir pasirinkant iš jų tinkamiausią. Tačiau tinklo delsa dažniausiai yra už tiekėjo kontrolės ribų, tokiu atveju SLA sąlygas būtų sudėtinga įgyvendinti.

SLA susitarimo įvertinimui turi būti atliekami šie žingsniai: 1) įvertinti kiekvienos šalies vaidmenį ir atsakomybę, 2) įvertinti veiklos lygmens politiką, 3) suprasti paslaugos ir įdiegimo modelio skirtumus, 4) identifikuoti veikimo veiksmingumo esminius kriterijus, 5) įvertinti saugumo ir privatumo reikalavimus, 6) įvertinti paslaugų valdymo reikalavimus, 7) pasirengti paslaugų netikėto nutrūkimo suvaldymui, 8) įvertinti gedimų atstatymo planą, 9) apibrėžti efektyvų valdymo procesą, 10) įvertinti paslaugų atsisakymo procedūrą (Cloud Standards Customer Council 2014).

Apsauga nuo paslaugos teikimo netikėto nutrūkimo. Debesies kompiuterijos paslaugos pagrįstos klasteriavimo principu, padedančiu išskirstyti gedimų riziką sugedus vienam iš įrenginių. Klasteris yra grupė nepriklausomų IT resursų, sujungtų tarpusavyje ir veikiančių kaip viena sistema (Erl *et al.* 2013). Sistemos gedimų tikimybė sumažėja, o paslaugų prieinamumas ir patikimumas išauga dėka to, kad sistema turi atkartojamumo (angl. *redundancy*) ir gedimų sprendimo atsarginių kopijų pagalba (angl. *failover*) savybes. Fizinės įrangos klasterio pagrindinė sąlyga – ją sudarančios sistemos turi turėti pakankamai identišką fizinę įrangą ir OS, idant būtų galima suteikti panašų veikimo veiksmingumo lygį, kai vienas sugedęs komponentas yra keičiamas kitu. Klasterį sudarantys įrenginiai sinchronizuojami ir sujungiami tarpusavyje greitaveikos ryšiu.

2.10. Debesies paslaugų tiekėjo pasirinkimas

Tiekėjams lenktyniaujant, kad patenkinti augančius IT paslaugų poreikius, debesies pasiūlymo pasirinkimo procesas tapo sudėtingesnis. Pravartu identifikuoti savo įmonės reikalavimus ir kriterijus, pagal kuriuos vertinami DK tiekėjai.

Siūlytina renkantis tiekėją užduoti šiuos klausimus (Interxion 2015):

Migravimas	Ar siūlomos migravimo į DK įgyvendinimo paslaugos gali padėti išvengti pernelyg staigaus naujos patirties perėmimo ir neviršys finansinių bei laiko apribojimų?
Programos architektūra	Ar architektūra debesyje, taikomi standartai ir paslaugos yra pritaikyti turimiems įmonės darbams ir valdymo poreikiams?
Tinklo architektūra	Ar tiekėjo naudojamas metodas izoliuoti vienų naudotojų atliekamas operacijas nuo kitų yra tinkamas įmonės poreikiams ir tinklo izoliavimo politikai?
Patogumas naudotis	Kiek geros tiekėjo siūlomos monitoringo paslaugos? Ar paprastas savarankiškas apsirūpinimas resursais, valdymas ir išplečiamumas?
Sąlygos ir apmokestinimas	Ar siūlomos sąlygos, kainos modelis ir apmokestinimo tipas suteikia norimą lankstumą?
Veiksmingumo garantijos	Ar SLA sąlygos atitinka įmonės poreikius ir ar siūlo aiškumą dėl veikimo veiksmingumo, naudojimo, kainų ir apmokestinimo modelio?
Saugumas	Ar suteikiamas saugumas ir ar taikomas tinkamas standartas? Ar sistema leidžia skirtingiems programos darbams susisiekti vienam su kitu išlaikant prieigos nustatymus dėl saugumo?

Veiklos tęstinumas	Ar tiekėjas siūlo atsparumą gedimams, verslo tęstinumą ir reikiamą atstatymo palaikymą?
Paslaugų valdymas	Ar aiškus ir tinkamas paslaugų valdymas? Ar tiekėjas palaiko įmonės keliamus reikalavimus?
Palaikymas	Koks palaikymas yra siūlomas ir kaip apmokestinamas?
Ateitis	Ar paslaugų teikėjas galės padėti įmonei dėl visų galimai įmonei ateityje iškiliančių debesies reikalavimų, nebūtinai tik dabartinių? Kokios galimybės išvengti neperkeliamumo rizikos – ar tiekėjas naudoja atvirus standartus ir API ir ar turi sutartyje numatytą debesies paslaugų atsisakymo procedūrą? Kokia bus išėjimo kaina?

Debesies paslaugas galima įsigyti tiesiai iš tiekėjo arba iš debesies paslaugų brokerio, perpardavinėjančio debesies platformos savininko teikiamas paslaugas. Tokia veikla gali suteikti papildomų galimybių (Talkin Cloud 2017):

- Tarpininkas suteikia pridėtinę vertę duodančias paslaugas, papildomai teikdamas jas jau egzistuojančių DK platformų pagrindu, pvz., tapatybės ar prieigos valdymą.
- Sąsajumas – tarpininkas „sulipdo“ kelias paslaugas, kad užtikrinti bendrą sistemų veikimą ir duomenų tarp jų judėjimą.
- Debesies paslaugų platus pasirinkimas suteikia lankstumo ir galimybę pragmatiškam sprendimui, nes rinktis siūloma iš daugybės panašių paslaugų.

Rinkoje siūlomi specialūs įrankiai, dažnai pasiekiami interneto sąsajos pagalba ir leidžiantys pagal norimus parametrus (procesoriaus galia, disko talpa, operacinė sistema, duomenų centro geografinė vieta ir kt.) pasirinkti paslaugų teikėją (žr. 2.10 pav). Nereikia tiekėjų ieškoti atskirai, nes viskas pateikiama vienoje vietoje, pavyzdžiui, *cloudorado.com*, *cloudscreener.com*, *planforcloud.com*.

The screenshot displays the Cloudorado.com search interface. On the left, there are filters for 'Location' (set to Europe), 'Cloud Features & Management' (with 'Hourly Pay-As-You-Go' checked), and a world map. The main area shows search criteria: RAM (4G selected), Storage (20 selected), CPU Power (2x selected), and OS (Linux selected). Below these, it states '7 cloud server providers found'. The results table lists providers, their server summaries, and prices per month.

Cloud Provider	Cloud Server Summary	Price per month	Action
antamc.net	General Purpose G2.4GB show details	\$40	Go to Provider
EXOScale	Medium - 4 GB RAM / 2 CPUs + 20 GB SSD show details	\$34 CHF 34	Go to Provider

2.10 pav. Cloudorado.com portale atliekama debesies teikėjų paieška pagal nustatytus kriterijus debesies serveriui [13]

2.11. Programos migravimo metodo pasirinkimas

Migravimo į DK būdo pasirinkimas yra susijęs su sprendimo migruoti į IaaS įgyvendinimu: kuri platforma ir migravimo technika siūlo didžiausią galimybę optimizuoti programas, kad jos labiausiai prisidėtų prie verslo ir IT procesų tikslų pasiekimo (plačiau PĮ adaptavimo debesies aplinkoje būdai analizuoti 1.4 poskyryje). Iš esmės pastaruoju metu praktikoje naudojami šie du PĮ perkėlimo į debesį metodai (Perucci 2016):

- Programų paėmimas iš senos sistemos ir perkėlimas į virtualizuotą infrastruktūrą.
- Programų (jų komponentų) perdarymas į debesiai skirtas (angl. *cloud-native*), kad jas paversti kiek įmanoma lankstesnėmis vystymui, diegimui, priežiūrai ir didesniai veiksmingumui.

Pirmasis būdas, programinės įrangos kaip vieneto (angl. *application stack*) perkėlimas yra klasikinis IaaS migravimas, kai programa visa apimtimi yra patalpinama debesies virtualioje mašinoje. Neatliekami papildomi sudėtingi adaptacijos debesies platformai veiksmai, tokie kaip programos kodo perrašymas. Vis tik tokiam migravimui būtina prielaida – galėjimas „ištraukti“ visą programą kaip vienetą ir jos nekeičiant perkelti į virtualiąją mašiną (Andrikopoulos *et al.* 2013). Žinoma, toks migravimo tipas nebus parankus visur ir visada priklausomai nuo to, kokio sudėtingumo programinę įrangą siekiama perkelti ir kokias galimybes (tiek technines, tiek išteklių požiūriu) bei poreikius įmonė turi.

Antrasis metodas paprastai reiškia senosios programos kodo perrašymą, tam, kad programa galėtų būti lengvai suderinama su debesies tiekėjo platforma ir visiškai pasinaudoti DK teikiama resursų efektyvumo nauda. Atliekama naudojant *re-architektūrinimą* ir *re-faktoringą*. Stambių komponentų programos paprastai nėra pakankamai lanksčios reaguoti į užklausų apkrovos pasikeitimus, tad nepasinaudoja debesyje siūlomais veiksmingumo pagerinimais. Dėl šios priežasties tokių programų architektūra yra pakeičiama į smulkius komponentus (Kwon, Tilevich 2014). Pasiekiamas optimalus išplečiamumas ir veiksmingumas, padaugėja DK diegimo alternatyvų keliuose debesyse.

Įmonė perdirbdama savo programas migravimui į IaaS turėtų išsiaiškinti šiuos dalykus (Varia 2010):

- Ar galima programą padalinti į komponentus ir juos paleisti atskirose VM kopijose. Pavyzdžiui, ar įmanoma interneto programą išskirstyti į atskirus komponentus (web, programos ir duomenų bazės) ir paleisti juos atskirose VM kopijose.
- Ar galimas programos dalinimas – apkrovos padalinimas didesniai skaičiui mažų virtualių mašinų vietoje kelių didelių.

- Ar galima nuo programos kodo atsieti diegimą ir konfigūraciją.

2.12. Vieningo standarto nebuvimas

Kaip matyti iš prieš tai nagrinėtų iššūkių ir migravimo modelių bei jų įgyvendinimo įvairovės, įmonės programų paleidimas debesies aplinkoje nėra iš anksto aiškus ir nuspėjamas procesas, o prie to prisideda ir tai, kad iki šiol nėra visuotinai pripažinto vieningo standarto. Vis dėlto, egzistuoja įvairios organizacijos, debesų kompiuterijos grupės, kurių tikslas - standartizuoti esančias praktikas, identifikuoti panašius reikalavimus. Viena jų, ETSI (Europos Telekomunikacijų Standartų institutas) savo 2016 metais išleistoje ataskaitoje išvardijo standartus, kiek tai liečia Perkeliamumą, Informacijos saugą, Privatumą, SLA ir Sertifikavimą (A priedas).

Tarp standartus rengiančių subjektų galima būtų paminėti šiuos: *IEEE Cloud Computing Standard Study Group* (IEEE CCSSG), *ITU Cloud Computing Focus Group*, *Cloud Security Alliance*, (CSA), *Distributed Management Task Force* (DMTF), *Storage Networking Industry Association* (SNIA), *Open Grid Forum* (OGF), *Open Cloud Consortium* (OCC), *Organization for the Advancement of Structured Information Standards* (OASIS) ir kiti. Siekdamas paskatinti DK naudojimą, šios organizacijos, nors ir turinčios skirtingų interesų, turėtų kartu parengti bendrus standartus.

Bendri standartai, paskatinsiantys perkeliamumą ir laisvą duomenų judėjimą tarp skirtingų tiekėjų platformų, būtų reikalingi bent šiose srityse: tinklo architektūra, duomenų formatas, matavimas ir apmokestinimas, paslaugų kokybė, resursų tiekimas, sauga, tapatybių valdymas ir privatumas (Rong *et al.* 2013).

Atviro kodo bendruomenės, tarp jų 2010 metais įkurtas projektas *OpenStack Community*, prisideda prie to, kad ateityje naudotojams būtų lengviau suderinti nuosavas IT sistemas ir debesies paslaugas. Siūlomos labiau pasaulio vystytojų bendruomenei prieinamos DK aplinkos su standartine fizine įranga. Jos lengviau integruojamos ir pagrįstos visų naudojamais įrankiais. Paleistas *Horizon* portalas, veikiantis kaip debesies valdymo prietaisų skydas (Internap 2016).

2.13. Perkeliamumas tarp debesies tiekėjų

Perkeliamumas (angl. *portability*) - galėjimas pasirinkti DK paslaugoms kitą tiekėją, taip pat yra vienas iš migravimo į DK iššūkių, apie kurį įmonė turėtų pradėti galvoti dar prieš imdamasi įgyvendinti savo IT infrastruktūros perkėlimo į IaaS projektą. Perkeliamumo laipsnis bus tuo didesnis, kuo mažiau pakeitimų, t.y. adaptacijos, programos perkėlimas į debesies aplinką pareikalavo, pvz., perrašant programos kodą arba migruojant tik tam tikras aplikacijos lygmens dalis į debesį (*re-*

faktoringo atveju). Tarp skirtingų DK modelių IaaS migravimas, kai migruojamos programos kaip vienetas ir viskas paleidžiama iš virtualios mašinos, siūlo, ko gero, didžiausią perkeliamumą (ETSI 2016).

Didesnis perkeliamumas naudingas DK naudotojui, nes šis nuo pat pradžių gali rinktis iš kuo daugiau tiekėjų ir jų pasiūlymus įvertinti nebijodamas prisirišti prie vieno konkretaus tiekėjo (angl. *vendor lock-in*), kas yra dažna, jeigu tiekėjo paslaugos suteikia specifines galimybes programinei įrangai.

Programos perkeliamumas DK (darbo apkrovos perkeliamumas) gali būti pasiekiamas naudojant: populiariausias programavimo kalbas, standartus, įrankius, programos veikimo (angl. *runtime*) elementus ir API. Galimas programos perkėlimo būdas yra „elgesio“ perkeliamumas, kai programos elgesys yra imituojamas naujoje programoje, nors ją sudarantys elementai gali būti visiškai kitokie (pvz., kiek tai liečia panaudotas technologijas).

Duomenų perkeliamumas suteikia galimybę be sunkumų perkelti duomenis iš pradinio DK tiekėjo į sekantį, nereikalaujant iš naujo suvesti duomenis. Čia svarbu įvertinti perkeliamų duomenų sintaksę ir semantiką. Jeigu sintaksė tarp buvusios ir būsimos DK aplinkų skirtinga, perkeliamumas vis tiek gali būti atliktas, nes yra įrankių, kurių pagalba galima atlikti duomenų transformaciją. Jeigu perkeliamų duomenų semantika neatitinka, duomenų perkėlimas bus sudėtingas arba visai neįmanomas.

Perkeliamumas debesies aplinkose yra jautrus klausimas, todėl reikės nemažai pastangų tiek kuriant tarptautinius standartus, tiek bendradarbiaujant ieškant kitokių šio klausimo sprendimo būdų. Šiuo atveju tai ypač liečia PaaS tiekėjus, kurie turėtų susitarti dėl apsikeitimo ir API mechanizmų, kuriuos galima būtų saugiai ir lengvai naudoti tarp skirtingų DK tiekėjų ir naudotojų.

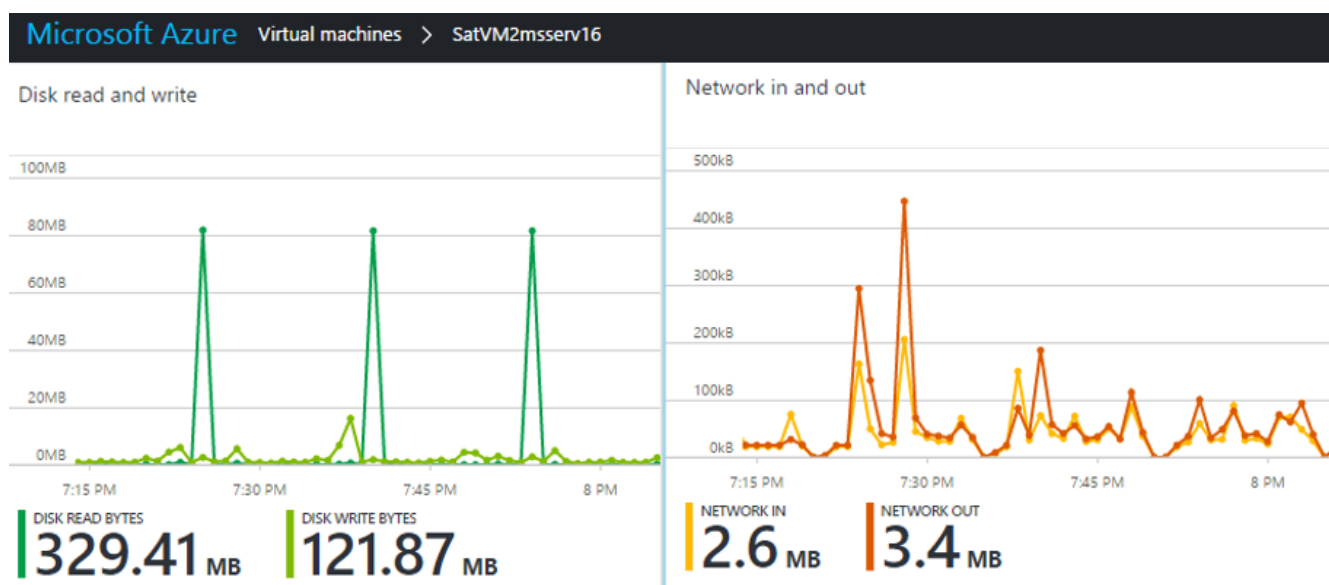
Galiausiai, naudotojai, siekiantys daugiau perkeliamumo, turėtų pirmiau atlikti kaštų ir naudos analizę, kad įvertinti šio veiksmo atsiperkamumą.

2.14. Rodiklių matavimas

Debesies aplinkoje įdiegtos įmonės PĮ veikimo veiksmingumas, naudojamų resursų kaštai ir kiti parametrai privalo būti nuolat peržiūrimi ir tinkamai įvertinami, jeigu iš migravimo į debesį siekiama gauti kuo didesnės naudos. Priklausomai nuo DK įgyvendinimo modelio, ne visus parametrus bus galima išmatuoti ir palyginti tarpusavyje (pavyzdžiui, galutiniam naudotojui testuojant viešąjį debesį, VM tankis ar galios rodikliai paprastai nėra išmatuojami) (SPEC OSGCloud working group 2012). Pagrindiniai rodikliai:

Elastingumas, sudarytas iš šių komponentų:	Kiti svarbūs rodikliai:
- Tiekimo intervalas (apibrėžiamas kaip laikas, reikalingas suteikti resursą arba imti jo nebeteikti, t.y. <i>spin-up time</i>) - Lankstumas (angl. <i>agility</i>) - Išplečiamumas į viršų/į apačią - Elastinis pagreitinimas - Apdorojimo sparta - Atsakymo laikas (angl. <i>reponse time</i>) - Veikimo veiksmingumo variacija	- Ilgaamžiškumas - Patikimumas - Galia - Kaina - Tankis (kiek VM gali būti efektyviai sudėta į <i>hypervizorių</i> neprarandant efektyvumo? OS licencijavimas ir <i>hypervizorių</i> kopijos yra tarpusavyje susiję, tad reikiamas <i>hypervizorių</i> kopijų skaičius gali išauginti kaštus)

Matavimo įrankiai. Monitoringas taip pat reikšmingas tobulinant perkeltos programos veiksmingumą debesyje, nes nesant rodiklių išmatavimo, nebus efektyvus ir pats tobulinimas. Kai kurie įrankiai pateikia sugeneruotus ar grafiškai atvaizduotus *log*’ų įrašus. Kiti įrankiai yra sukurti paties debesies paslaugų tiekėjo, lengvai instaliuojami ir realiam laike parodantys duomenų analizę, *log*’us ir perspėjimus. Pavyzdžiai: *CloudWatch*, *AWS Lambda*, *Application Insights Azure* (žr. 2.11 pav).



2.11 pav. Virtualaus serverio disko skaitymo/rašymo ir įeinančio/išeinančio tinklo srauto parametrų rodikliai Microsoft Azure platformoje [33]

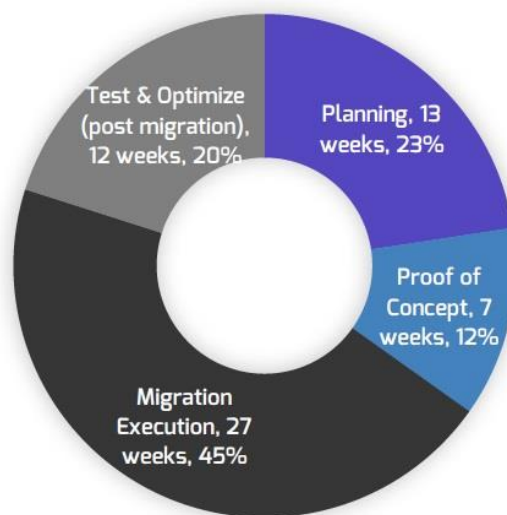
2.15. Migravimo trukmė

Migravimo įgyvendinimo proceso trukmė turi įtakos ne tik įmonės veiklai (nes gali būti įtakojami kasdieniai nepertraukiami verslo procesai, atsirasti žmogiškųjų ir kitų išteklių trūkumas), bet ir finansinėms sąnaudoms.

Migravimo į IaaS stadijų trukmė buvo tiriama *CloudEndure* vykdytoje 2016 Migravimo į debesį apklausoje, kai buvo apklausti 258 IT profesionalai iš įvairių valstybių. Rezultatai parodė, kad visas migravimo į debesų kompiuteriją procesas vidutiniškai trunka 12-16 mėnesių (žr. 2.12 pav).

Savaičių skaičius apskaičiuojamas žmonių, dalyvaujančių projekte skaičių, padauginus iš savaičių, sugaištų kiekvienai stadijai, skaičiaus (pvz., 3 asmenų 4 savaites rengiamo plano trukmė skaičiuojama kaip 12 savaičių)

Migravimo įgyvendinimo stadija truko vidutiniškai 6 mėnesius (27 žmogaus - savaitės) ir arti pusės (45%) viso projekto laiko, kai planavimas užtruko mažiau nei 3 mėnesius arba 13 savaičių (23%)



2.12 pav. Migravimo į IaaS trukmė pagal migravimo stadijas [11]

Tyrimo rezultatus iš esmės patvirtina ir kiti šaltiniai, pagal kuriuos migravimo į debesį strategijos nekuriamos daugiau nei 2 metams į priekį (Perucci 2016). Praktikoje dažnai šie planai trumpėja iki 18 mėnesių, arba vos iki 11 mėnesių, žinoma tai labai priklauso nuo migravimo įgyvendinimo būdo, apimties ir pačios programinės įrangos.

3. Migravimo procesas ir plano parengimas

Šioje darbo dalyje, atsižvelgiant į IT infrastruktūros migravimo iššūkių analizės rezultatus, bus sprendžiami šie du klausimai, apibūdinantys migravimo proceso planavimą:

1. Ar į įmonę turėtų migruoti savo programinę įrangą į debesies aplinką?
2. Kokius žingsnius atlikti, kad įgyvendinti sklandų migravimą į debesų kompiuteriją?

3.1. Sprendimas dėl migravimo

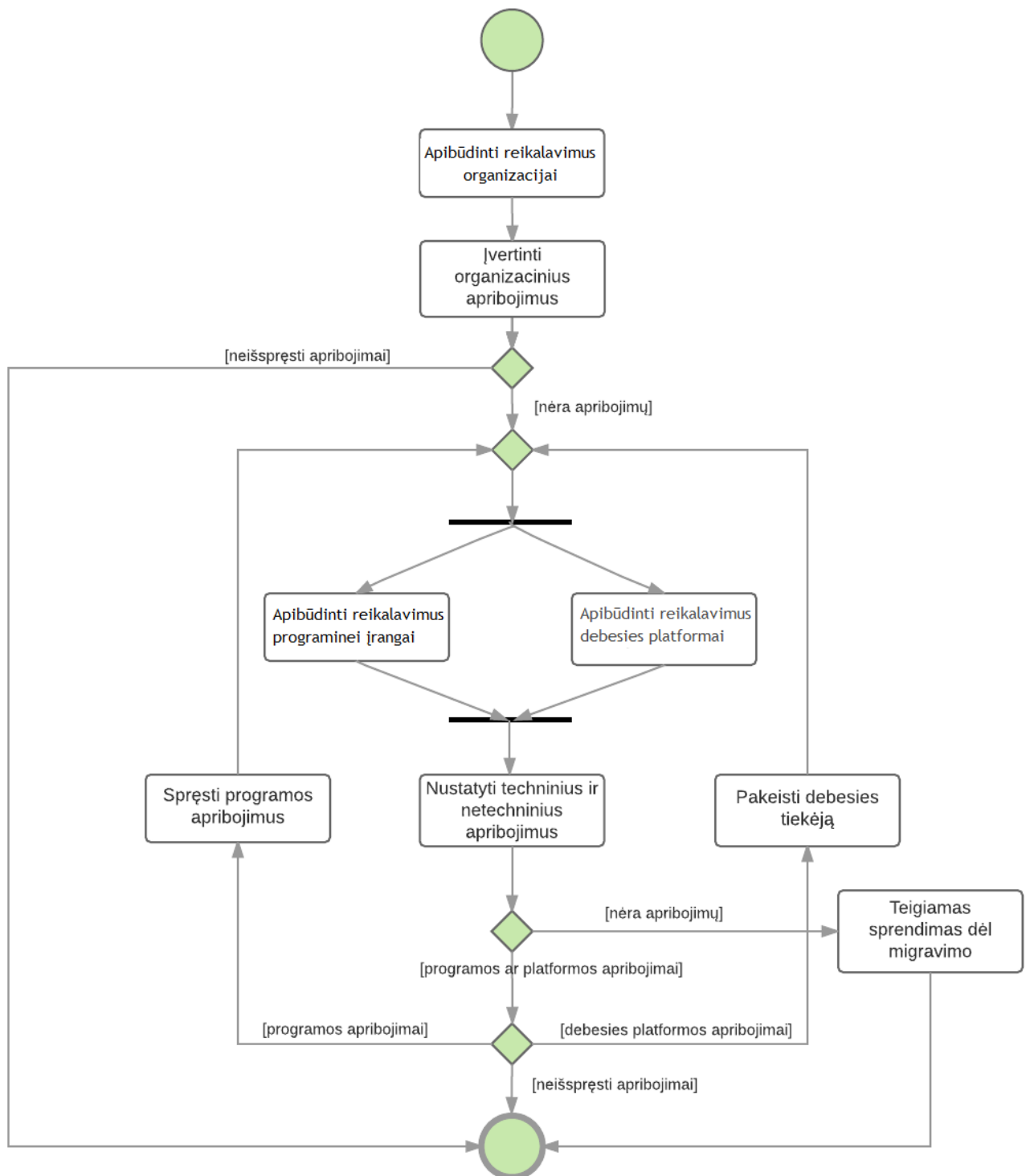
Organizacijai sprendžiant, ar apskritai verta migruoti į IaaS, kad paleisti nuosavą programinę įrangą iš debesies, prasminga naudoti *Cloudstep* procedūros pagrindu parengtą migravimo proceso modelį (žr. 3.1 pav). Jis leidžia detaliai išsiaiškinti sprendimo priėmimo kontekstą – 1) pačią organizaciją, 2) jos programinės įrangos savybes ir 3) potencialią debesies teikėjo platformą (Jamshidi *et al.* 2016).

Procesas yra sudarytas iš 8 veiksmų, kurių pagalba nustatomi egzistuojantys apribojimai (angl. *constraints*), be kurių išsprendimo įmonės IT infrastruktūros migravimas į DK laikomas negalimu. *Cloudstep* procesas aprašytas cikliška, t.y. modelį galima naudoti tiek kartų, kol nebus efektyviai išspręsti visi apribojimai, kas savo ruožtu reikš „taip“ migravimo procesui. Pažymėtina, kad modelis gali būti vienodai efektyvus ir migravimo įgyvendinimo metu. Migravimo proceso modelio aprašymas:

Apibūdinti reikalavimus organizacijai – teisinės ir administracinės organizacijos ypatybės (politikos, gairės, taikomi įstatymai, taisyklės). Nagrinėjami klausimai: kur įsigijami IT ištekliai? Kur vystomi ir testuojami bei diegiami programinės įrangos produktai ir paslaugos? Ar taikomi įstatymų ir kiti teisiniai apribojimai fizinei duomenų ar programų talpinimo vietai?

Įvertinti organizacinius apribojimus – pavyzdžiui, neatitikimas tarp įmonės patvirtintos politikos kritiniams atvejams ir tos, kurią siūlo teikėjas; rizika, kad neturintis leidimo asmuo prieš debesyje talpinamą kritinę verslo informaciją; IT išteklių valdymo pasikeitimo įtaka.

Apibūdinti reikalavimus programinei įrangai – užfiksujamos įmonėje esančios programos savybės, turinčios įtakos migracijai. Tai vėliau leis įvertinti jos tinkamumą potencialiems debesies architektūros modeliams. Ištiriamas jos naudojimas ir techniniai požymiai. Naudojimas: šio žingsnio tikslas - identifikuoti pagrindinius funkcinius ir nefunkcinius aspektus, kaip antai, kiek naudotojų ir iš kur turi prieigą prie programos, kokios jos naudojimosi ypatybės (užklausų pikai ar neaktyvumas),



3.1 pav. Sprendimo dėl programinės įrangos migravimo į debesų kompiuteriją priėmimo procesas

naudojimosi kaina. Techniniai požymiai: nusako technologijas, reikalingas įdiegti programai. Pavyzdžiui, dėl architektūros – kokie yra pagrindiniai komponentai ir jų tarpusavio sąsajos; dėl

technologijų – operacinė sistema, programavimo kalba, vystymo platforma, trečiųjų šalių komponentai, paleidimo aplinka. Duomenų valdymas ir duomenų perdavimas (ar yra griežtų QoS reikalavimų dėl prieinamumo, saugumo, veikimo veiksmingumo?). Kokia minimali konfigūracija reikalinga programos paleidimui? Ar yra kita programa ar sistema, nuo kurių veikimo priklauso ir migruojamos programos veikimas?

Apibūdinti reikalavimus debesies platformai – gali būti nagrinėjamos ir kelios DK platformos, įskaitant ir hibridinius debesis, siekiant nustatyti kaip jos atitinka (ar neatitinka) jau identifikuotus organizacinius ir techninius apribojimus. Nustatomi: paslaugų modeliai (IaaS), siūlomi resursai (VM, vieta talpinimui), kainų modeliai, SLA siūlomos garantijos, tiekėjo duomenų centrų vieta, papildomų paslaugų spektras (atsarginių kopijų kūrimas, monitoringas, auto išplečiamumas), saugumo mechanizmai, palaikomos technologijos (programavimo kalbos, PĮ licencijos).

Nustatyti techninius ir netechninius apribojimus – bendrai analizuojamas tarpusavio suderinamumas tarp organizacijos, programos ir debesies platformos apibūdinimų. Netechniniai apribojimai – organizaciniai ir finansiniai apribojimai. Techniniai – saugumo, komunikacijos – itin priklauso nuo įmonės naudojamų tinklo paslaugų kokybės (duomenų perdavimo sparta, pralaidumas, delsa), veikimo veiksmingumo, paslaugų nepetraukiamumo.

Spręsti programos apribojimus – šiuo žingsniu atliekama proceso iteracija, kur išsprendžiamas identifikuotas apribojimas.

Pakeisti debesies teikėją – atliekama iteracija, nes identifikavus konfliktus tarp platformų, gali reikėti nagrinėti naujos platformos savybes.

Teigiamas sprendimas dėl migravimo – procesas baigiamas, kai nesant neišspręstų apribojimų įmonė pritaria migravimui į DK. Po šio proceso rengiamas migravimo planas, kuriam tikslinga panaudoti šio modelio dėka jau surinktą ir papildomai gautą informaciją. Migravimo vykdymo metu organizacija gali sugrįžti prie šiame skyriuje aprašyto proceso modelio taikymo, išsprendžiant/apieinant nustatytus apribojimus, dėl ko migravimo proceso vykdymo metu galimos pakartotinės iteracijos.

3.2.Migravimo į debesies aplinką planas

Migravimo planavimo procese siekiama spręsti klausimą – kaip parengti įmonės IT infrastruktūros ir programinės įrangos migravimo į debesies aplinką planą atsižvelgiant į nustatytus ir išnagrinėtus migravimo iššūkius ir atlikus įmonės poreikių ir galimybių analizę, kurių pagrindu bus vykdoma architektūrinė paveldėtų sistemų migracija (Ahmad, Babar 2014).

Toliau aprašomos migravimo plano stadijos ir žingsniai. Planas rengtas rankiniu būdu, o procesas atliekamas vadovaujantis „iš apačios į viršų“ metodu – iš viso migravimo į debesies aplinką konteksto išskiriami žingsniai, kurie sudėliojami nuoseklia seka, o galiausiai sugrupuojami į stadijas. Iš esmės plane stadijos seka viena po kitos, tačiau taikant planą praktikoje neretais atvejais bus reikalinga grįžti į plano pradžią, kad išspręsti išaiškėjusius apribojimus ar optimizuoti migruotos PĮ veikimo veiksmingumą. Šiuo aspektu migravimo plano ciklas atitinka programinės įrangos gyvavimo ciklą, kuris irgi gali būti kartojamas esant poreikiui. Tad migracijos planas turi iteracinį pobūdį, leidžiantį grįžti į ankstesnę stadiją ir patikslinti migracijos detales.

Pažymėtina, kad 1-oji plano stadija „Analizė“ iš esmės atitinka ankstesniame poskyryje aprašyto proceso modelio žingsnius, todėl ši stadija gali iš esmės sutrumpėti, nes nagrinėjamoji informacija gali būti iš esmės tapati ir jau įvertinta. Be to, šio plano kai kurie žingsniai neprivalės būti įgyvendinami, pavyzdžiui, jeigu PĮ migravimas nereikalauja programos kodo transformacijos.

I. ANALIZĖ. Šioje stadijoje nustatoma, ar programos migravimas yra būtinas verslo tikslams pasiekti. Siekiama suprasti, kokios įmonės veiklos srities ir kokio dydžio problemą tikimasi išspręsti, bei identifikuojami sprendimai. Atlikus žingsnius surinkta informacija užfiksuojama dokumentuose.

1. Galimybių studija – vertinami TCO, kiti kaštai, migravimo įgyvendinimo trukmė, resursai, nauda, jau padarytos investicijos.

a) Veiklos kriterijai – šiame etape siekiama nustatyti pagrindinę migracijos motyvaciją ir užsibrėžtus tikslus. Veiklos kriterijai tada išskirstomi į detalesnius funkcinis ir/ar nefunkcinis reikalavimus, kurie paskui gali būti vertinami kokybiškai ar kiekybiškai:

- Sąnaudų analizė.
- Paslaugų lygis (PĮ veikimo veiksmingumas, saugumas ir privatumas, reguliacinė aplinka).
- Įtaka veiklai (įtaka pajamoms; naujų naudotojų įgijimas ar senų pririšimas; naudotojų pasitenkinimas; laiko, kad pateikti į rinką, sutrumpėjimas; kaštai piko apkrovų apdorojimui).

b) Organizaciniai kriterijai (ar atitiks: praktinius kriterijus (laikas, resursai, galimybės); socialinius faktorius (vertybės, pasitenkinimas); politinius faktorius (skaidrumas dėl procedūrų ir gaunamos naudos, rizikos). Įtaka suinteresuotiems subjektams).

c) Techniniai kriterijai (programos tipas (ne visos programos gali turėti naudos iš DK); elastingumas, tinklo pralaidumas, prieiga prie įrangos; netinkamos kandidatės: programos, kurios yra

itin integruotos, daug integravimo taškų/sąsajų, tad reikės daugiau pastangų jų migravimui atlikti. Taip pat programos su sudėtinėmis duomenų bazėmis).

2. Reikalavimų analizė - sprendžiama, kokios programos (komponentai) ir duomenys gali būti keliami į debesį (Sun, Li 2013):

a) Identifikuojamos funkcionuojančios informacinės sistemos, ryšiai tarp jų, o taip pat techninės ir programinės įrangos būklė. Renkama konfigūracijos informacija:

- apie visos sistemos įdiegimą – kiek yra fizinių serverių, kokia tinklo topologija;
- apie programos įdiegimą įmonės fiziniuose serveriuose;
- apie kiekvieną programos komponentą (OS, tarpinė PĮ (angl. *middleware*)).

b) Funkcinių ir nefuncinių reikalavimų analizė:

- procesoriaus galia, atmintis, talpa, I/O, tinklo duomenų pralaidumas, energijos suvartojimas;
- saugumo ir privatumo rizika, teisiniai apribojimai, įtaka kasdienei veiklai.

3. Potencialių debesies talpinimo platformų (aplinkų) analizė. Atsižvelgiama į projekto pobūdį ir aukščiau nagrinėtų reikalavimų bei galimybių studijos rezultatus (Chauhan, Babar 2012). Potencialių debesų pasirinkimui taip pat daro įtaką ir ilgalaikiai įsipareigojimai su konkrečiais DK tiekėjais.

a) Nustatoma, kokių paslaugų reikės.

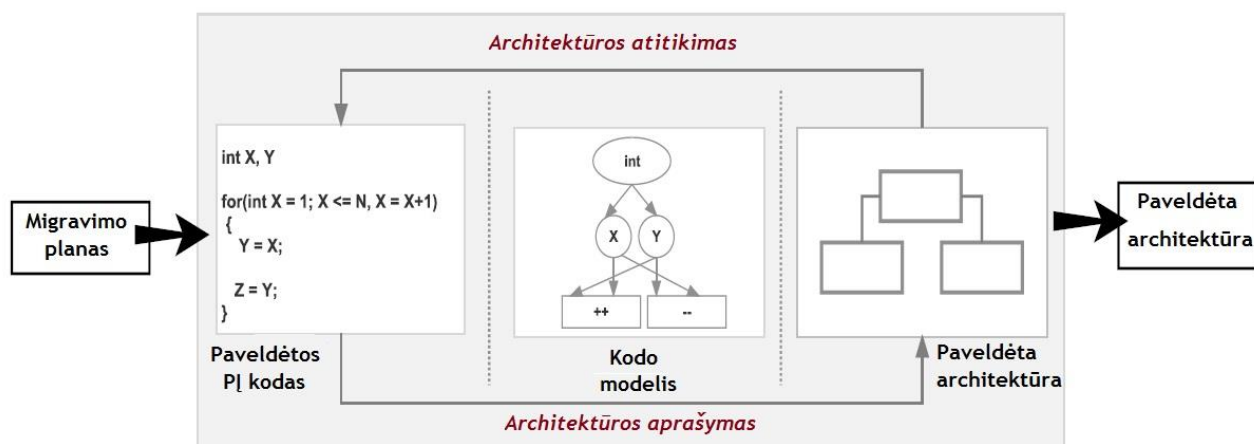
b) Identifikuojami DK teikėjų pasiūlymo aspektai: išplečiamumas, prieinamumas, saugumas, perkeliamumas, įkainių modelis, talpinimo galimybės ir geografinė, reglamentavimas.

c) Debesies aplinkų specifiniams debesų kompiuterijos kokybės kriterijams įvertinimas. Identifikuojami reikiami daryti pakeitimai nesuderinamumo klausimams išspręsti.

Sprendimo dėl migravimo į debesų kompiuteriją priėmimo taškas.

II. MODELIAVIMAS.

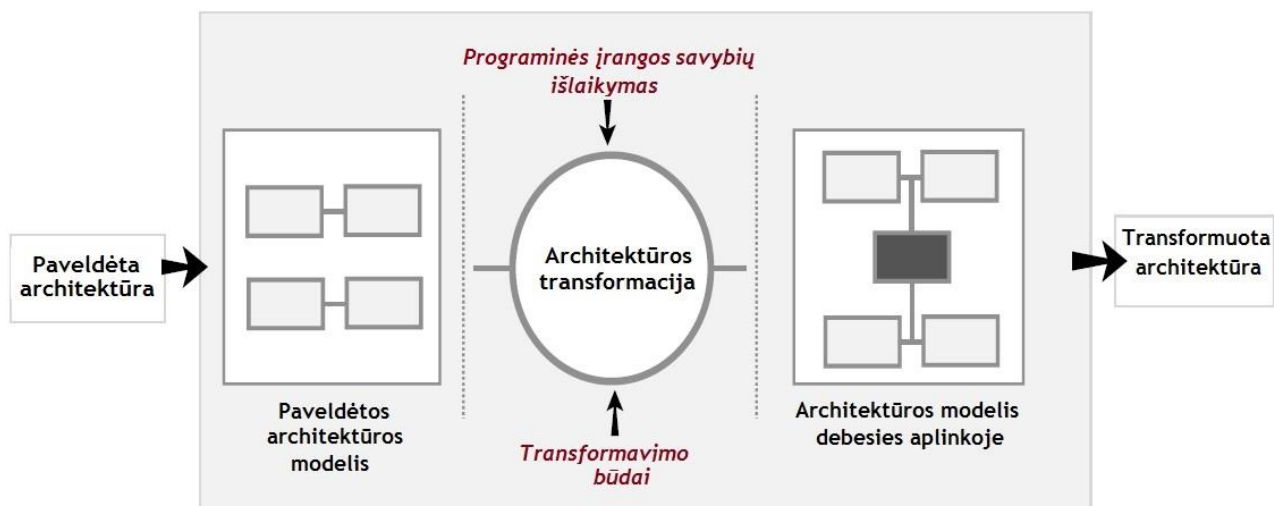
4. Esamos PĮ architektūros modelio parengimas. Sudaromas esamos PĮ architektūros modelis, nusakantis priklausomybes tarp programos komponentų, duomenų modelio ir sąsajos (Ahmad, Babar 2014):



3.2 pav. Esamos programinės įrangos architektūros modelio rengimas

- Identifikuojamas ir apibendrinamas esamas programinis kodas.
- Esamo kodo modeliavimas (atvaizdavimas).
- Esamos architektūros modeliavimas ir atitikimo pradiniam kodui patikrinimas.

5. Potencialaus PĮ architektūros modelio sukūrimas. Jis tenkins naujos sistemos reikalavimus ir sudaromas pagal analizės stadijoje surinktą informaciją apie įmonės senąją sistemą ir galimas DK platformas.



3.3 pav. Potencialaus programinės įrangos DK architektūros modelio rengimas

a) Migruojamų programos komponentų išrinkimas. V. Andrikopoulos pažymi, kad juos renkantis, atsižvelgiama į šiuos faktorius: duomenų privatumas, numatomas darbinių operacijų tipas, priimtina tinklo delsa ir veikimo veiksmingumo pokyčiai, DK serverių geografinė vieta, komponentų smulkumas debesyje.

b) Sukuriamas žemėlapis (angl. *mapping*) tarp siūlomų PĮ sprendimų ir potencialių debesų tiekėjų aplinkų palaikomų kokybės reikalavimų. Šią procedūrą užbaigus turėsime duomenų modelį:

- VM įdiegimo lygmuo.
- OS lygmuo.
- Tarpinės programinės įrangos lygmuo.
- Programos lygmuo.

c) Parengti bandomąjį modelį (angl. *proof of concept*)– siekiama kuo anksčiau pademonstruoti naujo modelio funkcionalumą ir privalumus suinteresuotiems subjektams.

d) Identifikuoti konfliktus tarp siūlomo sprendimo ir debesies aplinkos - esant neatitikimams, šios stadijos ankstesni žingsniai gali būti pakartojami.

6. Migravimo strategijos aprašymas.

- a) Migravimo būdas (pvz., perkėlimas, *re-architektūrinimas*, hibridinis).
- b) Migruotos į DK PĮ architektūros modelis.
- c) Migravimo etapai ir jų trukmė.
- d) Migravimo alternatyvos (jei yra).
- e) Parengiamos įgyvendinimo, ryšio užtikrinimo, migravimo paliestų darbuotojų, debesies rizikos valdymo, debesies tiekėjo valdymo strategijos (Conway, Curry 2012).
- f) Detalizuojama perkama debesies paslauga (kaip ji bus valdoma, kaip jungsis su egzistuojančiomis (likusiomis) sistemomis, apmokestinimo modelis).

7. Pasirenkamas tiekėjas.

- a) Atrenkamas tiekėjas – tiekėjo galimybių patikrinimas, derybos dėl SLA.
- b) Paruošiamas atsarginis planas, jeigu nepavyktų migracijos procesas.
- c) Pasirašoma sutartis su tiekėju, SLA ir įkainių dokumentai.

III.TRANSFORMACIJA. Šioje stadijoje atliekami realūs programos migravimo veiksmai.

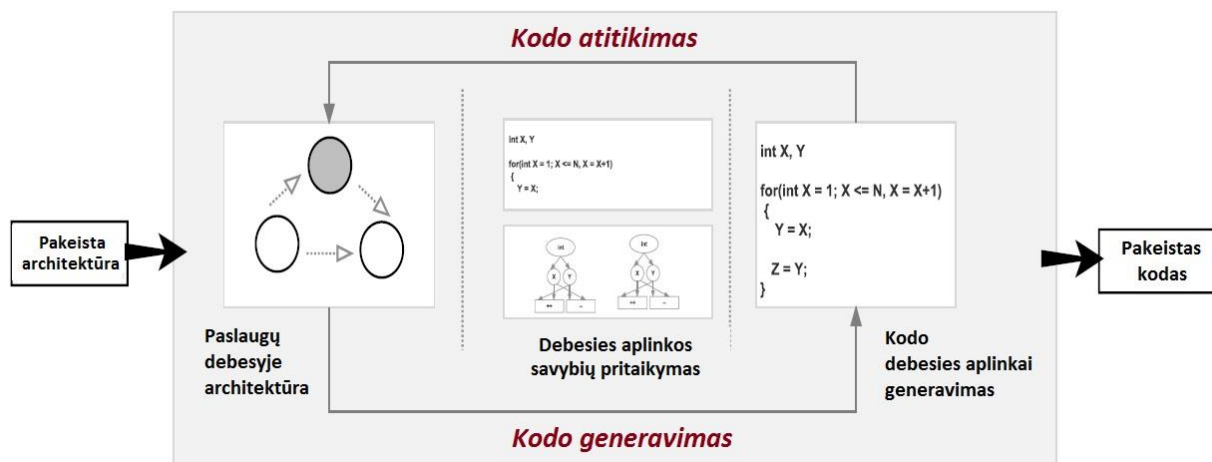
8. Projekto vykdymo komandos sudarymas.

- a) Reikalingų ir turimų IT kompetencijų įvertinimas, mokymai, nauji darbuotojai.
- b) Vaidmenų, užduočių ir atsakomybių priskyrimas.

9. Atsarginių kopijų kūrimas.

- a) Įmonės migruojamų duomenų bazių, failų sistemos ir sistemos būsenos (VM atvaizdai (angl. *snapshots*)), specifinių programų kopijavimas.
- b) Kopijų vientisumo patikrinimas (testavimas).

10. Programos kodo transformavimas. Debesies aplinkai pritaikyto kodo kūrimas (Ahmad, Babar 2014):



3.4 pav. Programos kodo transformavimas

a) Išsprendžiamas suderinamumas:

- Programos kodo *re-factoringas* – modifikuojamas senas programos kodas, kad galėtų veikti su DK paslaugomis.
 - Vystomi integratoriai/adaptatoriai. Jeigu *re-factoringas* brangus, tai gali būti alternatyva nesuderinamumams išspręsti. Adaptatoriai suteikia abstrakcijos sluoksnį, neliesdami programos kodo.
 - Duomenų adaptavimas – duomenų konversija, transformacija.
- b) Pritaikomi DK architektūros modelio kūrimo principai:
- Programos išskaidymas į komponentus - palengvina jų elastingumą ir išplečiamumą.
 - Programavimas be palaikymo atminčiai programų (angl. *stateless application*) – minimizuojami komponentų naudojami kontekstiniai duomenys.
 - Veikimo veiksmingumo svyravimo išsprendimas (angl. *transient fault handling*).

IV. MIGRACIJA. Veikiančios sistemos įdiegimas pasirinktame debesyje arba jų grupėje.

11. Diegimas. Darbas atliekamas paprastai mažo įmonės veiklos aktyvumo laikotarpiu.

a) PĮ paleidimo aplinkos debesies platformoje kūrimas. DK naują aplinką apima fiziniai serveriai, *hypervizoriai*, VM, OS, tarpinė PĮ ir migruojamos programos:

- Virtualaus tinklo infrastruktūros įdiegimas (viešame debesyje tai jau būna padaryta).
- Talpinimo vienetų (angl. *storage units*) debesyje sukūrimas.
- Individualių VM sukūrimas ir jų susiejimas su talpinimo vienetais.

- Įdiejami saugumo įrankiai – ugniasienės ir VPN maršrutizatoriai.
- b) Instaliuojama programinė įranga debesies aplinkoje:
 - Programų integravimas tarpusavyje ir su likusiomis įmonės viduje.
 - Įdiejami monitoringo ir DK resursų valdymo įrankiai.
 - Įdiegiama ir konfigūruojama PĮ, sauganti nuo kenksmingo programinio kodo ir kitų grėsmių.

c) Importuojami duomenys – gali vykti paraleliai su programos migravimu.

d) Aktualių duomenų atsarginių kopijų kūrimas migravimo eigoje.

12. Naujos aplinkos konfigūravimas.

a) Konfigūravimo automatizavimas. Naudojami VM šablonai, specialūs įrankiai pvz., *Chef*, *Puppet*, *AWS Elastic Beanstalk*.

b) Konfigūracijos parametrų duomenų perkėlimas į debesies aplinką (Cloud Standards Customer Council 2013).

13. Testavimas ir optimizavimas. Skirtas spręsti PĮ nesuderinamumui ir kitiems iškilusiems klausimams. Paprastai atliekamas kokybės užtikrinimo specialistų – kad nustatyti, ar nauja struktūra atitinka pradinius verslo tikslus. Testavimas gali būti kartojamas, ieškant klaidų ar nesuderinamumų, kol galutiniam naudotojui sistema tampa priimtina.

a) Aktyvuojamas SLA.

b) Testavimas (patvirtinti duomenų ir PĮ migravimą, sąsajumą, tinkamą prieigą bei valdymą).

c) Naudotojų atliekamas funkcionalumo patikrinimas.

d) Migravimo plano pakeitimas, žingsnių pakartojimas priklausomai nuo testavimo rezultatų.

e) Užbaigiamas senos sistemos atsarginių kopijų darymas.

14. Naujos sistemos patvirtinimas.

a) Pomigracinis testavimas, priimtino naudotojams tikrinimas.

b) Rezultatų ir naujos sistemos dokumentavimas.

15. Programos paleidimas naudojimui.

a) Naudotojų informavimas apie senos sistemos išjungimą.

b) Naujosios aplinkos paleidimas įmonės darbinėje aplinkoje.

16. PĮ rodiklių stebėseną ir priežiūra.

a) Tiekėjų valdymas vietoj vidinio resursų valdymo (DK paslaugų priežiūra integruojama su esamomis monitoringo ir ataskaitų sistemomis; periodiniai PĮ veikimo veiksmingumo matavimai).

b) Apsauga nuo netikėto debesies paslaugų tiekimo nutrūkimo. Pastovus atsarginių kopijų kūrimas ir sinchronizavimas.

c) Nenaudojamų migruotos PĮ kopijų deaktyvavimas.

17. Optimizavimas. Įvertinimui galima naudoti PDCA (angl. *Plan-Do-Check-Act*) ciklo procesą (Conway, Curry 2017). Stebima rinka, DK technologijų tendencijos, tiekėjų siūlymai, peržiūrimi reikalavimai PĮ.

a) Siekiama padidinti veikimo veiksmingumą.

b) Siekiama pridėti papildomų galimybių ar naudotojų iškeltų reikalavimų.

PĮ migravimo į debesų kompiuteriją stadijas ir žingsnius patogiu atvaizduoti diagramoje, šiuo atveju parengtoje pagal japonų mokslininko Kaoru Ishikawa *Fishbone* modelį (B priedas).

3.3. Hipotetinės įmonės IT infrastruktūros perkėlimas į debesų kompiuteriją

Hipotetinė įmonė *Hosen* yra vidutinė, apie 200 darbuotojų turinti tarptautinė privati logistikos įmonė, turinti 10 padalinių (logistikos sandėlių) išsidėsčiusių skirtingose Europos valstybėse. Įmonė organizuoja ir koordinuoja dviejų rūšių prekių sandėliavimą ir pristatymą: a) buitinės technikos ir b) pažangios medicininės ir biotechnologijų įrangos. Pirmosios prekių grupės, buitinės technikos, prekyba, o kartu ir susijusi logistikos veikla, pasižymi sezoniškumu, su keliais piko periodais per didžiąsias metų šventes. Antrosios prekių grupės pristatymai irgi yra nepastovūs ir itin imlūs ištekliams, nes įmonė dirba su stambiais projektais ir dideliais vienkartiniais pristatymais.

Įmonė pastebėjo, kad naudojama nuosava IT infrastruktūra ne visada užtikrina sklandžią padalinių veiklą, ypač kiek tai liečia paramą procesams tarp įmonės pagrindinės būstinės Lietuvoje ir jos padalinių užsienyje. Taip pat veiklos procesus įtakoja pasitaikantis netikėtas serverių išsijungimas, programų nesuderinamumas (padaliniai naudoja ir savarankiškai įsigytas bei įdiegtas programas), nepakankama kibernetinė sauga, rizika dėl duomenų nutekėjimo, padalinių sunkumai apdorojant užklausas piko periodais, sudėtingas ir brangus PĮ palaikymas. Iš kilo būtinybė rasti būdų padidinti efektyvumui, nes dalį veiklos laiko serverių resursai iki galo neišnaudojami. Be to, siekiama centralizuoti IT infrastruktūros stebėseną ir techninę priežiūrą, PĮ atnaujinimus, apsisaugoti nuo netikėtų išsijungimų ir kibernetinių grėsmių, geriau atlikti didelės apimties duomenų analizę.

Analizė.

1. Organizacijos ir jos programinės įrangos apibūdinimas:

Verslo procesams aptarnauti įmonė naudoja šias pagrindines sistemas:

- ERP PĮ (*Microsoft Dynamics*), veikianti centrinėje būstinėje ir skirta apdoroti visiems pagrindiniams resursų valdymo procesams. Į ERP bendrai suplaukiantys duomenys talpinami dviejuose centrinės būstinės serveriuose įdiegtose *SQL Server* duomenų bazėse.

- Padaliniuose (serveriuose) veikia Sandėlių valdymo ir užklausų apdorojimo PĮ, sujungta su centrine ERP sistema. Abiejų sistemų susiejimas (ERP ir Sandėlių valdymo PĮ) užtikrina, kad prekės pristatomos prieš tai patikrinus užsakymų duomenis, apmokėjimo ir pristatymo terminus. Sandėlių valdymo PĮ yra *ASP.NET* interneto programa, o jos galutiniai naudotojai – tiek įmonės padalinių darbuotojai, tiek tiekėjų atstovai. Pasinaudodami interneto sąsaja, galutiniai naudotojai gali nusiųsti/parsisiųsti dokumentus, keisti konfigūracijos nustatymus, generuoti ataskaitas. Padaliniuose įvedami duomenys yra talpinami *SQL Server* duomenų bazėje.

- Kaip operacinė sistema visuose serveriuose instaliuota *Microsoft Server OS*.

2. IT fizinės infrastruktūros ir personalo apibūdinimas:

- Centrinės būstinės IT infrastruktūra sudaryta iš 4 serverių (2 programinės įrangos serveriai ir 2 talpinimo serveriai), o padaliniuose yra po 2 serverius (1 programinės įrangos ir 1 talpinimo serveris). Tad įmonė valdo 24 serverius; iš viso 8 terabaitus (TB) duomenų.

- Jautrūs duomenys laikomi serveryje įmonės centrinės būstinės IT infrastruktūroje. Kai naudotojams iš padalinių reikia prieigos prie įmonės sistemų, naudojamas VPN ryšys.

- Centrinėje būstinėje dirba 3 IT specialistai, o padaliniuose - po 2 IT specialistus.

- Įmonės darbuotojai kasdienėje veikloje naudoja apie 200 asmeninių kompiuterių (galinių įrenginių).

3. Apribojimai migracijai:

- Saugumas – įmonės serveriuose iki šiol laikyti duomenys apie klientus, pristatomos įrangos dokumentacija, sutartys (kainos, kiekiai) su klientais yra konfidencialios.

- ERP programinė įranga šiuo metu negali būti perkeliama dėl jos svarbos verslui. Norima prieš tai atlikti sėkmingą kitos PĮ perkėlimą.

- ERP (įmonės viduje) ir Sandėlių ir užklausų valdymo sistemos (debesyje) naudojamų susisiekimo protokolų suderinamumas. Nėra garantijos, kad jie bus palaikomi debesyje, kad atsakymo laikas bus ne per ilgas, atsižvelgiant į apkrovas ir ypač plačiajuosčio ryšio su debesimi, kokybę.

- Būtinybė esant poreikiui bet kada papildyti programų funkcionalumą, todėl SaaS varianto buvo atsisakyta.

4. Migravimo į debesį sąnaudų įvertinimas:

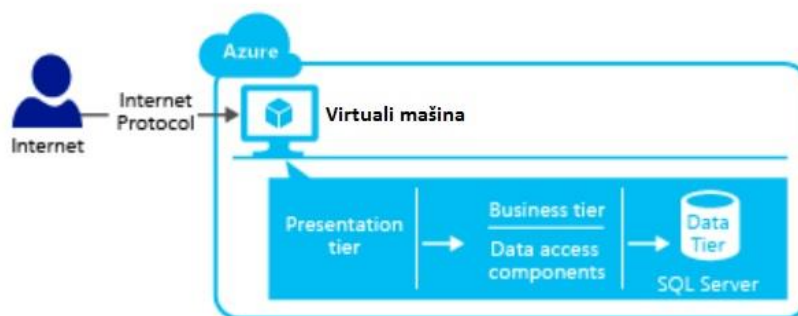
- Galimybių studija parodė, kad įmonės išlaidos IT infrastruktūrai išlaikyti sudaro svarbią dalį bendrose sąnaudose, ypač kiekvieno iš padalinių sąnaudose.
- Įvertinami tiesioginiai ir netiesioginiai migravimo į DK kaštai.
- Atliekamas preliminarus perkeliamos IT infrastruktūros 1 metų kainos įmonės viduje ir tapačių paslaugų debesyje skaičiavimas ir palyginimas (C priedas).

5. Įmonė pasirinko IaaS paslaugų modelį, tiekiamą viešojoje Microsoft Azure platformoje, dėl:

- Azure suderinamumo su jau turima PĮ (pvz., ERP). Dėl jau naudojamos Microsoft PĮ integravimo procesas nebus itin sudėtingas, nes *Hosen* galės tęsti darbą su dalimi senųjų sistemų.
- Azure siūlomi atitikimo standartams pažymėjimai tiek techninių saugiklių atžvilgiu, tiek saugumo ir duomenų privatumo atžvilgiu. Tai svarbu įmonei saugant savo duomenis nuo konkurentų, o kas liečia medicininės įrangos pristatymą – siekiant atitikti reguliacinius reikalavimus. Azure platforma atitinka ISO (27001 serijos), CSA (*Cloud Security Alliance*), HIPAA (*Health Insurance Portability and Accountability Act*), NHS (*National Health System*) *IG Toolkit*, SOC 1, 2, 3 Audit standartus.
- Sistemos valdymas ir priežiūra bus paprastesnis, nes Azure pasirūpins paslaugų operacijų valdymo ir platformos atnaujinimų diegimo užtikrinimu.
- Azure bendradarbiauja su nemažai PĮ tiekėjų, kurių suderinti su platforma produktai yra prieinami Azure aplinkoje.
- Numatoma, kad po migravimo įmonėje dirbs ne daugiau 12 IT specialistų.

Modeliavimas ir diegimas.

- Atsisakoma serverių, tinklo ir kitos fizinės įrangos. Serverių OS ir Sandėlių ir užsakymų valdymo programinė įranga bei susijusios duomenų bazės po perkėlimo į debesį bus paleidžiami iš VM debesies aplinkoje (žr. 3.5 pav).



3.5 pav. Programinės įrangos sluoksnių talpinimas vienoje VM [33]

- Centrinėje įmonės būstinėje paliekami tik 2 fiziniai serveriai (1 programų ir 1 talpinimo), palaikantys ERP PĮ ir talpinantys jautrius duomenis.
- Atliekamas esamų sistemų atsarginių kopijų kūrimas ir planas kritiniams atvejams.
- Kaip VM serveriai Azure platformoje įdiegiamos 1 VM centrinės būstinės ir 10 VM padaliniams (viso 11 vnt.), jose instaliuojama *Microsoft Datacenter Server OS*. Šis kiekis beveik dvigubai mažesnis už naudotą fizinių serverių skaičių iki šiol, tačiau DK leis greitai ir automatizuotai išplėsti/sumažinti VM kiekį esant poreikiui dėl padidėjusių/sumažėjusių sezoninių ar kitų apkrovų.
- Įmonės SQL duomenų bazės perkeliamos į *SQL Azure Server 2016* (11 vnt.).
- Duomenys talpinami patikimesniuose ir greitesniuose SSD diskuose. Be to, *SQL Server AlwaysOn* parinktis gali užtikrinti aukštą paslaugų tiekimo nepertraukiamumą.
- 11 VM įdiegiama sandėlių procesams apdoroti skirta *ASP.NET* interneto programa. Ši programa migruojama į debesį perkėlimo būdu.
- Paraleliai įvykdytas duomenų iš *SQL* duomenų bazių migravimas, sąryšio gijos (angl. connection strings) programos viduje buvo pakeistos, kad nukreiptų į *SQL Azure* kopiją debesyje.
- Tiek *SQL*, tiek *ASP.NET* perkėlimo atvejais pasiektas *Azure* platformos VM-ų suderinamumas su įmonėje likusia infrastruktūra.
- Srauto tolygiam paskirstymui užtikrinti įdiegiamas ir sureguliuojamas automatinis srauto skirstytojas. Tai leis nesunkiai išplėsti resursus padidėjus naudotojų skaičiui.
- Virtualios mašinos debesyje susiejamos, kad užtikrinti apsaugą nuo atskirų VM gedimo.
- Jungtis su centrinės būstinės sistema užtikrinama VPN tunelio pagalba, o didelės ir netrukdomos spartos ryšiui galės būti naudojamas *Azure Express Route* ryšys.
- Naudotojai *ASP.NET* programą debesyje pasiekia interneto sąsajos pagalba, o ją VM'e paleidžia (įjungia) *Remote Desktop Connection* programos pagalba.
- Programos kodo pritaikymui naujoje platformoje įmonė pasinaudojo *Azure Software Development Kit* (SDK) – specialiu tarpplatforminiu įrankių rinkiniu, palengvinančiu esamo kodo įdiegimą debesyje. Taip pat naudojami *Azure* siūlomi koreguojami šablonai, specialiai skirti sujungti debesyje veikiančioms ir įmonėje likusioms programoms.
- Prie duomenų bazių debesyje, naujoje architektūroje, įdiegtas naujas komponentas - daugia naudotojiškumas, *Azure Web Role* dėka leidžiantis pasiekti didesnę to paties resurso naudojimo efektyvumą.

- Saugumui užtikrinti: a) Įdiegiamas autentifikavimas, b) URL filtravimas (konkreto naudotojo identifikavimui), c) Ryšiui tarp debesųje ir įmonėje esančių sistemų naudojamas HTTPS per SSL arba VPN, d) Kiekvienam padaliniui (naudotojui) debesies aplinkoje įdiegiama priskirtoji duomenų bazė, e) *SQL Database* ugniasienė skiriama apriboti prieigai prie duomenų bazės serverio ar pačių duomenų bazių, kol nėra konkrečiai nustatoma, kuriems kompiuteriams tai leidžiama.

Sistemos valdymas ir optimizavimas.

- Kasdieniam IT resursų debesųje monitoringui *Hosen* būstinėje galės būti naudojamas monitoringo ir rodiklių analizės įrankis *Azure Insight*.
- Parengiamos gairės dėl ERP sistemos būsimo migravimo į *Azure*, greičiausiai *Docker* konteinerių pagalba. ERP sistema bus: 1) *re-architektūrinama*, kas leis ją išskaidyti į komponentus ir mikro paslaugas, 2) įvedami nauji palaikomieji komponentai, 3) įgalinamas tęstinis pristatymas (angl. *continous delivery*) naudojant konteinerizaciją. 4) Įmonė planuoja, kad ERP perkėlimas bus užbaigtas per 12 mėnesių.

Rezultatai.

- Perkelta IT infrastruktūros dalis sumažino TCA ir personalui skirtas išlaidas beveik dvigubai (C priedas). *Capex* (kapitalo) išlaidas pakeitė *Opex* (veiklos) išlaidos.
- IT sistema tapo centralizuota, o jos valdymui ir priežiūrai pakanka 12 IT specialistų (du centriniame būstinėje ir po vieną dešimtyje padalinių). IT infrastruktūrą tapo lengviau stebėti ir optimizuoti iškilus būtinybei ar naujų funkcijų poreikiui.
- Vienoda ir naudotojams draugiška Sandėlių ir užklausų valdymo programos sąsaja leidžia padaliniams sklandžiai įvesti duomenis į sistemą, siųsti dokumentus ir valdyti jiems priskirtas VM iš bet kur, net ir paprasto galinio įrenginio pagalba.
- Iki migracijos 1 serverio įsigijimas ir įdiegimas įmonei vidutiniškai trukdavo apie 4 savaites, tuo tarpu dabar vidutiniškai trunka apie 1 valandą ir gali būti automatizuotas.
- Dėka didesnio automatinio išplečiamumo resursai naujiems naudotojams ar apkrovoms tapo neriboti ir savalaikiai. Padidėjo IT sistemų lankstumas.
- Padidėjo PĮ veikimo veiksmingumas ir nepertraukiamumas, išaugo saugumas ir duomenų privatumas.
- Ataskaitų parengimas SQL duomenų bazėse paspartėjo kelis kartus.
- Naujoji sistema atitinka įmonės identifikuotus apribojimus migracijos pradžioje.

- Pašalinta rizika, kad padalinio viduje veikiančio serverio gedimas ar techninė priežiūra sutrikdytų įmonės (jos padalinio) veiklą.
- Išaiškėjo, kad įmonė nėra pasiruošusi iškart perkelti visas savo sistemas ir duomenis į DK, todėl dabartinis perkėlimas gali neleisti pasiekti galimo viso DK suteikiamo teigiamo efekto.
- Vėliau vyksiantis ERP sistemos perkėlimas į privatų debesį reiškia, kad įmonė savo IT infrastruktūrai naudos *hibridinį debesį* (žr. 22 puslapyje), kuris leis įmonei lanksčiai atsižvelgti į savo veiklos poreikius – esminę PĮ ir jautrius duomenis laikyti privačiame debesyje, o nesudėtingą ir ne tokius konfidencialius duomenis naudojančią PĮ – viešame debesyje.

Išvados

1. Išanalizavus debesų kompiuterijos literatūrą ir kitus šaltinius nustatyta, kad IT infrastruktūros migravimas į debesies aplinką yra nulemtas svarbių skirtingas sritis apimančių iššūkių: techninių (saugumo ir privatumo debesies aplinkoje, paslaugų nepertraukiamo tiekimo, resursų panaudojimo efektyvumo, programos migravimo metodo pasirinkimo, veikimo veiksmingumo, programinės įrangos suderinamumo, vieningo standarto nebuvimo, paslaugų rodiklių matavimo ir kokybės užtikrinimo, perkeliamumo) ir labiau su pačia organizacija bei jos veiklos sritimi susijusių (finansinių, teisinių ir reguliacinių, debesies paslaugų teikėjo pasirinkimo, migravimo trukmės).

2. Darbo rezultatas – atskleista IT infrastruktūros perkėlimo į debesų kompiuteriją faktorių visumos įtaka šiam procesui ir migravimo į IaaS planavimui, o tai savo ruožtu lemia, ar informacinių technologijų resursai po migravimo bus naudojami efektyviai.

3. Atliekant migravimo iššūkių analizę, jie suprantami ne vien tik kaip pavojai, tai – ir naujos galimybės efektyviau valdyti, stebėti, matuoti IT resursus, juos išplėsti ir optimizuoti. Nustatyta, kad iššūkiai:

- gali būti tiek techninio, tiek netechninio pobūdžio, o jų svarba įmonei priklauso nuo pasirinkto migravimo į DK paslaugų modelio, debesies diegimo būdo bei programos adaptavimo debesies aplinkai metodo, organizacijos pobūdžio ir jos dydžio, duomenų saugumo ir privatumo prioritetų, pačios migruojamos programinės įrangos;
- tarpusavyje susiję ir vienas kitą įtakojantys, todėl migravime į DK būtina į juos atsižvelgti kompleksiškai.

4. Nustatyta, kad lyginant su kitais DK paslaugų modeliais, IaaS leidžia turėti lanksčią, standartizuotą ir virtualizuotą aplinką įmonės nuosavų programų diegimui ir suteikia didžiausią IT resursų ir jų efektyvumo kontrolę.

5. Sklandaus migravimo į debesų kompiuteriją planas atsižvelgia į minėtus iššūkius ir nurodo žingsnius jiems įvertinti ir suvaldyti. Šiam planui būdinga iteracija, leidžianti kai kurias plano stadijas ir žingsnius migravimo eigoje pritaikyti konkrečiam migravimui.

6. Migracijos į debesų kompiuteriją proceso tyrimas parodė, kad vienas visiems atvejams migravimo planas neegzistuoja.

7. Išnagrinėjus hipotetinį įmonės IT infrastruktūros perkėlimo į debesį atvejį, išaiškėjo, kad:

- įmonės programinės įrangos ir debesies platformos suderinamumas palengvina migravimą;
- kaskart reikia įvertinti, ar jautrius ir konfidencialius duomenis bei kritinę įmonės veiklai programinę įrangą tikslinga migruoti į debesies platformą;
- debesies tiekėjo platforma, tokia kaip *Azure*, suteikia priemones apsirūpinti reikiamais IT resursais, užtikrinti priimtina sistemų ir ryšio saugumą, atlikti išsamią ir aktualią IT infrastruktūros stebėseną ir valdymą, optimizuoti šią infrastruktūrą, o kartu ir įmonės veiklą bei išteklius;
- įmonei, siekiančiai lanksčiai atsižvelgti į savo veiklos specifiką ir identifikuotus migravimo į debesų kompiuteriją apribojimus, tikslinga apsvarstyti hibridinio debesies naudojimą.

Literatūros ir kitų šaltinių sąrašas

1. Ahmad, A., Babar, M. 2014. A Framework for Architecture-driven Migration of Legacy Systems to Cloud-enabled Software, *Proceedings of the WICSA 2014 Companion Volume* Article No. 7 : 1-8.
2. Andrikopoulos V., *et al.* 2013. How to adapt applications for the Cloud environment. Challenges and solutions in migrating applications to the Cloud, *Computing* 95: 493–535.
3. Andrikopoulos, V., Strauch, S., Leymann, F. 2013. Decision Support for Application Migration to the Cloud: Challenges and Vision. *Institute of Architecture of Application Systems*: 149-155.
4. Armbrust, M., *et al.* A view of cloud computing, *Communications of the ACM*, Vol.53, No.4 (2010): 50-58 [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <<https://cacm.acm.org/magazines/2010/4/81493-a-view-of-cloud-computing/fulltext>>.
5. Amazon Web Services. AWS Total Cost of Ownership (TCO) Calculator. Interneto svetainė [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <<https://awstcocalculator.com>>
6. Bessera P.V., *et al.* 2012. Cloudstep: A Step-by-Step Decision Process to Support Legacy Application Migration to the Cloud, *IEEE*: 1-10.
7. Brebner, P. 2012. Is your Cloud Elastic Enough? Performance Modeling the Elasticity of Infrastructure as a Service (IaaS) Cloud Applications, *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering*: 263-266.
8. Chan, M. 2017. How to calculate the true cost of migrating to the cloud. Networkworld interneto svetainė [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <<http://www.networkworld.com/article/3164444/cloud-computing/how-to-calculate-the-true-cost-of-migrating-to-the-cloud.html>>
9. Chauhan, M., Babar M. 2012. Towards Process Support for Migrating Applications to Cloud Computing, *IT University of Copenhagen*: 1-8.
10. Cloudendure. 2016. 5 Approaches for Migrating Workloads to the Cloud: 1-15 [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <<https://www.cloudendure.com/blog/ebook-5-approaches-migrating-workloads-cloud/>>.
11. Cloudendure. 2016 Cloud Migration Survey. Cloud Migration Challenges and Best Practises: 1-25 interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <<https://www.cloudendure.com/blog/2016-cloud-migration-survey/>>.

12. Conway, G.; Curry, E. Managing Cloud Computing: a Life cycle approach: 1-10 [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <https://www.researchgate.net/publication/220035265_Managing_Cloud_Computing_A_Life_Cycle_Approach>.
13. Clouddorado interneto svetainė. [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <<https://www.clouddorado.com>>
14. Cloud Standards Customer Council. 2013. Migrating Applications to Public Cloud Services: Roadmap for Success: 1-28.
15. Cloud Standards Customer Council. 2014. Practical guide to Cloud computing. Version 2: 1-36.
16. Cripps, P. 2015. Non-Functional Requirements and the Cloud [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <<https://softwarearchitectureen.wordpress.com/2015/04/24/non-functional-requirements-and-the-cloud/>>.
17. ETSI. 2016. ETSI SR 003 391 V2.1.1. Special report. Cloud Standards Coordination Phase 2; Interoperability and Security in Cloud Computing. *Sophia Antipolis*: 1-40.
18. Erl, T., *et al.* 2013. *Cloud computing. Concepts, technology and architecture*. Prentice Hall.
19. Gartner. 2011. Five Options for Migrating Applications to the Cloud: Rehost, Refactor, Revise, Rebuild or Replace, Gartner The Future of IT Conference: 1-28 [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <https://gartnerinfo.com/futureofit2011/MEX38L_A2%20mex38l_a2.pdf>.
20. Gharehbagh, S. 2015. Obstacles and solutions of migration of organizations and outsourcing their it resources into the cloud, *International Journal of Academic Research*, Vol. 7. No. 4. Iss.1.: 1-11.
21. Gholami, M. *et al.* 2016 Cloud migration process—A survey, evaluation framework, and open challenges, *The Journal of Systems and Software* (2016) 31–69.
22. How AWS Pricing Works. 2016. Amazon Web Services [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <<http://aws.amazon.com/whitepapers/>>.
23. IBM Corporation. 2010. Cloud Computing. IBM System Storage Academic Initiative: [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <<http://www.ibm.com/cloud>>.

24. Internap. 2016. Cloud, the Essential Buyer's Guide: 1-12 [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <<https://www.internap.com/resources>>.
25. Interxion. 2015. A Practical guide to cloud onboarding: How to prepare your application workloads: 1-16 [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <http://www.interxion.com/globalassets/_documents/whitepapers-and-pdfs/cloud/WP_CLOUDONBOARDING_en_0715.pdf>.
26. Jafarpour, S.; Yousefi, A. 2016. Security Risks in Cloud Computing: A Review, *International Journal of Current Engineering and Technology*, Vol.6, No.4 (Aug 2016): 1174-1179.
27. Jamshidi, P. et al. 2016. Pattern-based Multi-Cloud Architecture Migration, *Software – Practice and experience*: 1-25.
28. Khajeh-Hosseini, A., et al. 2012. The Cloud Adoption Toolkit: supporting cloud adoption decisions in the enterprise, *Software – Practice and experience*: 447-465.
29. KPMG. 2011. The Cloud. Changing the Business Ecosystem [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <<http://www.tele.net.in/news/item/7154-kpmg-report-on-the-cloud-changing-the-business-ecosystem?format=pdf>>.
30. Kwon, Y., Tilevich, E. 2014. Cloud refactoring: automated transitioning to cloud-based services. *Automated Software Engineering* (2014) 21: 345–372.
31. Loeffler, B. 2011. Cloud Computing: What is Infrastructure as a Service, TechNet Magazine [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <<https://technet.microsoft.com/en-us/library/hh509051.aspx>>.
32. Maenhaut P. et al. 2015. Migrating legacy software to the cloud: approach and verification by means of two medical software use cases, *Software – Practice and experience*: 1-24.
33. Microsoft Azure interneto svetainė [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <<https://azure.microsoft.com>>.
34. Oliveira T., et al. 2014. Assessing the determinants of cloud computing adoption: An analysis of the manufacturing and services sectors, *Information & Management* 51: 497–510.
35. Perucci, D. 2016. Making the Case For and Against Lift-and-Shift Cloud Migration, DZone interneto svetainė [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <<https://dzone.com/articles/making-the-case-for-and-against-lift-and-shift-clo>>.
36. Rong, C. et al. 2013. Beyond lightning: A survey on security challenges in cloud computing, *Computers and Electrical Engineering* 39 (2013): 47–54.

37. Schneider, S., Sunyaev, A. 2015. *CloudLive: a life cycle framework for cloud services*. Electron Markets (2015) 25: 299–311.
38. Serverless Computing. 2017. Wikipedia. Serverless Computing [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <https://en.wikipedia.org/wiki/Serverless_computing>.
39. Somani, G., et al. 2015. *DDoS Attacks in Cloud Computing: Issues, Taxonomy, and Future Directions*. ACM Computing Surveys, Vol. 1, No. 1, Article 1: 1-44.
40. SPEC OSGCloud working group. 2012. Report on Cloud Computing: 1-51.
41. Strauch, S. et al. 2014. Migrating e-Science Applications to the Cloud: Methodology and Evaluation, *Institute of Architecture of Application Systems*: 1-26.
42. Sun, K., Li, Y. 2013. Effort Estimation in Cloud Migration Process, *IEEE Seventh International Symposium on Service-Oriented System Engineering*: 84-91.
43. Talkin Cloud. Cloud Services Brokerage Company List and FAQ [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <<http://talkincloud.com/cloud-services-broker/cloud-services-brokerage-company-list-and-faq>>.
44. Techterms. 2017. Web application definition [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <https://techterms.com/definition/web_application>.
45. Top Threats Working Group. 2016. The Treacherous 12 - Cloud Computing Top Threats in 2016, Cloud Security Alliance: 1-35 [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <<https://cloudsecurityalliance.org/group/top-threats>>
46. Varia, J. 2010. Amazon Web Services - Migrating Your Existing Applications to the AWS Cloud [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <<https://www.aws.com>>.
47. Wang. C. 2016. Containers 101: Linux containers and Docker explained. Infoworld [interaktyvus], [žiūrėta 2017 m. balandžio 15 d.]. Prieiga per internetą: <<http://www.infoworld.com/article/3072929/linux/containers-101-linux-containers-and-docker-explained.html>>.

Priedai

- A priedas.** ETSI (Europos Telekomunikacijų Standartų Instituto) 2016 metų ataskaitos standartai
- B priedas.** IT infrastruktūros migravimo į debesų kompiuteriją plano diagrama
- C priedas.** Preliminarus perkeliama į debesų kompiuteriją IT infrastruktūros 1 metų kainos įmonės viduje ir tapačių paslaugų debesyje palyginimas

A priedas

ETSI (Europos Telekomunikacijų Standartų Instituto) 2016 metų ataskaitos standartai

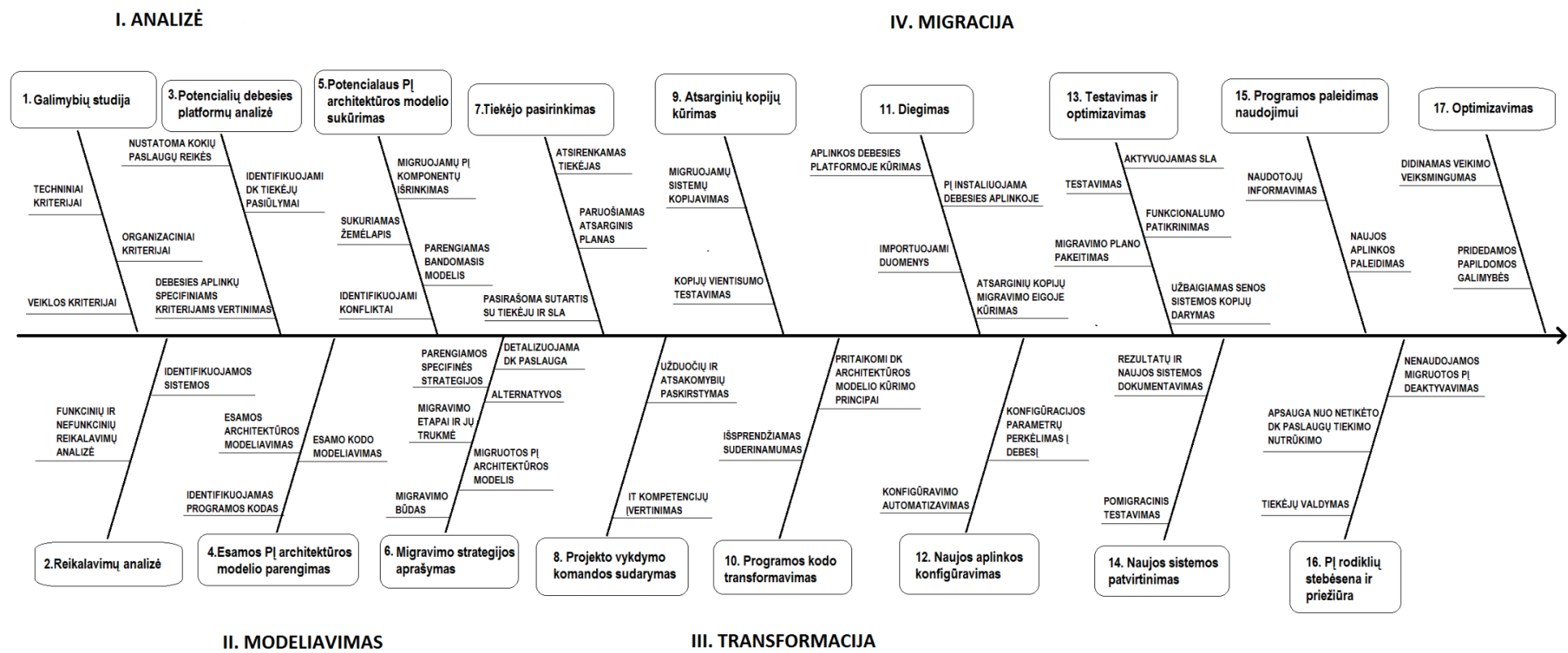
	Debesų kompiuterijos standartai	Bendri standartai (ne specifiniai debesų kompiuterijai)
Perkeliamumas	• Draft ISO/IEC 19041 [i.10]: "Information technology - Cloud computing - Interoperability and Portability". • Draft ISO/IEC 19044 [i.11]: "Information technology - Cloud computing - Data and its flow across devices and cloud services". • ISO/IEC 17203 [i.12]: "Information Technology - Open Virtualization Format Specification". • ISO/IEC 17826 [i.13]: "Information Technology - Cloud Data Management Interface (CDMI)". • ISO/IEC 19099 [i.14]: "Information Technology - Virtualization Management specification". • ISO/IEC 19831 [i.15]: "Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol - An Interface for Managing Cloud Infrastructure". • DMTF DSP0243 [i.16]: "Open Virtualization Format Specification". • DMTF DSP0263 [i.17]: "Cloud Infrastructure Management Interface - CIMI". • OASIS CAMP [i.18]: "Cloud Application Management for Platforms". • OASIS TOSCA [i.19]: "Topology Orchestration Specification for Cloud Applications". • OGF OCCl [i.20] "Open Cloud Computing Interface". • SNIA CDMI [i.21]: "Cloud Data Management Interface".	
Informacijos sauga	• Draft ISO/IEC 27017 [i.32]: "Information technology - Security techniques - Code of practice for information security controls based on ISO/IEC 27002 [i.24] for cloud services". The ISO/IEC 27017 [i.32] - Recommendation ITU-T X.1631 [i.34] standard was approved as an International Standard on the 1st of October 2015 following balloting. The standard however has yet to be formally	• ISO/IEC 27000 [i.22]: "Information technology - Security techniques - Information security management systems - Overview and vocabulary". • ISO/IEC 27001 [i.23]: "Information technology - Security techniques - Information security management systems - Requirements".

	published as a standard and will for this reason remain as 'draft' when referenced in the present document. • Recommendation ITU-T X.1601 [i.33]: "Cloud computing security - Security framework for cloud computing". • Draft Recommendation ITU-T X.1631 [i.34]: "Code of practice for information security controls based on ISO/IEC 27002 [i.24] for cloud services". NOTE: Same as ISO/IEC 27017 [i.32]. • Draft ISO/IEC 19086-4 [i.35]: "Information technology - Cloud computing - SLA framework and terminology - Part 4: Security and Privacy". • CSA CCM [i.36]: V3.0.1, " Cloud Control Matrix". • CSA CTP [i.37]: " Cloud Trust Protocol". • CSA A6 [i.38]: "Cloud Audit". • CSA CAIQ [i.39]: "Consensus Assessments Initiative Questionnaire". • CSA TCI [i.40]: "Reference Architecture - Trusted Cloud Initiative". • Draft NIST SP 500-299 [i.41]: "Cloud computing security reference architecture". • NIST SP 800-125 [i.42]: "Guide to security for full virtualization technologies". • NIST SP 800-144 [i.43]: "Guidelines on security and privacy in public cloud computing".	• ISO/IEC 27002 [i.24]: "Information technology - Security techniques - Code of practice for information security controls". • ISO/IEC 27006 [i.25]: "Information technology - Security techniques - Requirements for bodies providing audit and certification of information security management systems". • ISO/IEC 27014 [i.26]: "Information technology - Security techniques - Governance of information security". • ISO/IEC 27031 [i.27]: " Information technology - Security techniques - Guidelines for information and communication technology readiness for business continuity". • ISO/IEC 24760-1 [i.28]: ""Information technology - Security techniques - A framework for identity management - Part 1: Terminology and concepts". • ISO/IEC 29115 [i.29]: "Information technology - Security techniques - Entity authentication assurance framework". • OpenID [i.30]: see specification at http://openid.net/developers/specs/. • IETF RFC 6749 [i.31]: "The OAuth 2.0 Authorization Framework".
Privatumas	• Draft ISO/IEC 19086-4 [i.35]: "Information technology - Cloud computing - SLA framework and terminology - Part 4: Security and Privacy". • ISO/IEC 27018 [i.46]: "Information technology - Security techniques - Code of practice for PII protection in public clouds acting as PII processors". • CSA PLA [i.47]: "Privacy Level Agreement". • NIST SP 800-144 [i.43]: "Guidelines on security and privacy in public cloud computing".	• ISO/IEC 29100 [i.44]: "Information technology - Security techniques - Privacy framework". • ISO/IEC 29101 [i.45]: "Information technology - Security techniques - Privacy architecture framework".
SLA	• Draft ISO/IEC 19086-1 [i.50]: "Information technology - Cloud computing - SLA framework and terminology - Part 1: Overview and concepts". • Draft ISO/IEC 19086-2 [i.51]: "Information technology - Cloud computing - SLA framework and terminology - Part 2: Metrics". • Draft ISO/IEC 19086-3 [i.52]: "Information technology - Cloud computing - SLA framework and terminology - Part 3: Core requirements". • Draft NIST SP 500-307 [i.53]: "Cloud Computing Service Metrics Description".	• OGF GFD.192 [i.48]: "Web services agreement specification". • Draft OGF GFD.193 [i.49]: "Web services agreement negotiation specification".

	• TMF GB963 [i.54]: "Cloud SLA application note".	
Sertifikavimas	• AICPA SOC 1 [i.55]. • AICPA SOC 2 [i.56]. • AICPA SOC 3 [i.57]. • Certified Cloud Service -TüV Rheinland [i.58]. • CSA Attestation - OCF Level 2 [i.59]. • CSA Certification - OCF Level 2 [i.60]. • CSA Self Assessment - OCF Level 1 [i.61]. • EuroCloud Self Assessment [i.62]. • EuroCloud Star Audit Certification [i.63]. • ISO/IEC 27001 [i.23] Certification. • Payment Card Industry (PCI) Data Security Standard v3 [i.64]. • Leet Security Rating Guide [i.65]. • Cloud Industry Forum Code of Practice [i.66]. • FedRAMP [i.67].	
Kiti standartai ir reglamentavimas	• C-SIG "Code of conduct" (see note 1) [i.77]. NOTE 1: See http://ec.europa.eu/newsroom/dae/document.cfm?doc_id=11194. • ISO/IEC 17788 [i.73]: "Information Technology - Cloud computing - Overview and vocabulary". • ISO/IEC 17789 [i.74]: "Information Technology - Cloud computing - Reference architecture". • Recommendation ITU-T Y.3500 [i.75]: "Information Technology - Cloud computing - Overview and vocabulary". NOTE 2: Same as ISO/IEC 17788 [i.73]. • Recommendation ITU-T Y.3502 [i.76]: "Information Technology - Cloud computing - Reference architecture". NOTE 3: Same as ISO/IEC 17789 [i.74].	• Directive 95/46/EC [i.7]: "EU's Data Protection Directive". • EIF: "European Interoperability Framework" [i.68]. • ISACA COBIT: "Control Objectives for Information and related Technology". • ISO/IEC 20000-1 [i.70]: "Information Technology - Service management system requirements". • ISO 22301 [i.71]: "Societal security - Business Continuity Management Systems - Requirements". • ITIL: "Information Technology Infrastructure Library" [i.72].

B priedas.

IT infrastruktūros migravimo į debesų kompiuteriją planas



C priedas. Preliminarus perkeliamos IT infrastruktūros 1 metų kainos įmonės viduje ir tapačių paslaugų debesyje palyginimas

Azure sistemoje veikimas			Nuosava atitinkama įranga		
Resursas	Formulė	Skaičiavimas	Resursas	Formulė	Skaičiavimas
Azure VM kopija (11 vnt.): Windows Server Datacenter 2016 (4 cores, 28 GB RAM, 56 GB SSD storage, LB). Kaina = 364,53 EUR/mėn.	1 VM kopijos kaina per mėn. X VM kopijų skaičius X 12 mėn.	364,53 EUR X 11 X 12 = 48117,96 EUR	Serveris (22 vnt.) 4 cores, 8 GB RAM Kaina = 1000 EUR	Serverio kaina X Serverių skaičius / 3 metai	1000 EUR X 22 /3 = 7333 EUR
Talpinimas (LRS) (8 TB) Kaina: 1TB = 9,59 EUR /mėn.	1 TB kaina X Duomenų kiekis (TB) X 12 mėn	9,59 EUR X 8 X 12 = 920,46 EUR	Tinklo įrengimas ir serverių konfigūravimas 1-ame padalinyje ar būstinėje Kaina (be PĮ) = 7500 EUR	(Kaina 1 padaliniui X 11) / 3 metai	(7500 EUR X 11) / 3 = 27500 EUR
SQL Database PĮ (11 vnt.) (20 DTU, 250GB storage per DB, 0.0403\$/ val. = 28,09 EUR/mėn. DTU (database transaction units) 20DTU kaina = 25,30 EUR/mėn	20 DTU kaina X SQL Database skaičius X 12 mėn	25,30 EUR X 11 X 12 = 3339,60 EUR	Serverių PĮ Windows Server 2016 Datacenter (visiems serveriams) = 3000 EUR, Microsoft SQL server (talpinimo serveriams) = 3000 EUR.	PĮ kaina 1 serveriui X Serverių skaičius / 3 metai + PĮ kaina 1 serveriui X Serverių skaičius / 3 metai	3000 EUR X 22 / 3 + 3000 EUR X 11/3 = 33000 EUR
IT specialisto darbo vieta Visa kaina = 2600 EUR	1 darbo vietos kaina X IT specialistų skaičius X 12mėn	2600 EUR X 12 X 12 = 374400 EUR	IT specialisto darbo vieta Visa kaina = 2600 EUR	1 darbo vietos kaina X IT specialistų skaičius X 12mėn	2600 EUR X 23 X 12 = 717600 EUR
VPN ryšys tarp On-Premise ir Azure debesies (8 val. per dieną) 0.1603 EUR / val.	VPN įkainis X trukmė per 1 dieną X 365 X 11	0,1603 EUR X 8 X 365 X 11 = 5148,84 EUR	Elektros energijos išteklių Serveris sunaudoja 500 W (0,5 kW/val.) 1kwh = 0,14 EUR	Serverio galia X veikimo val./mėn. X 12 mėn. X 1 kwh kaina X Serverių skaičius	0,5 kWh X 732 X 12 X 0,14 EUR X 22 = 13527,36 EUR
BENDRA SUMA		431.926,86 EUR			798.960,36 EUR
Papildomos parinktys					
App Insights (14,05 EUR /mėn.)					
EXpressRoute ryšys Kaina = 0,1603 EUR /val.					