

Vilniaus universitetas  
Matematikos ir informatikos fakultetas  
Programų sistemų katedra

**Informacinių sistemų integravimo būdų tyrimas ir palyginimas**  
**Research and comparison of information system integration methods**

Magistro baigiamasis darbas

Studentas \_\_\_\_\_ Povilas Milgevičius  
(parašas)

Darbo vadovas \_\_\_\_\_ Dr. Albertas Šermokas  
(parašas)

Recenzentas \_\_\_\_\_ Andrius Adamonis  
(parašas)

## **Santrauka**

Sistemų integracija yra neatsiejama šiuolaikinių informacinių sistemų dalis, kuri leidžia automatizuoti verslo procesus ir sumažinti veiklos sąnaudas. Šiame darbe atliekamas apsikeitimo failais, nuotolinio funkcionalumo, pranešimų integravimo ir bendros duomenų bazės integravimo būdų palyginimas, kai integruojamos sistemos, naudojamos skirtingose įmonėse. Taip pat atliekama semantinių ir sintaksinių lygmenų bei duomenų, pranešimų ir procesų lygių integravimo scenarijų analizė ir vertinimas. Darbe sudaroma metodika, tinkama integravimo būdams ir integravimo scenarijams vertinti, atliekamas jų vertinimas ir pateikiama metodika, leidžianti susieti reikalavimus su charakteristikomis, parenkant integravimo būdą ir integravimo scenarijų.

Raktiniai žodžiai: Sistemų integravimas, integravimo būdai, integravimo scenarijus, saugumas, delsa, kodo sudėtingumas, sistemų sukibimas.

## **Summary**

Allowing business processes to be integrated, system integration reduces costs, making it essential part of current information systems. File transfer, remote procedure call, shared database and messaging integration methods are analysed in this paper. Integration scenarios in semantic and syntactic levels and data, message and process layers are also analysed and evaluated. Methods to evaluate integration methods and integration scenarios are provided and showcased in this paper. Methodology to map requirements to integration method and scenario via characteristics is also provided.

**Keywords:** integration, integration methods, integration scenarios, security, latency, coupling, code complexity.

## TURINYS

ĮVADAS .....	6
1. INTEGRAVIMO BŪDŲ IR SCENARIJŲ ANALIZĖ .....	8
1.1. Integravimo būdo vertinimo charakteristikos .....	8
1.2. Sistemų integravimo scenarijai .....	12
1.2.1. Litan integravimo scenarijai .....	12
1.2.2. Kaneshima integravimo scenarijai.....	13
1.2.3. Izza integravimo scenarijai .....	14
1.3. Integravimo būdai .....	18
1.3.1. Apsikeitimas failais .....	18
1.3.2. Bendra duomenų bazė .....	19
1.3.3. Nuotolinio funkcionalumo iškvietimas .....	22
1.3.4. Pranešimų integravimas.....	23
2. INTEGRAVIMO BŪDŲ IR SCENARIJŲ VERTINIMAS .....	26
2.1. Integravimo būdų vertinimo metodika .....	26
2.2. Integravimo būdų vertinimas .....	28
2.2.1. Apsikeitimas failais .....	28
2.2.2. Bendra duomenų bazė .....	39
2.2.3. Nuotolinio funkcionalumo iškvietimas .....	45
2.2.4. Pranešimų integravimas.....	50
2.3. Integravimo būdų palyginimas .....	57
2.4. Integravimo scenarijų vertinimas.....	59
2.4.1. Vertinimo metodika.....	60
2.4.2. Vertinimas .....	61
3. METODIKOS TAIKYMAS .....	63
3.1. Reikalavimų siejimas su integravimo būdu ir scenarijumi .....	63
3.2. Reikalavimų ir charakteristikų siejimo pavyzdys .....	66
REZULTATAI IR IŠVADOS .....	69

LITERATŪROS SĄRAŠAS.....	71
PRIEDAI .....	75
1 priedas. Integruoto baudžiamojo proceso informacinės sistemos reikalavimai integracijai...	75

## IVADAS

Darbo efektyvinimas ir greitesnis paslaugos klientui suteikimas yra verslo tikslas, dėl kurio kyla poreikis automatizuoti verslo procesus. Kai verslo procesas yra susijęs su keliomis informacinėmis sistemomis, neatsiejama automatizavimo dalimi tampa sistemų integravimas [GY13]. Nepriklausomai nuo įmonės dydžio ar sektoriaus procesų automatizavimas per sistemų integraciją teikia finansinę naudą [Zor11]. Problema, su kuria susiduria organizacijos norėdamos atlikti sistemų integraciją, yra tinkamo integravimo būdo pasirinkimas. Taigi verslui reikalinga metodika integruojamų sistemų integravimo būdui parinkti.

Atliekant sistemų integravimą reikalingi sistemų pakeitimai yra neatsiejama integravimo dalis. Sistemoms gali reikėti pakeisti duomenų modelį, pridėti naują funkcionalumą ar automatizuoti veiklas. Sistemų integracija apima ir integracijos palaikymą bei vystymą po jos sėkmingo paleidimo, sistemos turi tarpusavyje suderinti testavimą ir versijavimą. Taigi integraciją apibūdina ne tik jai keliami reikalavimai, bet ir integravimo scenarijus – pakeitimai, reikalingi sistemose ir jų įgyvendinimo eiga, reikalinga programinė įranga, integravimo objektai (pranešimai, duomenys ar procesai) ir integracijos gyvavimo ciklas.

Integravimo scenarijus ir integravimo būdas nepriklauso vienas nuo kito. Izza [Izz09] išskiria keturias savybes, kuriomis skiriasi integracijos. Šių savybių rinkinys – integravimo scenarijus. Galima vertinti integravimo scenarijaus autonomiškumą, paskirstymą ir lankstumą. Todėl turint integracijos reikalavimus galima nustatyti tinkamiausią integravimo scenarijų, kuris, pasinaudojant sukaupta patirtimi, leidžia paspartinti sistemų integravimą. Integravimo būdų ir integravimo scenarijų derinimas tarpusavyje leidžia maksimizuoti tam tikras charakteristikas (pvz.: sumažinti duomenų delsą, padidinti plečiamumą), todėl reikalinga metodika integravimo būdui ir scenarijui parinkti. Šio darbo tikslas sukurti metodiką integravimo būdui ir scenarijui parinkti remiantis integracijos reikalavimais.

Siekiant palyginti integravimo būdus, reikalingas jų vertinimas. Tam reikia atrinkti vertinimui tinkamas charakteristikas. Šios charakteristikos turi būti matuojamos, jos turi priklausyti nuo integravimo būdo ir turi būti aktualios pasirinktam integravimo būdų vertinimui. Kadangi šiame darbe apsiribojama integracijomis tarp skirtingų organizacijų informacinių sistemų, reikia įvertinti saugumą, sistemų sukibimą, duomenų delsą ir plečiamumą. Norint užtikrinti plečiamumą, apsikeitimas duomenimis turi būti efektyvus, todėl reikia vertinti duomenų nuskaitymo ir įrašymo laiką, šiame darbe vadinamą duomenų transliacija.

Galimi įvairūs informacinių sistemų integravimo būdai. Darbe vertinti pasirinkti G. Hohpe ir B. Woolf [HW00] išskiriami integravimo būdai (kartais vadinamus stiliais). Tai – apsikeitimas failais (angl. *file transfer*), bendros duomenų bazės integravimas (angl. *shared database*), nuotolinio funkcionalumo iškvietimas (angl. *remote procedure call*) ir pranešimų integravimas (angl. *messaging*). Integravimo būdai turi skirtingas savybes, jų taikymas priklauso nuo integruojamoms sistemoms keliamų reikalavimų, kurie siejasi su charakteristikomis, pasirinktomis vertinti.

Atlikus Litan, Kaneshima ir Izza integravimo scenarijų analizę, vertinimui pasirinkti pastarieji. Remiantis S. Izza modeliu yra  $N$  skirtingų integravimo scenarijų:

$$N_{4D} = N_{Lm}N_{Lg}N_S N_P, \quad (1)$$

kur  $N_{Lm}$  – skirtingų lygmenų skaičius,  $N_{Lg}$  – skirtingų lygių skaičius,  $N_S$  – skirtingų sričių skaičius,  $N_P$  – skirtingų perspektyvų skaičius. Izza modelyje  $N_{Lm} = 3$ ,  $N_{Lg} = 4$ ,  $N_S = 2$ ,  $N_P = 3$ .

Įstatę šias vertes į 1 formulę gauname:

$$N = 3 \cdot 4 \cdot 2 \cdot 3 = 72.$$

Darbe nagrinėjama tik dalis Izza minimų scenarijų. Dėl per didelės darbo apimties atsisakyta vertinti iš skirtingų perspektyvų. Be to, apsiribota tik tarpsteminėmis integracijomis ( $N_S = 1$ ). Kadangi aparatinės įrangos ir platformų integracijos nėra taikomos versle, nėra poreikio jų nagrinėti. Lygio dimensijoje apsiribota semantinė ir sintaksinė integracija, todėl  $N_{Lg} = 2$ . Ketvirtoje dimensijoje nagrinėjami visi lygmenys. Atsisakant perspektyvos dimensijos Izza scenarijų erdvė sutraukiama į trimatę ir gaunamas  $N_{3D}$  dydžio scenarijų rinkinys, kuris yra šio darbo tyrimo objektas:

$$N_{3D} = N_{Lm}N_{Lg}N_S = 3 \cdot 2 \cdot 1 = 6.$$

Apibrėžus panaudos atvejus ir parinkus charakteristikas, integravimo būdams ir scenarijams vertinti, atliekamas vertinimas. Kokybinių charakteristikų vertinimui naudojami analitiniai metodai, kiekybinės charakteristikos vertintos laboratorinėje aplinkoje. Atlikus reikalavimų ir charakteristikų siejimą, gauti vertinimo rezultatai panaudoti integravimo būdai ir scenarijui parinkti. Pasirinkus konkrečiam sistemų integravimui keliamus reikalavimus, nustatyti tinkamiausi integravimo būdai ir integravimo scenarijai.

# 1. INTEGRAVIMO BŪDŲ IR SCENARIJŲ ANALIZĖ

## 1.1. Integravimo būdo vertinimo charakteristikos

Sistemų integracijai keliamų reikalavimų įgyvendinimas priklauso nuo tinkamo integravimo būdo pasirinkimo. G. Hohpe ir B. Woolf [HW00] išskiria keturis integravimo būdus (kartais vadinamus stiliais) – apsikeitimą failais (angl. *file transfer*), bendrą duomenų bazę (angl. *shared database*), nuotolinio funkcionalumo iškvietimą (angl. *remote procedure call*) ir pranešimų integravimą (angl. *messaging*). Integravimo būdai susiformavo dėl didėjančio poreikio greičiau perduoti duomenis ir užtikrinti jų vientisumą. Jų taikymas priklauso nuo integruojamoms sistemoms keliamų reikalavimų, todėl norint parinkti integravimo būdą, reikalingas jų palyginimas pagal šiuos reikalavimus. Integravimo būdų palyginti pagal visus galimus reikalavimus neįmanoma, tačiau juos galima lyginti pagal charakteristikas – abstrakčius reikalavimus, gautus juos apibendrinant.

Charakteristikos, kurių neįmanoma įvertinti naudojant formalius metodus ar statistinį vertinimą, negali būti naudojamos integravimo būdams vertinti. G. Hohpe ir B. Woolf [HW00] siūlo vertinti specialistų susipažinimą su integravimo būdu, duomenų formato standartą, asinchroniškumą ir faktą, ar tai yra duomenų ar funkcionalumo integravimas. Šios charakteristikos negali būti naudojamos vertinime. Integravimo būdo pažįstamumas gali būti įvertintas atliekant specialistų interviu, kuriame reikėtų įvertinti ne tik ar integravimo būdas žinomas, tačiau ir pamatuoti to būdo supratimo lygį. Toks tyrimas matuoja specialistų imties suvokimą ir yra sunkiai pritaikomas siekiant parinkti integravimo būdą, nes nematuoja integraciją atliekančių specialistų žinių.

Apsikeitime failais, nuotolinio funkcionalumo iškvietimo ir pranešimų integravime duomenų formatas gali būti parinktas nepriklausomai nuo integravimo būdo. Bendros duomenų bazės integracijoje duomenų formatas priklauso nuo duomenų bazės tipo. Todėl duomenų formato standartas nepriklauso nuo integravimo būdo ir nėra vertinimui tinkama charakteristika. Asinchroniškumas taip pat nepriklauso nuo pasirinkto integravimo būdo. Bendra duomenų bazė ir nuotolinio funkcionalumo iškvietimas veikia sinchroniškai, apsikeitime failais ir pranešimų integravime artimą sinchroniškam veikimui galima pasiekti trumpinant laiką tarp failų (pranešimų) nuskaitymo. Apsikeitimas failais ir pranešimų integravimas veikia asinchroniškai, bendros duomenų bazės ir nuotolinio funkcionalumo iškvietime asinchronišką veikimą galima pasiekti programiškai. Kviečiama funkcija (ar duomenų bazės užklausa) nesiblokuoja ir laukia vykdymui reikalingą laiką. Laukdama rezultato funkciją kviečianti sistema veikia stabiliai, reguliariais intervalais tikrindama, ar dar nėra įvykdytas iškvietas funkcionalumas. Taigi duomenų formato



standartas ir asinchroniškumas nėra vertinimui tinkamos charakteristikos, kadangi nepriklauso nuo integravimo būdo.

Pagal apibrėžimą integravimo objektas priklauso nuo integravimo scenarijaus ir mažai priklauso nuo integravimo būdo. Apsikeitimo failais duomenų integravimas pasiekiamas failuose perduodant struktūrizuotus duomenis, suprantamus integruojančioms sistemoms. Duomenų integravimas nereikalauja apsikeitimo failais automatizavimo, failas iš vienos sistemos į kitą gali būti pernešamas ir fiziškai, tai gali užtrukti neapibrėžtai ilgai. Kai tarp sistemų failais apsikeičiama automatizuotai ir apibrėžtais intervalais, pasiekiamas pranešimų integravimas. Sujungus pranešimus į sekas, galima palaikyti procesą ir taip pasiekti procesų integravimą. Nuotolinio funkcionalumo iškvietime ir pranešimų integravime integravimo objektai yra analogiški apsikeitimui failais. Bendros duomenų bazės integravime pranešimų integravimas negalimas, kadangi sistemos nesikeičia pranešimais. Taigi integravimo objektas nepriklauso nuo integravimo būdo ir nėra vertinimui tinkama charakteristika.

Integracijai keliami reikalavimai nulemia kompiuterinę infrastruktūrą bei programinę įrangą, naudojamą sistemų integravimui. Siekiant padidinti pasiekiamumą, galima įsigyti tai užtikrinančią programinę įrangą; norint padidinti pralaidumą, galima papildomai įsigyti serverių. Tiek programinės įrangos, tiek serverių kaina nepriklauso nuo integravimo būdo. Pasirinktai programinei įrangai galima vertinti technologinį sudėtingumą, kuris apima dokumentacijos detalumą ir suprantamumą bei techninės pagalbos greitį. Taigi integracijos kaina ir technologinis sudėtingumas nėra tinkamos vertinimui charakteristikos, kadangi nepriklauso nuo pasirinkto integravimo būdo.

B. Medjahed ir B. Benatallah [MB14] vertinant integravimo būdus siūlo vertinti sistemų heterogeniškumą. Heterogeniškos sistemos – panašios naudojamomis technologijomis, verslo procesais, programavimo paradigmomis, operacinėmis sistemomis ir kitomis savybėmis. Heterogeniškumas vertina konkrečių sistemų panašumą, o ne integravimo būdą, todėl nėra tinkamas vertinimui. Nuo pasirinkto integravimo būdo priklauso kaip greitai integracijoje galima pridėti naują vienetinę funkciją (pvz.: tekstinę vertę gražinančią funkciją, be įeigos parametrų). Galima vertinti tiek laiką, reikalingą tokią funkciją padaryti prieinamą integruojančioms sistemoms (vadinamą keičiamumu), tiek reaktyvumą, laiką reikalingą integruojančioms sistemoms pradėti naudotis nauju funkcionalumu. Kintamumas ir reaktyvumas yra tinkamos vertinimui charakteristikos. Šiame darbe integravimo būdų vertinime jos prijungtos prie plečiamumo vertinant sistemų sukibimą.

Integravimo būdų vertinimui tinkamos charakteristikos yra programinio kodo sudėtingumas, sistemų sukibimas, saugumas, plečiamumas, duomenų transliacijos laikas ir duomenų delsa. Programinio kodo sudėtingumą galima vertinti naudojant kodo analizės įrankius. Analizuodami programinį kodą, įrankiai matuoja Halstedo (angl. *Halstead*) sudėtingumą. Toks matavimas galimas tik atliekant realių programų analizę. Modelinėms sistemoms, kurios naudojamos integravimo būdų vertinimui šiame darbe, netinka. Modelinėms sistemoms tinka McCabe ciklomatinio sudėtingumo matas, nusakantis kiek yra galimų nepriklausomų programos vykdymo kelių. Sudėtingumas priklauso nuo duomenų formato, naudojamo integracijoje, todėl failų apsikeitime, nuotolinio funkcionalumo iškvietime ir pranešimų integravime ši charakteristika matuojama dažniausiai naudojamam duomenų formatui, nors jis gali būti naudojamas ir kitame integravimo būde.

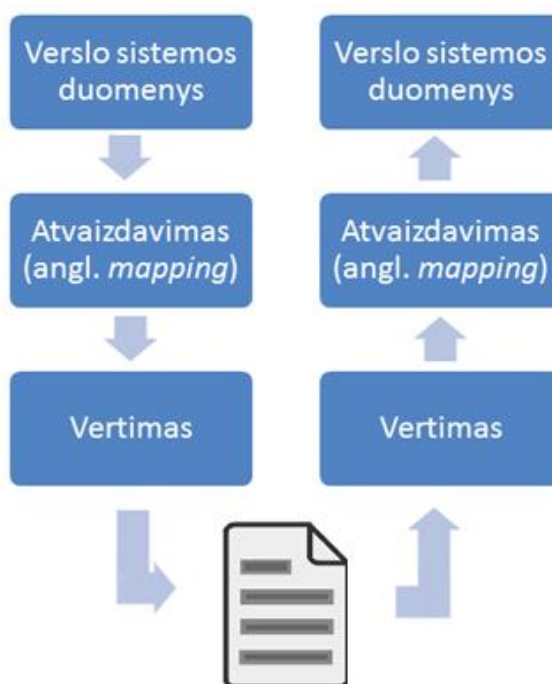
Pautasso[PZL08] sistemų sukibimą dekomponuoja į tris dedamąsias ir vertina, ar kiekviena iš jų yra užtikrinama. Mažas sukibimas tarp sistemų pasiekiamas, kai užtikrinami, bent du iš trijų komponentų. Pautasso sistemų sukibimo komponentai, naudojami integravimo būdams vertinti:

1. **Pasiekiamumas.** Jis užtikrinamas, kai sutrikus vienos sistemos veikimui, kita sistema veikia stabiliai, sutrikimo metu neapdorota informacija (pvz.: neišsiųsti pranešimai, neišsaugoti duomenys), atstačius sistemos veikimą, gali būti apdorota arba pašalinta.
2. **Nekintama sąsaja.** Ji užtikrinama, kai duomenų formatas nekinta ir galimos elementarios funkcijos (pvz.: įrašyti, atnaujinti, ištrinti) ir jų aibė nėra kintamos.
3. **Plečiamumas.** Jis užtikrinamas, kai praplėtus duomenų modelį arba funkcionalumą, be papildomo programavimo jį gali aptikti integruojamos sistemos.

Savola[SA09] darbe sukurta pranešimų programinės įrangos saugumo vertinimo metodika tinka ir integravimo būdams vertinti. Vienas saugumo komponentų Savola darbe yra prieinamumas. Kadangi, remiantis Pautasso, jis priskirtas sistemų sukibimui, saugumo vertinime jis nėra naudojamas. Savola saugumo komponentai, naudojami integravimo būdams vertinti:

1. **Autentifikavimas.** Jis užtikrinamas, kai galima identifikuoti pranešimo šaltinį ar prieigą prie apsaugoto išteklių (pvz.: pranešimo, failo).
2. **Autorizacija.** Ji užtikrinama, kai prieigą prie tam apsaugotų išteklių galima suteikti atskiroms sistemoms, remiantis jų grupėmis ar rolėmis.
3. **Integralumas.** Jis užtikrinamas, kai informacija, kuria apsikeičia sistemos, negali būti paveikta (pvz.: pakeista, ištrinta) trečiųjų šalių.
4. **Konfidencialumas.** Jis užtikrinamas, kai tik autorizuotos sistemos turi prieigą prie resursų.

5. **Neišsigynimas.** Jis užtikrinamas, kai sistemos, dalyvaujanti integracijoje, negali išsižadėti savo veiksmų (pvz.: žinutės išsiuntimo ar gavimo, dokumento įrašymo ar ištrynimo).



1pav. Duomenų transliacijos veiklos

Duomenų transliacijos laikas parodo, kaip greitai apdorojamas failas, kuriuo apsiukeičia sistemos. Jis neįskaito failo perdavimui skirto laiko, todėl matuoja tik nuskaitymo efektyvumą, kai naudojamas konkretus duomenų formatas. Duomenų transliacija apima verslo duomenų atvaizdavimą ir vertimą integruojamoje sistemoje, bei vertimą ir atvaizdavimą integruojančioje sistemoje, kaip pavaizduota 1 pav. Duomenų formatas gali būti pasirinktas nepriklausomai nuo integravimo būdo (pvz.: XML apsiukeitimas failais, JSON nuotolinio funkcionalumo iškvietimas, XML nuotolinio funkcionalumo iškvietimas), todėl reikia vertinti ne tik transliacijos laiką skirtinguose duomenų formatuose, bet ir tų pačių duomenų formatų transliacijos laiką skirtinguose integravimo būduose.

Įskaitant duomenų keliavimą iš vienos sistemos į kitą, gaunama duomenų delsa – laikas kol vienos sistemos duomenys tampa kitos sistemos duomenimis. Duomenų delsa priklauso nuo failų dydžio, tinklo pralaidumo ir nuo papildomų veiklų, kurias turi atlikti sistemos norinčios apsiukeisti informacija (pvz.: autentifikavimas). Mažėjant veiklų skaičiui, duomenų delsos laikas mažėja, tačiau jis niekada negali būti mažesnis už duomenų transliacijos laiką. Tinklo pralaidumas

nepriklauso nuo integravimo būdo, dėl to vertinime jis turi būti eliminuotas. Tai galima pasiekti duomenų delsa normuojant į failo dydį.

Apibendrinant G. Hohpe ir B. Woolf [HW00], Pautasso [PZL08] bei Savola [SA09] siūlomas charakteristikas ir atsižvelgiant į tai, kad šiame darbe apsiribojama integracijomis, kurios atliekamos tarp skirtingų organizacijų sistemų, atrinktos vertinimui tinkamos charakteristikos. Tai – saugumas, sistemų sukibimas, duomenų delsa ir duomenų transliacija ir programinio kodo sudėtingumas. Integravimo objektas, asinchroniškumas, specialistų susipažinimas su integravimo būdu, kaina, technologinis sudėtingumas netinka vertinimui, nes nepriklauso nuo integravimo būdo.

## **1.2. Sistemų integravimo scenarijai**

Nepriklausomai nuo pasirinkto integravimo būdo, sistemų integravimas yra dėsningas. Šį dėsningumą nusako integravimo scenarijus. Integravimo scenarijaus sąvoka nėra apibrėžta programų sistemų inžinerijoje, šiame darbe ji suvokiama kaip apimanti pakeitimus, reikalingus sistemose ir jų įgyvendinimo eigą, reikalingą programinę įrangą, integravimo objektus (duomenis, pranešimus ar procesus) ir integracijos gyvavimo ciklą. Integravimo scenarijus apibendrina sukaupią sistemų integravimo patirtį ir leidžia ją pasinaudoti naujose integracijose.

Nors vadinamas kitaip, integravimo scenarijus yra kitų autorių tyrimo objektas. D. Litan, M. Velicanu ir kt. [LVT+11] išskiria 3 galimus integravimo scenarijus ir juos vadina integracijos lygiais. Kaneshima ir Braga [KB12] integracijos scenarijus – atsakymų, į pačių sudarytą klausymą, kombinacijų aibė. Izza [Izz09] susistemindamas kitų tyrėjų darbus įveda integracijos dimensijų sąvoką, kur integravimo scenarijus yra viena iš dimensijų kombinacijų.

### **1.2.1. Litan integravimo scenarijai**

D. Litan, M. Velicanu ir kt. straipsnyje [LVT+11] išskiriami trys integracijos lygiai: duomenų, paslaugų ir procesų. Integracijos sudėtingumas didėja, pereinant nuo duomenų integracijos prie procesų integracijos, todėl procesų integracija yra ilgesnė ir brangesnė lyginant ją su duomenų ir paslaugų. Atliekant sistemų integraciją reikia nuolat vertinti, ar tenkinami būsimų integracijos vartotojų poreikiai. Tik patenkinus vartotojų poreikius, integracijos užsakovas ją vertins kaip sėkmingą. Tad integracijos biudžetas nulemia integravimo lygį, o integracijos sėkmė priklauso nuo vartotojų poreikių tenkinimo.

Autoriai pastebi koreliaciją tarp integracijos sėkmės (pagal užsakovo vertinimą) ir išankstinio pasiruošimo integracijai. Dėl integracijos gali tekti atlikti pakeitimus pačiame versle – gali būti keičiamos paslaugos, vartotojų veiksmai ir verslo procesai. Pakeitimai gali būti plataus

mašto; sudėtingiausi pakeitimai apima verslo procesų automatizavimą siekiant pagerinti pačios informacinės sistemos efektyvumą ir sumažinti veiklos sąnaudas. Norint automatizuoti verslo procesus, anot Litan, reikia atlikti šiuos žingsnius:

1. Praplėsti verslo funkcionalumą, siekiant tam tikro abstrakcijos lygio, kad būtų galima sujungti skirtingų sistemų komponentus.
2. Panaudoti informacinių technologijų žinių nereikalaujančius ir lengvai suprantamus įrankius verslo procesų automatizavimui.
3. Panaudoti verslo procesų valdymo įrankius, leidžiančius susieti visus verslo procesus.

Anot straipsnio, paslaugų ir procesų integravimas yra pranašesnis už duomenų integravimą. Paslaugų ir procesų integracija garantuoja mažą sistemų sukibimą, procesų autonomiškumą, veikimą be būsenos (angl. *statelessness*) ir pakartotinį panaudojamumą. Autoriai išskiria du būdus atlikti paslaugų integravimą. Tai paslaugų architektūra (angl. *Service-oriented architecture*; toliau – *SOA*) ir tarnybų magistralė (angl. *Enterprise Service Bus*; toliau – *ESB*). Paslaugų architektūra leidžia apsieisti duomenimis kviečiant pavienes funkcijas, o jų apjungimas panaudojant orkestravimą leidžia integruoti verslo procesus. Tarnybų magistralė integruojant sistemas sumažina jungčių tarp atskirų sistemų skaičių, suteikdama bendrą, centralizuotą integravimo sąsają.

Litan integravimo scenarijų privalumas – pateikiami žingsniai sistemų integravimui, tačiau autorių siūlomas integravimo scenarijų rinkinys yra per daug abstraktus ir neatskleidžia visų integracijų skirtumų. Litan neatsižvelgia į tai, ar sistemos priklauso tai pačiai organizacijai ir kas yra integracijos vartotojas (verslas ar informacinių technologijų specialistai). Maža dalis integracijai keliamų reikalavimų nulemia integravimo objektus ir integravimo scenarijų. Litan scenarijai gali būti naudojami, kai mažai žinoma apie sistemas, arba žinomas tik integracijai skiriamas biudžetas. Taigi šis integravimo scenarijų rinkinys nėra tinkamas vertinimui, nes jis yra per mažai granuliarus.

### **1.2.2. Kaneshima integravimo scenarijai**

Konkretesnį nei Litan sistemų integravimo scenarijų rinkinį siūlo Kaneshima ir Braga [KB12]. Autoriai išskiria penkias savybes, kuriomis skiriasi integravimo scenarijai. Tai informacijos prieinamumas, greitaveikos svarba, sinchronizacijos svarba, galimybė ne tik skaityti, bet ir rašyti duomenis ir lokalias informacijas (nematomos vienai iš integruojamų sistemų) egzistavimas. Identifikuojant, ar kiekviena iš šių savybių yra svarbi integruojant sistemas (atsakant „ Taip“ arba „ Ne“) nusakomas integravimo scenarijus.

Kiekvieną iš scenarijų Kaneshima ir Braga atliko sieja (angl. *map*) su jų siūlomais integravimo būdais:

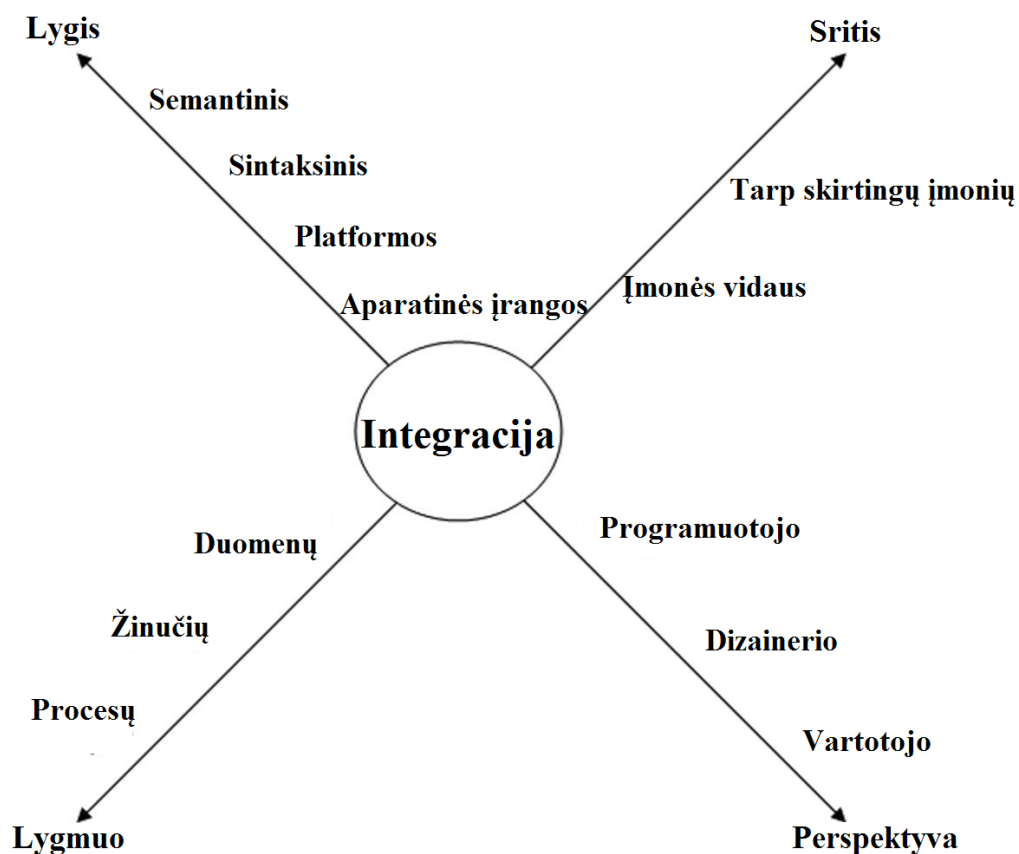
- Tiesioginė prieiga prie duomenų bazės.
- Tiesioginė prieiga prie duomenų bazės per žiniatinklio paslaugas (angl. *web-services*).
- Prieiga prie duomenų bazės per tarpinę sistemą.
- Prieiga prie duomenų bazės per tarpinę sistemą naudojant žiniatinklio paslaugas.
- Prieiga prie duomenų bazės per tarpinę sistemą naudojant žiniatinklio paslaugas ir priežiūrą.

Tarpinė sistema šiuose integravimo būduose yra atsakinga už duomenų kopijavimą iš integruojamos sistemos duomenų bazės ir jų konvertavimą integruojančioms sistemoms suprantamą formatą.

Kaneshima ir Braga integravimo scenarijų privalumas yra tai, kad integracijai keliami reikalavimai lengvai siejami su autorių išskirtomis integracijos savybėmis. Identifikavus reikšmingas savybes, nusakomas integravimo scenarijus, kuris iš karto nusako ir integravimo būdą. Visgi autorių siūlomas integravimo scenarijų rinkinys yra nulemtas mažo savybių skaičiaus. Dėl to tik dalis integracijai keliamų reikalavimų nulemia integravimo scenarijų, o tuo pačiu ir integravimo būdą. Be to, integravimo scenarijus susietas 1-1 ryšiu su integravimo būdu. Tai nėra teisinga, nes integravimo scenarijus gali būti siejamas su daugiau nei vienu integravimo būdu.

### **1.2.3. Izza integravimo scenarijai**

Atsižvelgdamas į kitų autorių darbus, S. Izza [Izz09] susistemino didelį kiekį integracijų ir išskyrė keturias savybes, kuriomis jos skiriasi, pavadindamas jas integravimo dimensijomis. Tai lygmuo (angl. *layer*), lygis (angl. *level*), sritis (angl. *scope*) ir perspektyva (angl. *perspective*), kaip pavaizduota 2 pav. Šių savybių rinkinys, arba taškas keturmatėje integracijų savybių erdvėje, yra tai, kas šiame darbe vadinama integravimo scenarijumi.



2 pav. Integracijos dimensijos

Pagal sritį integracija gali būti vidinė, integruojant sistemas, naudojamas vienoje organizacijoje, arba išorinė, atliekama tarp skirtingų organizacijų. Kai įmonė naudoja daugiau nei vieną informacinę sistemą, verslo procesai gali apimti skirtingas informacines sistemas įmonės viduje. Siekiant užtikrinti duomenų vientisumą tarp šių sistemų, atliekama jų integracija. Vidinė integracija taip pat gali būti atliekama siekiant sujungti skirtingus organizacijos lygmenis. Vidinėje integracijoje sistemos naudoja organizacijos nustatytą duomenų formatą, todėl jo nereikia derinti tarp sistemų. Šis formatas nebūtinai turi būti standartizuotas, tačiau patvirtinto standarto (pvz.: XML, JSON) naudojimas palengvina vidinės integracijos sąsajos pakartotinį panaudojamumą išorinei integracijai. Taigi vidinė sistemų integracija užtikrina duomenų vientisumą tarp skirtingų informacinių sistemų, naudojamų organizacijoje.

Išorinė sistemų integracija yra sudėtingesnė lyginant su vidine. Išorinėje integracijoje standartizuotas duomenų formatas yra būtinas. Tai gali būti patvirtintas standartas ar tarp įmonių suderintas duomenų formatas. Tarp įmonių suderintas formatas apsunkina galimybę integracijos sąsają panaudoti kitoms sistemoms. Priimant, kad organizacijos intranetas yra saugus, vidinė integracija nereikalauja papildomų priemonių siekiant užtikrinti integracijos saugumą. Integruojant skirtingų organizacijų sistemas privalu užtikrinti autentifikavimą, autorizaciją,

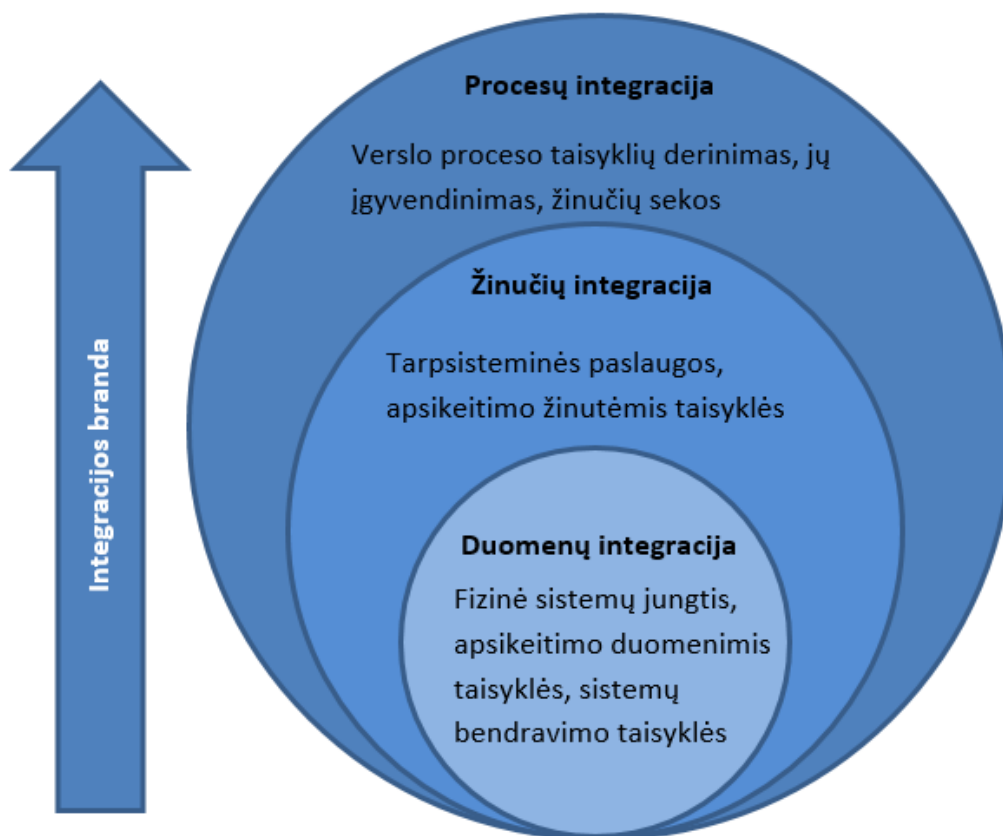
integralumą ir konfidencialumą. Taigi išorinis integravimas yra sudėtingesnis jį lyginant su vidiniu.

Sistemų integracija siekiama pagerinti verslo procesus, todėl ją atliekant privalu konsultuotis su susijusiais asmenimis. S. Izza [Izz09] išskiria tris susijusių asmenų tipus – vartotoją, dizainerį ir programuotoją. Iš vartotojo perspektyvos vertinami skirtumai vartotojų veiklos procesuose, naudojantis integruojamomis sistemomis. Vartotojo perspektyvos integracija siekiant sujungti skirtingų vartotojų veiklas į vieną procesą. Iš dizainerio perspektyvos vertinami integruojamų sistemų duomenų modelio skirtumai, siekiant juos sumažinti ar panaikinti. Iš programuotojo perspektyvos vertinami programinės įrangos ir programavimo paradigmu skirtumai, siekiant vieningo informacinių sistemų veikimo. Tarp sistemų esantys skirtumai gali būti mažinami keičiant verslo procesus, darbuotojų veiklas, integruojamų sistemų dizainą ar naudojamą programinę įrangą.

Sistemos gali būti integruojamos skirtingais lygmenimis. Linthicum [Lin04] išskiria du integravimo lygmenis: duomenų modelio ir verslo modelio. Verslo modelio lygmuo toliau skaidomas į programos lygmenį, metodo lygmenį ir vartotojo grafinės sąsajos lygmenį. Stonebraker [Sto99] taip pat išskiria du integravimo lygmenis: duomenų ir įvykių. Lublinsky [LT03], kaip ir Litan[LVT+11] išskiria tris integravimo lygmenis – duomenų, žinučių ir procesų. Visų autorių išskiriami lygmenys yra panašūs. Jiems bendras duomenų lygmuo; Linthicum programos ir vartotojo grafinės sąsajos lygmuo atitinka Lublinsky žinučių lygmenį, o metodo lygmuo atitinka procesų lygmenį; Stonebraker įvykių lygmuo gali būti išskaidytas į Lublinsky žinučių ir procesų lygmenis. Lublinsky aprašyti integracijos lygmenys, kurį taip pat naudoja ir Izza, apima kitų autorių aprašomus lygmenis. Taigi sistemų integracija galima žinučių, procesų ir duomenų integravimo lygmenimis.

Kiekvienas integravimo lygmuo reikalauja skirtingų procesų vykdymo. Atliekant duomenų integraciją galima atlikti fizinę sistemų jungtį (pvz.: per bendrą duomenų bazę) arba fiksuoti apsiikeičiamų duomenų modelį ir nustatyti apsiikeitimo duomenimis taisykles. Duomenų lygmens integracija galima, kai apsiikeičiamų duomenų modelis yra nekintantis. Norint lankstesnio duomenų modelio, atliekama pranešimų integracija, kuri remiasi paslaugomis (pvz.: kviečiant funkcijas per žiniatinklio paslaugas). Apibrėžus pranešimų apsiikeitimo procesą, verslo taisykles ir nustatčius žinučių vykdymo sekas, integracija gali palaikyti verslo procesą, taip pranešimų integracija tampa procesų integracija. Pereidama nuo duomenų prie procesų lygmens integracija geba palaikyti sudėtingesnius verslo procesus, todėl didėja integracijos branda (kaip pavaizduota 3 pav.).





3 pav. Procesai integravimo lygmenyse

Integruojamos sistemos gali veikti skirtingose platformose, būti parašytos skirtingomis programavimo kalbomis, naudoti skirtingas duomenų struktūras. Šie skirtumai gali būti suskirstyti į keturis lygius. Tai aparatinės įrangos – kompiuterinės įrangos, tinklų ir kt.; platformos – operacinės sistemos, duomenų bazių platforma ir kt.; sintaksinis – kaip aprašomas duomenų modelis ir operacijos ir semantinis – duomenų schemas ir operacijų reikšmė. Norint atlikti integraciją kiekviename iš aukštesnių lygių, privaloma užtikrinti, kad yra integruojami ir žemesni lygiai. Aparatinės įrangos ir platformų integracija verslui neteikia naudos, todėl nėra atliekama. Atliekant informacinių sistemų integraciją integruojama semantiniu arba sintaksiniu lygmeniu.

Izza pasiūlytas integravimo scenarijų rinkinys yra bendresnis lyginant jį su Litan ir Kaneshima rinkiniais. Jeigu scenarijus sutrauktume iki lygmens dimensijos, gautume Litan scenarijų rinkinį, jei nagrinėtume sintaksinę duomenų integraciją iš programuotojo perspektyvos, gautume Kaneshima scenarijus. Izza scenarijų rinkinys apima didelį kiekį integracijos savybių, todėl leidžia lanksčiai parinkti integravimo scenarijų pagal integracijos reikalavimus. Be to, Izza scenarijus galima nagrinėti tik pagal vieną dimensijų ar dimensijų kombinaciją, pvz.: atsižvelgiant nuo perspektyvos nagrinėjami ir lyginami tik duomenų, žinučių ir procesų lygmenys, semantiniame ir sintaksiniame lygyje, vidinėse ir išorinėse integracijose. Taigi Izza pasiūlytas integravimo scenarijų rinkinys apibendrina kitų siūlomus rinkinius.

### 1.3. Integravimo būdai

Informacinių sistemų integravimo problemoms spręsti yra nusistovėję skirtingi informacinių sistemų integravimo būdai – apsikeitimas failais, bendra duomenų bazė, nuotolinio funkcionalumo iškvietimas ir pranešimų integravimas. Nėra vieno geriausio integravimo būdo, kiekvienas jų turi skirtingas savybes, ir gali būti parenkamas priklausomai nuo integracijai keliamų reikalavimų. Kiekvienas integravimo būdas lygio dimensijoje gali būti derinamas tiek su semantiniu, tiek su sintaksiniu integravimo scenarijumi. Integravimo būdas nepriklauso ir nuo integravimo scenarijaus lygmens, tačiau duomenų integravimui dažniau naudojamas apsikeitimas failais ir bendra duomenų bazė, pranešimų ir procesų integracijai – pranešimų integravimas ir nuotolinio funkcionalumo iškvietimas.

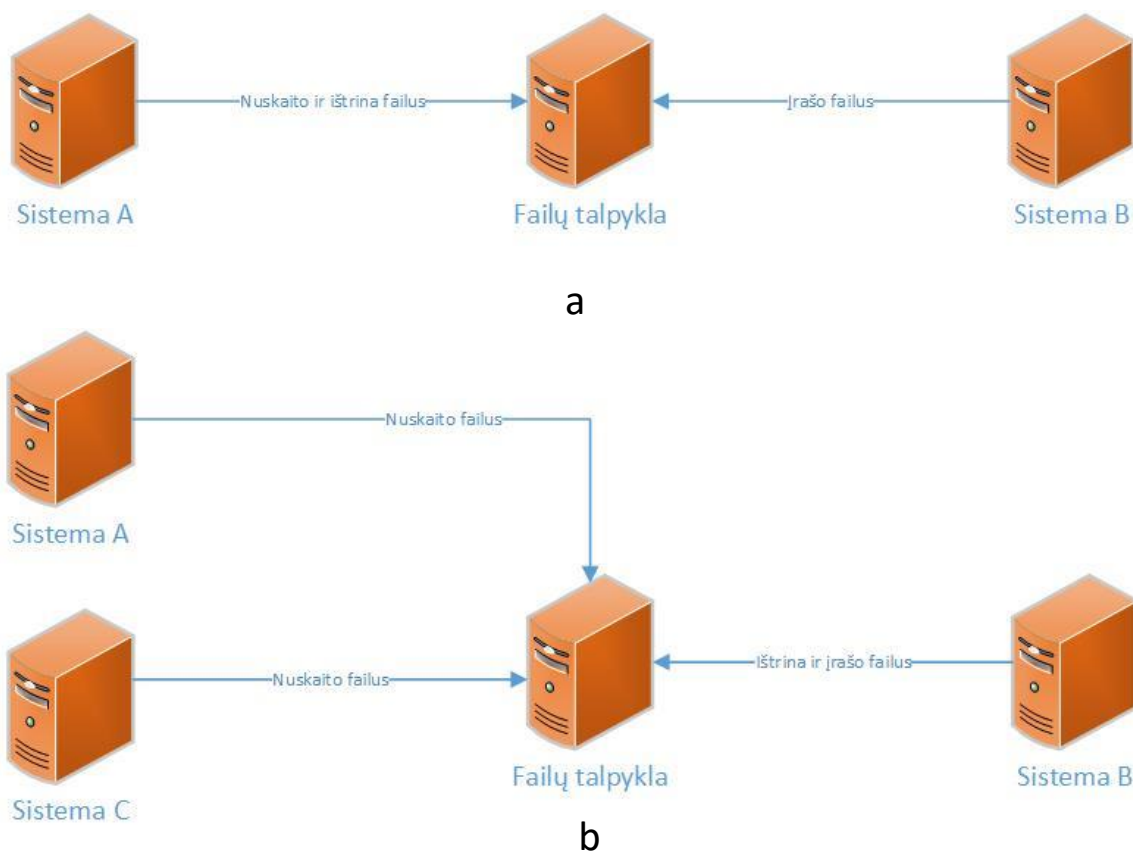
#### 1.3.1. Apsikeitimas failais

Sistemų integravimo galimybės atliekant jį su nebevystomomis sistemomis yra ribotos. Atliekant tokią integraciją integruojami nebepalaikomos sistemos duomenys į vystomą sistemą. Paprasčiausias būdas įgyvendinti tokią integraciją – duomenų eksportas iš vienos sistemos ir jų importas į kitą sistemą. Kadangi integruojamoje sistemoje išeities failo struktūra nekinta, tarp integruojamų sistemų sukuriama mažai sukibusi sąsaja apsikeičiant failu. Kai integruojama sistema turi galimybę eksportuoti duomenis, joje nereikalingi jokie pakeitimai norint atlikti apsikeitimo failais integravimą. Pakeitimai reikalingi tik integruojančioje sistemoje, aprašant duomenų nuskaitymo programą. Siekiant automatizuoti apsikeitimo failais procesą, integravimo būdą galima realizuoti kaip pavaizduota 4 pav. Tokioje integracijoje sistemų veikimas yra nepriklausomas, todėl jos yra mažai sukibusios.

Saugus apsikeitimas failais užtikrinamas įgyvendinant 1.1 poskyryje minimus saugumo komponentus. Siekiant užtikrinti apsikeičiančių šalių autentiškumą galima pasinaudoti Axelsson[Axe10] metodu. Šis metodas negarantuoja konfidencialumo, todėl gali būti taikomas tik intranete. Kai failais apsikeičiama internetu ar bet koku kitu būdu, galima taikyti Keech [Kee02] metodiką. Ji užtikrina failo autentiškumą ir leidžia autorizuoti failo siuntėją. Apsikeitimas failais yra tiek saugus, kiek saugios metodikos taikomos realizuojant šį integravimo būdą.

Apsikeitimui failais būdinga didelė duomenų delsa. Trumpinant intervalą tarp duomenų eksporto ir importo, duomenų delsa galima sumažinti, tačiau tai sistemoms sukuria papildomą apkrovą. Be to, duomenų delsa priklauso nuo duomenų transliacijos laiko. Kai duomenų, naudojamų apsikeitime, formatas nėra standartinis, jo nuskaitymo sparta priklauso nuo to, kiek efektyvus yra nuskaitymo algoritmas. Kai nuskaitymo algoritmas yra neefektyvus, padidėja

duomenų transliacijos laikas, o tuo pačiu ir duomenų delsa. Taigi duomenų apsikeitimo intervalas ir duomenų transliacijos laikas apsikeitimo failais nulemia duomenų delsa.

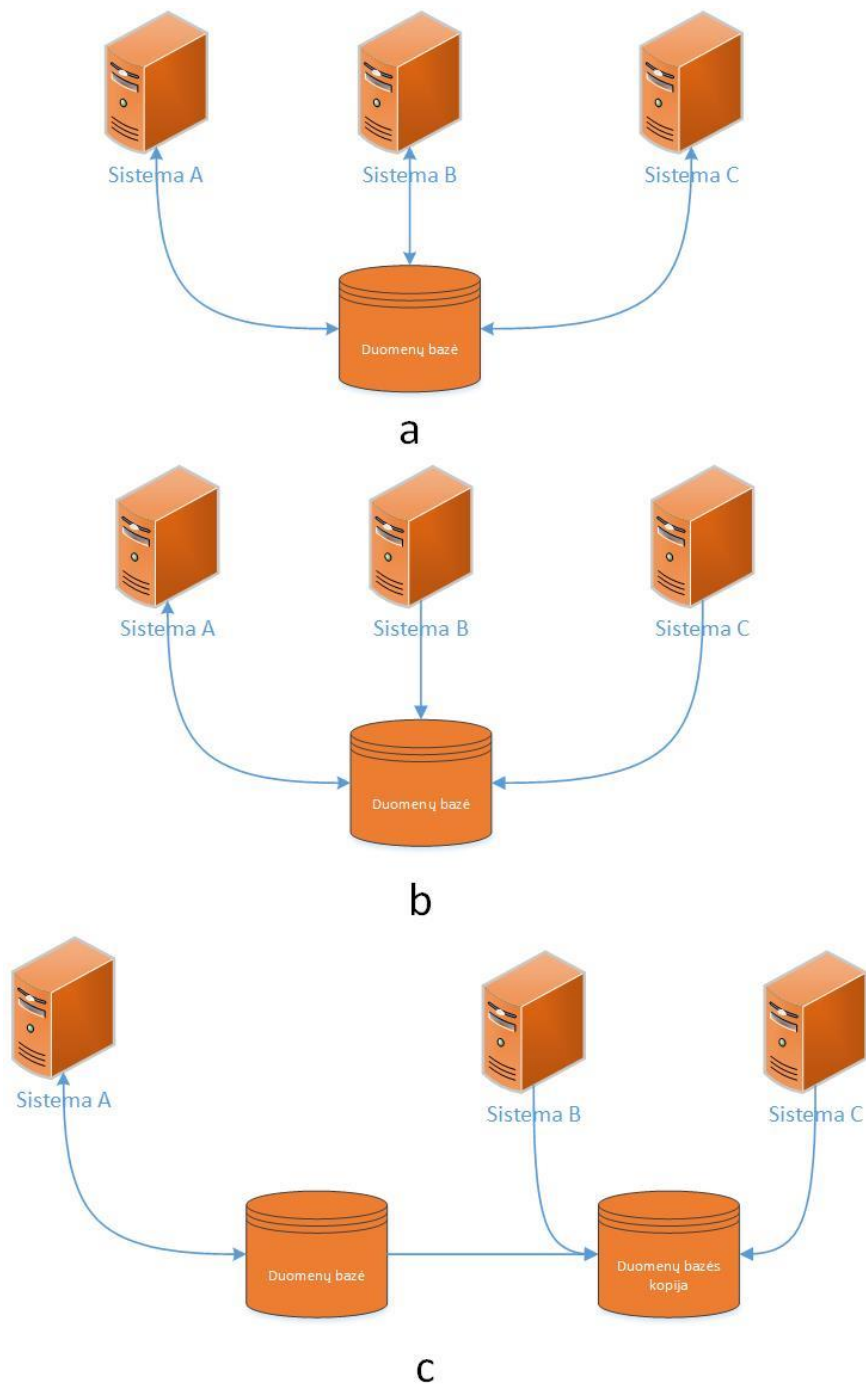


4 pav. Apsikeitimo failais integravimo būdo principinė schema: a – kai nuskaitančioji sistema atsakinga už failo pašalinimą; b – kai įrašančioji sistema atsakinga už failo pašalinimą

Taigi apsikeitimą failais galima taikyti tada, kai viena iš sistemų nėra vystoma ir turi galimybę eksportuoti duomenis. Šis integravimo būdas gali būti taikomas ir kitais atvejais, naudojant standartinį duomenų formatą ir priimant didelę duomenų delsa. Siekiant mažesnės duomenų delsos, sistemoms su panašiu duomenų modeliu, galima taikyti bendros duomenų bazės integravimą. Kitais atvejais, galima taikyti pranešimų arba nuotolinio funkcionalumo iškvietimo integravimą.

### 1.3.2. Bendra duomenų bazė

Siekiant mažos duomenų delsos, kelias panašias tos pačios organizacijos viduje naudojamas informacines sistemas galima integruoti naudojant bendrą duomenų bazę, kaip pavaizduota 5a pav. Siekiant išvengti nesutapimų tarp integruojamų sistemų, viena sistema gali turėti įrašymo teisę, o kitos sistemos tik skaitymo teises, kaip pavaizduota 5b pav. Kai didelė duomenų delsa priimtina ir norima užtikrinti sistemų greitį, integraciją galima realizuoti kaip pavaizduota 5c pav.



5 pav. Bendros duomenų bazės integravimo būdo principinė schema: a – kai visos integruojamos sistemos turi tiek rašymo, tiek skaitymo teises; b – kai tik viena sistema turi rašymo ir skaitymo teises, o kitos sistemos tik skaitymo teises; c – kai viena sistema turi rašymo ir skaitymo teises, o kitos tik skaitymo teises iš duomenų bazės kopijos

Bendros duomenų bazės integravime, priešingai nei apsikeitime failais, tarp sistemų nereikia derinti duomenų formato, kadangi integruojamos sistemos naudoja tą pačią duomenų bazę, tą patį duomenų formatą ir duomenų modelį. Kadangi duomenų bazė yra kiekvienos sistemos dalis, šis integravimo būdas nereikalauja papildomos programinės ar aparatinės įrangos. Bendros duomenų

bazės saugumas priklauso nuo to, kiek apsaugota pati duomenų bazė. Duomenų bazės prieigai privaloma autentifikuotis, resursai reikalauja autorizacijos. Konfidencialumui užtikrinti galima taikyti šifravimą, o duomenis dešifruoti atskirose sistemose. Taigi bendra duomenų bazė yra tiek saugus integravimo būdas, kiek užtikrintas duomenų bazės saugumas.

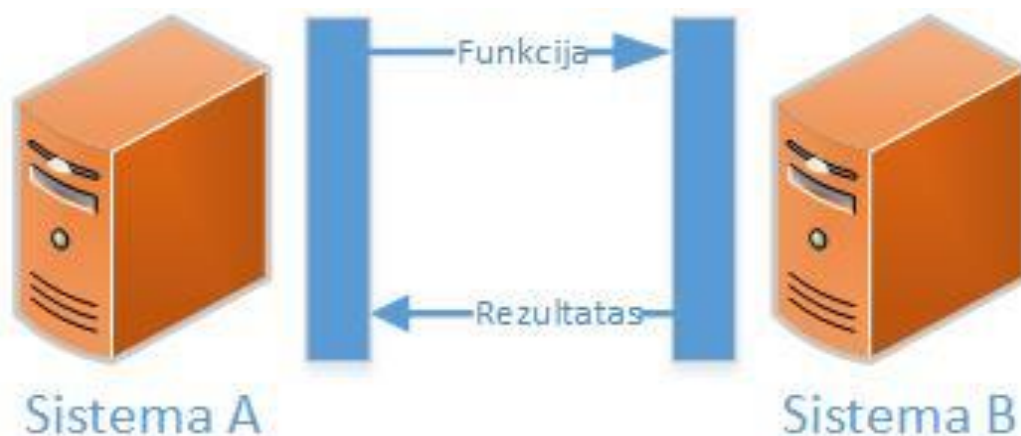
Bendroje duomenų bazėje galima duomenų vientisumo problema. Kai sistemos integruojamos 5a pav. pavaizduotu būdu, tuos pačius duomenis vienu metu gali keisti kelios sistemos. Dėl šios priežasties galima situacija, kai nėra aišku, kurios sistemos duomenys yra teisingi ir įrašytini į duomenų bazę. Ši problema gali būti sprendžiama išskiriant kiekvieno duomenų vieneto vadovą (angl. *master*), kurio įrašytini duomenys laikomi teisingais. Be to, kol duomenys redaguojami vienoje sistemoje, jų redagavimas kitoje sistemoje gali būti ribojamas. Dėl didelės apkrovos bendra duomenų bazė gali tapti kamščiu (angl. *bottleneck*) jį naudojančioms sistemoms. Kai tik viena sistema turi įrašymo teises, bendros duomenų bazės integracija realizuojama kaip pavaizduota 5b pav. Norint sumažinti duomenų bazės apkrovą ir toleruojama atsirandanti duomenų delsa, integruojančios sistemos gali naudoti duomenų bazės kopiją, kaip pavaizduota 5c pav. Taigi bendros duomenų bazės greitaveika gali nukentėti siekiant sumažinti duomenų delną.

Bendros duomenų bazės integracija skirtingų organizacijų informacinėse sistemose sunkiai įmanoma. Vienos įmonės viduje integruojant šiuo būdu duomenų schema tampa sudėtinga. Be jau egzistuojančio bendro duomenų rinkinio integruojamos sistemos turi tik joje naudojamus duomenis. Dėl šios priežasties duomenų schema tampa sudėtingesnė didėjant integruojamų sistemų skaičiui. Tai labai apsunkina specialistų (programuotojų, duomenų bazės administratorių ir kitų) darbą ir didina integracijos palaikymo kaštus. Išorinės integracijos atveju, duomenų schema gali tapti labai sudėtinga ir sunkiai suprantama. Dėl šios priežasties bendros bazės integravimo būdas yra naudojamas, kai integruojamos kelios panašios sistemos, naudojamos toje pačioje organizacijoje.

Taigi bendros duomenų bazės integravimo būdas garantuoja nekintantį duomenų formatą, yra paprastai realizuojamas, lengvai keičiamas ir saugus integravimo būdas. Pagrindinis šio integravimo būdo privalumas – maža duomenų delsa. Atliekant integravimą naudojant bendrą duomenų bazę gali nukentėti sistemų greitaveika, o didėjant integruojamų sistemų skaičiui sudėtingėja duomenų schema ir jos skaitomumas, didėja tikimybė akliavietei ir tikimybė prarasti duomenų vientisumą.

### 1.3.3. Nuotolinio funkcionalumo iškvietimas

Kai sistemų integravimo tikslas palaikyti procesą, bendros duomenų bazės ir apsikeitimo failais integravimo būdai nebetinka. Tokiu atveju atliekamas pranešimų arba nuotolinio funkcionalumo iškvietimo integravimas. Nuotolinio funkcionalumo iškvietimas sinchroniškas integravimo būdas, leidžiantis per integruojamos sistemos paskelbtą sąsają iškviešti norimą funkcionalumą. Nuotolinio funkcionalumo iškvietimo principinė schema pavaizduota 6 pav.



6 pav. Nuotolinio funkcionalumo iškvietimo integravimo būdo principinė schema

Nuotolinio funkcionalumo iškvietime vidinė integruojamos sistemos logika lieka uždara (angl. *encapsulated*), todėl integruojama sistema gali atlikti pakeitimus, nuo kurių nepriklauso kitos sistemos veikimas. Be to, verslo logikos paslėpimas leidžia apsaugoti nuo galimo verslo paslapčių atskleidimo. Integraciją lengva plėsti, kadangi funkcijos įeigos parametrų struktūra gali būti pildoma, nepriklausomai nuo jau egzistuojančių integracijų. Šis integravimo būdas, kaip ir bendros duomenų bazės integravimo būdas užtikrina mažą duomenų delną, be to, jame užtikrinamas duomenų vientisumas.

Nuotolinio funkcionalumo iškvietimas sukuria didelį sukibimą tarp sistemų. Užsiblokuojantis funkcionalumo iškvietimas gali sukelti greitaveikos problemų sistemoms, jei jis yra prieinamas nepatyrusiems sistemų vartotojams. Be to, šis integravimo būdas reikalauja papildomų techninių žinių – susipažinimo su integracijai reikalinga programine įranga, gero naudojamų protokolų ir standartų supratimo.

Be minėtų trūkumų Kukreja [KG14] pastebi, kad šis integravimo būdas nesuteikia galimybės iškviešti to paties funkcionalumo daugiau negu vienoje sistemoje vienu iškvietimu. Perduodant parametrus tarp sistemų negalima perduoti rodyklių, kadangi objektai, į kuriuos rodoma, nebus prieinami kitoje sistemoje, be to, negalima naudoti globalių kintamųjų. Klientui pareikalavus

didelio kiekio duomenų ar bet kokio kito ilgai trunkančio proceso atlikimo, iškviečianti sistema blokuojasi. Dėl to gali nukentėti iškviečiančios sistemos greitimeika ir stabilumas. Taigi šis integravimo būdas sukuria didelį sukibimą tarp integruojamų sistemų, reikalauja papildomo specialistų pasirengimo, riboja globalių kintamųjų naudojimą ir neleidžia kreiptis į kelias sistemas vienu metu.

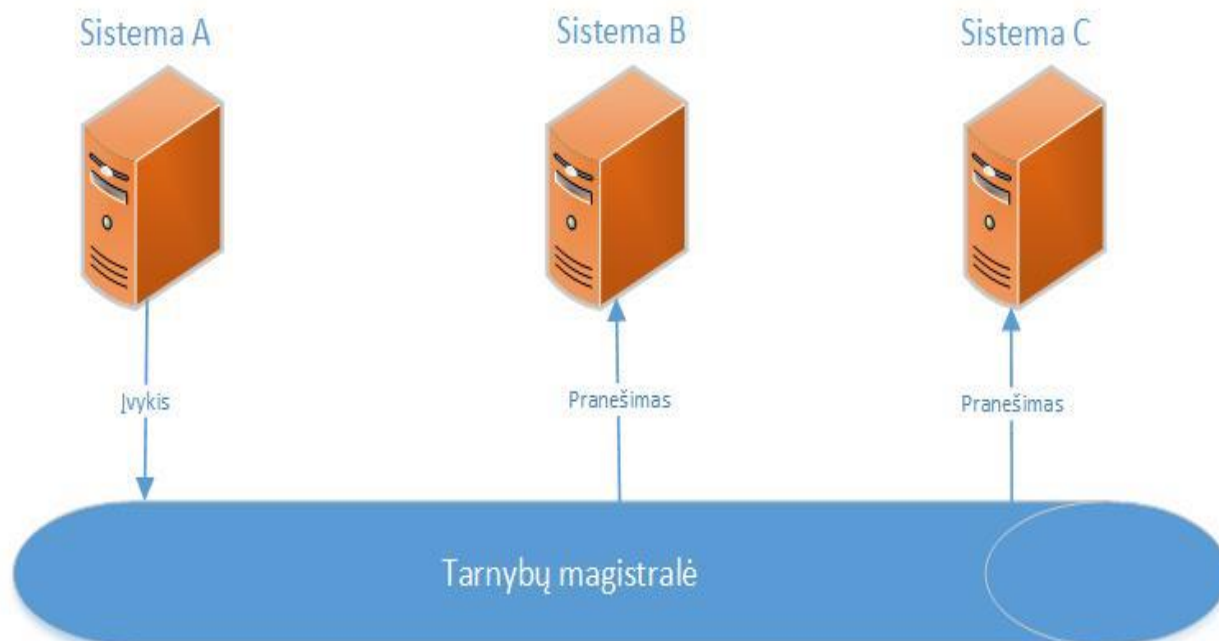
Nepaisant nuotolinio funkcionalumo iškvietimo trūkumų, jis dažnas integravimo būdas. Šis būdas turi daug realizacijų: CORBA [OMG15], COM [Mic15a], .NET Remoting [Mic15b], Java RMI [Ora15], žiniatinklio paslaugos naudojant SOAP ar REST protokolus. Pastarąjį dešimtmetį daug dėmesio skiriama SOAP ir REST žiniatinklio paslaugų, bei jas palaikančių technologijų tyrimams. Potti [PSK+12] atliko 20 skirtingų paslaugų iškvietimo atsako laiko bei pralaidumo (angl. *throughput*) matavimą naudodamas SOAP ir REST žiniatinklio paslaugas. Autorius nustatė nežymų REST žiniatinklio paslaugų pranašumą, tačiau išskėlė prielaidas, kad tokiems rezultatams įtakos galėjo turėti serverio galia ar tinklo pralaidumas.

Wagh [WT12] palyginamajame tyrime išskiria SOAP pranašumus kaip geriau pažįstamos technologijos, labiau tinkančios B2B integraciniams scenarijams. Kaip pagrindinius SOAP trūkumus autorius išskiria prisirišimą prie XML duomenų formato, papildomų įrankių poreikį, sunkų pritaikymą išskirstytose sistemose, daug didesnę atsako dydį. Pagrindiniai REST privalumai, anot autoriaus, yra visų duomenų tipų palaikymas, daug paprastesnis integracijos kūrimas, mažesni atsakai, mažas kliento ir serverio sukibimas. Tačiau autorius išvelgia ir trūkumas, kadangi tai gan nauja technologija, kuri nepilnai užtikrina saugumą ir žinučių patikimumą. Wagh [WT12] ir Potti [PSK+12] apžvelgia SOAP ir REST žiniatinklio paslaugas iš skirtingų perspektyvų, tačiau nepaisant nežymių trūkumų REST įvardijamas kaip pranašesnis.

Nuotolinio funkcionalumo iškvietimas yra gerai ištirtas integravimo būdas, todėl šiame darbe nėra vertinamos atskiros jo realizacijos. Dėl kitų autorių nustatytų privalumų ir populiarumo vertinimui pasirinktos REST ir SOAP žiniatinklio paslaugos. Žiniatinklio paslaugos, kaip nuotolinio funkcionalumo realizacija, vertinimui naudojamos tik tada, kai negalima vertinti paties integravimo būdo, nepasirinkus jokios realizacijos.

#### **1.3.4. Pranešimų integravimas**

Pagrindinis nuotolinio funkcionalumo iškvietimo trūkumas yra didelis sistemų sukibimas. Be to, įvykus klaidai integruojamoje sistemoje, integruojančių sistemų veikimas gali sutrikti. Mažas sukibimas ir klaidų tolerancija užtikrinama apsikeitime failais, tačiau jis turi didelę duomenų delną. Visas šias problemas išsprendžia pranešimų integravimas, kai greitai apsikeičiama nedideliais failais. Principinė šio integravimo būdo schema pavaizduota 6 pav.



7 pav. Pranešimų integravimo principinė schema

Plečiamumas ir mažas sistemų sukibimas – pagrindiniai pranešimų integravimo privalumai. Šis integravimo būdas garantuoja saugų ir patikimą pranešimų perdavimą. Pirmiausia, užtikrinama, kad pranešimas pasieks gavėją (galimi skirtingi užtikrinimo būdai, pvz.: kad gavėją pasiektų bent kartą arba tik kartą) ir kad jį pasieks būtent tą gavėją, kurį ir turėtų. Be to, šis būdas palaiko asinchroniškumą. Integruojamos sistemos nesiblokuoja ir išvengia greitaveikos problemų. Pranešimų integravimas leidžia sistemoms apsibrėžti norimą duomenų apsikeitimo formatą. Užtikrinant dažną ir greitą apsikeitimą failais duomenų delsą galima minimizuoti. Saugumas pranešimų integravime užtikrinamas taip pat kaip ir apsikeitime failais, papildomai galima naudoti duomenų šifravimą ar saugų ryšio kanalą. Pranešimų integravimas yra lengvai plečiamas, tačiau plečiamumas priklauso nuo pranešimų integravimo programinės įrangos.

Žinučių integravimas reikalauja papildomos programinės įrangos, dėl to gali būti brangus. Apsikeitimo žinutėmis programinė įranga, naudojant ją komerciniais tikslais, dažnai yra mokama. Be to, specialistai yra priklausomi nuo šios įrangos dokumentacijos ir techninės pagalbos kokybės. Dėl to šis integravimo būdas gali reikalauti daugiau specialistų darbo valandų lyginant jį su kitais integravimo būdais. Netinkamas asinchroninių procesų įvertinimas gali sukelti duomenų delsos problemų, kai žinutės yra apdorojamos vėliau negu tikimasi. Taigi žinučių integravimas gali būti brangus ir jo realizacija gali ilgai užtrukti, bei turi didesnę delsą lyginant jį su nuotolinio funkcionalumo iškvietimu.



Šiuo metu sukurta daug programinės įrangos, skirtos pranešimų integravimui, todėl ši programinė įranga tapo tyrimų objektu. Ahuja [AM14] savo tyrime vertino Active MQ [ASF15], Open MQ [Jav15] ir Mantaray MQ [Spa15] pagal žinučių skelbimo ir žinučių priėmimo pralaidumą ir delsą. Autorius nustatė, kad Active MQ turi didžiausią žinučių skelbimo ir priėmimo pralaidumą bei mažiausią delsą. Be to, autorius kokybiškai matavo programavimo lengvumą, programinės įrangos sudėtingumą, architektūrinį dizainą, programavimui skirtą laiką, saugumą, dokumentacijos detalumą ir techninės pagalbos kokybę. Ahuja nustatė, kad lengviausiai programuoti Mantaray MQ, tačiau visi kiti kokybiniai rodikliai geresni Active MQ. Taigi lyginant Open MQ, Mantaray MQ ir Active MQ, pastaroji geriausia vertinant tiek pagal kiekybinius, tiek pagal kokybinius rodiklius.

Informacinių sistemų integravimo būdai turi skirtingas savybes. Apsikeitimas failais ir bendra duomenų bazė yra primityvūs integravimo būdai, taikomi tik specifiniais atvejais. Be to, šiuose būduose sunku realizuoti pranešimų ar procesų integravimą, dėl to jie retai taikomi išorinėse integracijose. Nuotolinio funkcionalumo iškvietimas ir pranešimų integravimas leidžia integruoti procesus ir dėl to taikomas atliekant išorinę sistemų integraciją. Abu integravimo būdai užtikrina mažą duomenų delsą. Tai leidžia pasiekti didesnę vartotojų pasitenkinimą. Dėl to tiriant išorinę sistemų integraciją, pagrindiniai tyrimo objektai yra pranešimų integravimas ir nuotolinio funkcionalumo iškvietimas.

## 2. INTEGRAVIMO BŪDŲ IR SCENARIJŲ VERTINIMAS

### 2.1. Integravimo būdų vertinimo metodika

Vertinant integravimo būdus, programinio kodo sudėtingumas turi būti objektyvus ir matematiškai tikslus. Integravimo būdų kodo sudėtingumą galima vertinti dvejopai – pagal jau veikiančių integracijų programinį kodą arba programos dizainą. Siekiant palyginti integravimo būdus konkrečiu panaudos atveju, programos dizaino vertinimas yra tinkamesnis. Halstead sudėtingumas tenkina keliamus objektyvumo ir matematinio tikslumo reikalavimus. Jį galima išmatuoti bet kokios programos kodui, tačiau jis negali būti suskaičiuotas programų dizainui. McCabe[MB89] sudėtingumas, kuris taip pat tenkina minėtus reikalavimus, naudoja grafų analizę, dėl to tinka sistemų dizaino sudėtingumui vertinti. Kadangi gali būti skaičiuojamos programos dizainui, šiame darbe integravimo būdo programinio kodo sudėtingumui vertinti naudojamas McCabe ciklomatinis sudėtingumas.

Programinio kodo sudėtingumo vertinimui pasirinktas versle sutinkamas panaudos atvejis. Integruojanti sistema iš integruojamos gauna duomenų masyvą, kurį apdoroja. Šiame darbe vertinimui naudotas atvejis, kai masyvas sudarytas iš dviejų tekstinių verčių poros. Kiekvienam integravimo būdai aprašius integruojamos ir integruojančios programų grafus, išmatuotas programinio kodo sudėtingumas. Programinio kodo sudėtingumas pagal McCabe:

$$M = E - N + 2, \quad (2.1.1)$$

kur  $E$  – grafo briaunų skaičius,  $N$  – viršūnių skaičius. Sudėjus integruojančios ir integruojamos programų sudėtingumą, gaunamas integracijos programinio kodo sudėtingumas:

$$M = M_1 + M_2, \quad (2.1.2)$$

kur  $M_1$  – integruojančioje sistemoje parašytos programos ciklomatinis sudėtingumas,  $M_2$  – integruojamoje sistemoje.

Sistemų integravimo kontekste, sukibimas vertintas remiantis Pautasso [PZL08] darbu. Sukibimas dekomponuotas į tris dedamąsias – prieinamumą, stabilią sąsają ir funkcinį plečiamumą. Anot Pautasso, prieinamumas yra užtikrinamas, kai integruojamos sistemos veiklai sutrikus, klientų veiklai nėra pakenkiama. Pautasso prieinamumo sąvoka šiame darbe praplėsta į ją įskaitant ir užtikrinimą, kad laikino sutrikimo metu neapdorota informacija yra apdorojama arba sunaikinama, atstačius stabilų sistemos veikimą. Stabili sąsaja Pautasso darbe įvardijama kaip nekintamas elementarių operacijų rinkinys. Šiame darbe naudojamos programų sistemų inžinerijos elementarios operacijos – nuskaityti, įrašyti, atnaujinti ir ištrinti. Funkcinis plečiamumas, anot Pautasso, reikalingas integracijos evoliucijai. Integravimo būdas turi palaikyti galimybę praplėsti

duomenų modelį, pridėti naujas funkcijas užtikrinant, kad pakeitimai nepaveiks stabilaus sistemų veikimo ir integruojančios sistemos juos galės automatiškai aptikti ir pradėti naudoti.

Saugumas, lyginant integravimo būdus, vertinamas kaip atskirų saugumo komponentų užtikrinimas. Savola[SA09] darbe sukurta pranešimų programinės įrangos saugumo vertinimo metodika pritaikyta integravimo būdams vertinti. Prieinamumas, minimas Savola darbe, šiame darbe minimas vertinant sistemų sukibimą, todėl iš saugumo vertinimo jis pašalintas. Likę penki saugumo komponentai, kurie vertinami pagal 1.1 poskyryje nustatytas taisykles yra autentifikavimas, autorizacija, konfidencialumas, integralumas ir neišsigynimas.

Integracijos greitaveika priklauso nuo duomenų apdorojimo greičio. Chiu, Govindaju ir Bramley[CGB02] nustatė, kad XML duomenų vaizdavimas nulemia komunikacijos spartą tarp integruojamų sistemų ir yra kamštis SOAP pranešimų apdorojime. Duomenų vaizdavimo laiko vertinimas šiame darbe praplečiamas lyginant su Chiu, Govindaju ir Bramley darbu, vertinant vaizdavimą ne tik nuskaitymo, bet ir įrašymo sistemose ir matuojant šių laikų sumą, ją vadinant duomenų transliacijos laiku. Kiekvienas integravimo būdas susietas su kartu su juo versle taikomu duomenų vaizdavimo būdu, apskaitime failais vertinta kableliu atskirtų verčių masyvo dokumento transliacija, bendros duomenų bazės būde – reliacinėje duomenų bazėje saugomų duomenų, nuotolinio funkcionalumo iškvietime – XML dokumentų, pranešimų integravime – JSON dokumentų.

Transliacijos laikas matuotas, kaip laikų prieš vaizdavimą ir laikų po vaizdavimo skirtumų integruojančioje ir integruojamose sistemose suma. Vertinti du transliavimo atvejai – paprasti duomenys, kuriuose yra 16 ženklų masyvas ir kompleksiniai duomenys, kuriuose yra 16 ženklų tekstinio lauko, sveikojo skaičiaus ir loginės reikšmės rinkinys. Matavimai atlikti masyvo dydį didinant tarp 10.000 ir 500.000, kiekvieną matavimą pakartojant 1000 kartų. Matavimai atlikti sistemoje, kurios parametrai:

- Du 2.6GHz Intel Xeon;
- Microsoft Windows Server 2012 R2 Standard;
- 6144Mb atminties;
- Sun Java v1.8.0\_51.

Sistemų integravime duomenų transliacija yra apatinė riba, nulemianti, kaip greitai vienos sistemos duomenys gali tapti kitos sistemos duomenimis. Visas laikas, kurį duomenys keliauja, G. Hohpe ir B. Woolf integravimo būdų kontekste vadinamas duomenų delsa (angl. *latency*). Duomenų delsa įskaito duomenų keliavimą iš vienos sistemos į kitą bei kitas su sistemų

komunikavimu susijusias veiklas (pvz.: autentifikavimą). Duomenų delsa šiame darbe vertinta analizuojant duomenų apsikeitimo proceso žingsnius kiekviename integravimo būdu. Duomenų delsa nulemia duomenų parsisiuntimo laikas, kuris priklauso nuo duomenų dydžio. Dėl to vertinant duomenų delną įvertintas ir kiekviename integravimo būdu perduodamas duomenų dydis.

## **2.2. Integravo būdu vertinimas**

### **2.2.1. Apsikeitimas failais**

Apsikeitimo failais integravimo būdas naudojamas atliekant duomenų arba procesų integraciją. Jei atliekant duomenų integraciją failais apsieičiama retai, galima neatsižvelgti į tai, kad failas gali būti nuskaitomas ir įrašomas vienu metu. Atliekant procesų integraciją, integruojančios sistemos pasirenka tokį apsikeitimo failais intervalą, kad nebūtų pakenkta stabiliam sistemų veikimui. Kai failais apsieičiama dažnai, integravimas tampa panašus į pranešimų. Šiuo atveju bendra laikmena tarnauja kaip tarnybų magistralė. Toks integravimo būdas šiame darbe nėra nagrinėjamas.

Priklausomai nuo integruojamų sistemų skaičiaus ir jų turimų teisių ir apsikeitimą failais galima realizuoti dvejopai. Apsikeitimas failais gali būti realizuojamas kai nuskaitančioji sistema yra atsakinga ir už failų ištrynimą (kaip pavaizduota 4a pav.) ir, kai įrašančioji sistema yra atsakinga už failų ištrynimą (kaip pavaizduota 4b pav.). 4a pav. pavaizduotame integravimo būde išvengiama problemos, kad failas bus nuskaitomas ir trinamas vienu metu, nes tai atlieka ta pati sistema. Toks integravimo būdas tinka tik integruojant dvi sistemas, todėl, integruojant daugiau sistemų, įrašančioji yra atsakinga už jo ištrynimą.

#### **2.2.1.1 Programinio kodo sudėtingumas**

Vertinant integracijos programinio kodo sudėtingumą apsikeitimo failais integravimo būde vertinamas integruojamos ir integruojančios sistemų programos kodas. 2.1 poskyryje aprašytas panaudos atvejis, kai integruojanti sistema siekia nuskaityti duomenų masyvą, apsikeitime failais gali būti realizuotas dviem būdais. Pirmuoju būdu integruojanti sistema yra atsakinga už failų pašalinimą, antruoju – integruojama.

Kai integruojanti sistema yra atsakinga už failų ištrynimą, įrašančios programos pseudokodas pavaizduotas 8 pav.. Šioje programoje eilutė po eilutės duomenų masyvas įrašomas į failą, kuris vėliau perduodamas kitoms sistemoms nuskaityti. Programos grafo, kuris pavaizduotas 9 pav., viršūnės atitinka pseudokodo elementus su atitinkamu numeriu. Įrašymo programos grafe yra  $N=5$  viršūnės ir  $E=5$  briaunos. Remiantis 2.1.1 formule šios programos ciklomatinis sudėtingumas:

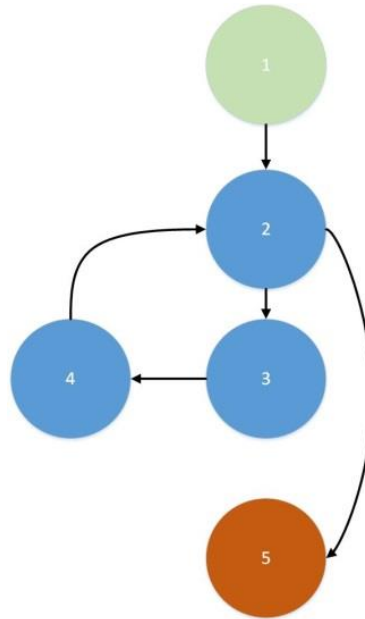
$$M = E - N + 2 = 5 - 5 + 2 = 2.$$

```

for ((1)int j=0;(2)j<įrašų_skaičius;(4)j++){
(3) pridėti atributą nr. 1;
(3) pridėti atributą nr. 2;
...
    pridėti eilutės pabaigos ženklą;
}
(5) išeiti;

```

8 pav. Įrašnčios programos pseudokodas, kai integruojanti sistema atsakinga už failų pašalinimą



9 pav. Failo įrašymo programos grafas, kai integruojanti sistema yra atsakinga už failų pašalinimą

Kai integruojanti sistema yra atsakinga už failų pašalinimą, nuskaitanti sistema turi įsitikinti, kad failas nėra įrašomas prieš pradėdama jį nuskaityti. Nagrinėjamu panaudos atveju pridėjus apribojimą, kad vienas failas yra skirtas tik vienai sistemai, prieš ištrinant failą nebūtina įsitikinti, kad jis yra nuskaitomas. Failas yra nuskaitomas eilutė po eilutės pildant nuskaitymo masyvą, kaip parodyta programos pseudokode 10 pav. Baigus nuskaitymą, failas yra ištrinamas. Šios programos grafas, kuris pavaizduotas 11 pav., turi  $N=11$  viršūnių ir  $E=14$  briaunų. Remiantis 2.1.1 formule šios programos ciklomatinis sudėtingumas:

$$M = E - N + 2 = 14 - 11 + 2 = 5.$$

Ciklomatinis sudėtingumas šiai programai didėja didėjant vieno masyvo elemento atributų skaičiui. Kiekvienas papildomas atributas prideda papildomą sąlygos sakinį, todėl programos grafe papildomai atsiranda viena viršūnė ir dvi briaunos. Todėl programos ciklomatinis

sudėtingumas yra priklausomybė nuo elemento atributų skaičiaus  $n$ . Remiantis 2.1.1 formule 10 pav. pavaizduotam pseudokodui priklausomybė yra:

$$M(n) = 3 + n. \tag{2.2.1}$$

```

(1) while(įrašoma) {
(2)     laukti;
    }

(3) nuskaityti_failą;

    for((3)int k =0;(4)k<eilučių_skaičius; (10)k++){

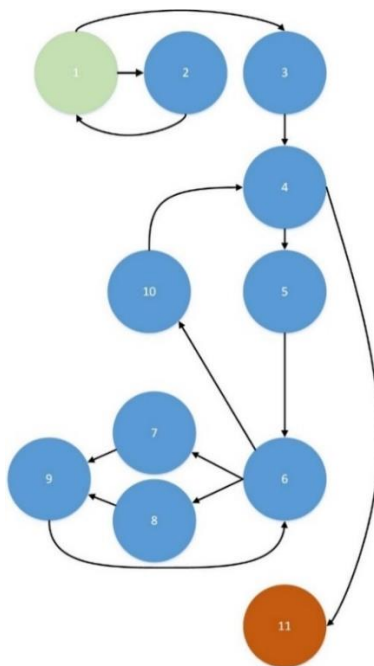
        for((5)int i=0;(6)i<verčių_skaičius;(9)i++){

            (6) switch (i){
                (7) case 0:{
                    įrašyti vertę 0 iš k-osios eilutės;
                    break;
                }
                (8) case 1:{
                    įrašyti vertę 1 iš k-osios eilutės;
                    break;
                }
                ...
            }
        }
    }

(11) ištrinti failą;

```

10 pav. Nuskaitančios programos pseudokodas, kai ji atsakinga už failų ištrynimą



11 pav. Failo įrašymo programos grafas, kai nuskaitančioji sistema yra atsakinga už failų pašalinimą

Kai integruojama sistema įrašo ir ištrina failus, programos kodo sudėtingumas yra didesnis. Įrašius naują failą, senas failas yra ištrinamas, jei jis nėra nuskaitomas (papildomai 5 ir 6

žingsniai). Programos pseudokodas yra pavaizduotas 12 pav. Tokios programos grafas, kuris pavaizduotas 13 pav., turi  $N=7$  viršūnės ir  $E=8$  briaunos. Šios programos sudėtingumas yra didesnis nei anksčiau nagrinėtos programos dėl papildomai vykdomų 5 ir 6 žingsnių. Pagal 2.1.1 formulę, ciklomatinis programos sudėtingumas yra:

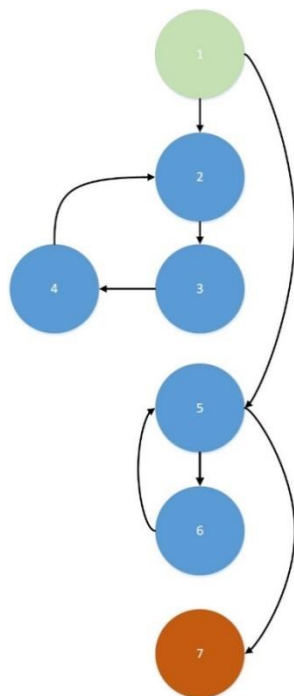
$$M = E - N + 2 = 8 - 7 + 2 = 3.$$

```

for ((1)int j=0;(2)j<įrašų_skaičius;(4)j++){
    (3) pridėti atributą nr. 1;
    (3) pridėti atributą nr. 2;
    ...
    pridėti eilutės pabaigos ženklą;
}
(5) while(nuskaitomas){
(6)     laukti;
}
(7) ištrinti seną failą;

```

12 pav. Integruojamos sistemos programos pseudokodas, kai ji atsakinga už failų pašalinimą



13 pav. Failo įrašymo programos grafas, kai integruojama sistema yra atsakinga už failų pašalinimą

Kai integruojama sistema yra atsakinga už failų pašalinimą, integruojančios sistemos programos kodas yra panašus į prieš tai nagrinėtą atvejį, kai integruojama sistema ištrina failą. Patikrinus, ar failas nėra įrašomas arba trinamas, failas nuskaitomas eilutė po eilutės. Programos

pseudokodas pavaizduotas 14 pav. Tokios programos grafas, kuris pavaizduotas 15 pav., turi  $N=12$  viršūnių ir  $E=16$  briaunų. Šios programos ciklomatinis sudėtingumas pagal 2.1.1 formulę:

$$M = E - N + 2 = 16 - 12 + 2 = 6.$$

Ciklomatinis sudėtingumas šiai programai didėja didėjant nuskaitomo masyvo atributų skaičiui  $n$ . Kiekvienas naujas atributas, kaip ir anksčiau nagrinėtu atveju, pridės vieną sąlygos sakinį. Dėl to grafe papildomai atsiras viena viršūnė ir dvi briaunos. Taigi sudėtingumo priklausomybė nuo nuskaitomų verčių skaičiaus yra tiesinė:

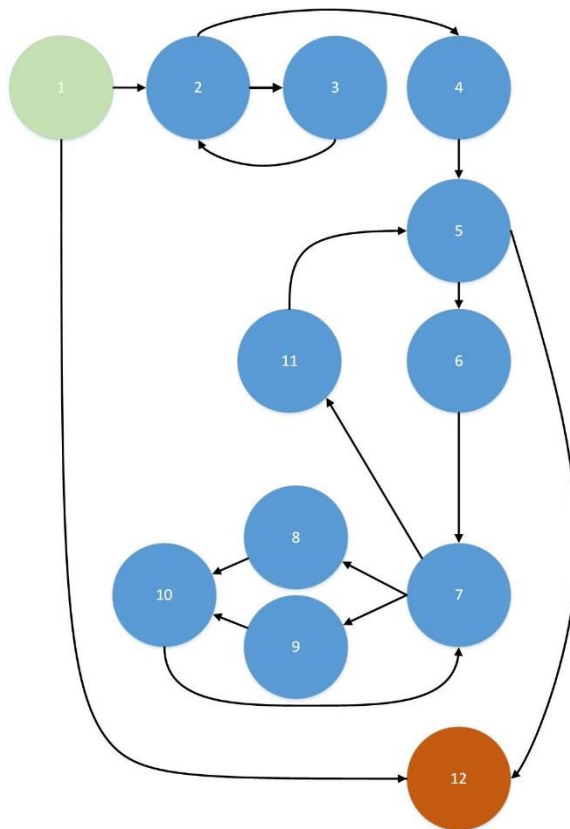
$$M(n) = 4 + n. \tag{2.2.2}$$

```
(1) if(!netrinamas){
(2) while(įrašomas){
(3)     laukti;
        }
(4) nuskaityti_failą;

    for((4)int k =0;(5)k<eilučių_skaičius; (11)k++){
        for((6)int i=0;(7)i<verčių_skaičius;(10)i++){
(7)     switch (i){
(8)     case 0:{
                įrašyti vertę 0 iš k-osios eilutės;
                break;
            }
(9)     case 1:{
                įrašyti vertę 1 iš k-osios eilutės;
                break;
            }
                ...
        }
    }
}
(12)išeiti;
```

14 pav. Failo nuskaitymo programos pseudokodas, kai įrašąčioji sistema yra atsakinga už failų pašalinimą





15 pav. Failo nuskaitymo programos grafas, kai įrašnčioji sistema yra atsakinga už failų pašalinimą

Dėl mažesnio sąlygos sakinių skaičiaus integracijai, kurioje integruojanti sistema ištrina failus, reikalingas mažiau sudėtingas kodas, lyginant ją su integracija, kurioje integruojama sistema ištrina failus. Kai masyvo elementas, įrašomas į failą, turi du atributus, o integruojanti sistema yra atsakinga už failo ištrynimą, remiantis 2.1.2 formule integracijos ciklominis sudėtingumas yra:

$$M = M_1 + M_2 = 2 + 5 = 7,$$

kur  $M_1$  ir  $M_2$  – atitinkamai integruojamos ir integruojančios sistemų programinio kodo sudėtingumai. Kai integruojama sistema yra atsakinga už failų ištrynimą, integracijos ciklominis sudėtingumas yra:

$$M = M_1 + M_2 = 3 + 6 = 9.$$

### 2.2.1.2 Sistemų sukibimas

Atliekant sistemų integravimą, kaip pavaizduota 4a ir 4b pav., prieinamumas yra užtikrinamas, kadangi sistemų veikimas tarpusavyje nepriklauso. Kai laikinai neprieinama sistema A (4b būde sistema A arba sistema C), atstačius normalų šios sistemos veikimą, ji nuskaito visus

talpykloje susikaupusius failus ir juos ištrina kaip įprasta. Tuo atveju, kai neprieinama sistema B, sistema A (4b būde sistemos A ir C) negauna duomenų. Sistemai B atstačius normalų veikimą, failai gali būti talpinami į talpyklą, kur juos nuskaityt sistema A. Kai failų talpyklos veikimas sutrinka, sistemos negali tinkamai veikti ir negali apsieisti informacija. Norint užtikrinti integracijos veikimą, galima pridėti antrinę talpyklą, į kurią kreipiamasi, kai pagrindinė talpykla yra nepasiekiamas. Taigi apsieitime failais prieinamumas užtikrinamas, kai failais apsieičiama per failų talpyklą.

Apsieitime failais stabili sąsaja nėra užtikrinama. Integravimo būde elementarios funkcijos yra failo įrašymas ir ištrynimai. Šios funkcijos nekinta, tačiau apsieitimo failais procesas nėra standartinis. Vienintelis būdas užtikrinti stabilumą – susitarimas tarp sistemų. Susitariant nusakoma, kas ir kada įrašys, nuskaitys ir pašalins failus. Priklausomai nuo susitarimo tarp sistemų, joms turi būti suteikiamos reikalingos teisės. Failų, kuriais apsieičiama, formatas šiame integravimo būde nėra apibrėžtas. Sistemos gali naudoti tiek tarpusavyje suderintą formatą, tiek naudoti duomenų formato standartą (pvz.: XML, JSON). Taigi apsieitime failais stabili sąsaja priklauso nuo susitarimo tarp integruojamų sistemų, todėl nėra užtikrinama.

Papildomumas apsieitime failais yra užtikrinamas tada, kai failo, kuriuo apsieičiama, duomenų modelį galima praplėsti. Kai sistemos naudoja tarpusavyje suderintą duomenų apsieitimo formatą, jis gali turėti mažą klaidų toleranciją, dėl to naujų struktūrų pridėjimas gali sutrikdyti vienos iš sistemų veikimą. Be to, duomenų modelio pakitimų aptikimas integruojančiose sistemose negali būti automatizuotas. Keičiant duomenų modelį, reikia visas integruojamas sistemas perjungti prie naujo duomenų modelio vienu metu arba kurį laiką palaikyti abu duomenų modelius atitinkančio failų įrašymą ir nuskaitymą. Taigi papildomumas failų apsieitime nėra užtikrinamas, nes neužtikrinamas automatizuotas duomenų modelio keitimas.

1 lentelė. Sistemų sukibimas apsieitime failais

Sistemų sukibimo komponentas	Užtikrinamas
Prieinamumas	1
Stabili sąsaja	0
Papildomumas	0
<b>Iš viso:</b>	<b>1</b>

Apsieitime failais sistemos yra stipriai sukibusios, kadangi užtikrinamas tik vienas iš trijų mažo sukibimo komponentų. Kaip pavaizduota 1 lentelėje, apsieitime failais yra užtikrinamas tik prieinamumas ir tik tuo atveju, kai sistemos failus talpina į failų talpyklą, kurios veikimas nepriklauso nuo integruojamų sistemų. Apsieitimo failais sąsaja nėra standartinė, remiamasi

susitarimu tarp sistemų, todėl negalima užtikrinti jos stabilumo. Papildomumas apskaitime failais nėra užtikrinamas, nes duomenų struktūros pakitimai gali sutrikdyti integruojamų sistemų veikimą.

### 2.2.1.3 Saugumas

Autentifikavimas failų apskaitime yra užtikrinamas. Atliekant sistemų integravimą, kaip pavaizduota 4 pav., failų perdavimui naudojamas FTP protokolas. Siekiant užtikrinti autentifikavimą, anoniminis prisijungimas prie failų talpyklos yra negalimas, sistemos autentifikavimui naudoja vartotojo vardo ir slaptažodžio porą. Norint užšifruoti turinį, FTP gali būti apsaugotas SSL/TLS protokolu. Kai integruojamos daugiau nei dvi sistemos, autorizacijai užtikrinti naudojant FTP protokolą kiekvienai sistemai sukuriama atitinkama direktorija. Direktorijose talpinami failai pagal jo adresatą, reikiamiems vartotojams priskiriamos skaitymo ir rašymo teisės. Taigi autorizacija ir autentifikavimas apskaitime failais yra užtikrinami.

Skaitymo ir rašymo teisės leidžia apsaugoti failus nuo neleistinos prieigos. Panaudojant viešojo rakto kriptografiją, failai gali būti užšifruojami ir perskaitomi tik vartotojams, turintiems privatų raktą. Taip apskaitime failais užtikrinamas konfidencialumas. Apsikeičiant failais, paskutinis failo perdavimo į FTP serverį žingsnis yra patikrinti ar kontrolinė suma, kurią gauna serveris sutampa su siuntėjo kontroline suma. Tokiu atveju užtikrinama, kad failas nebuvo pakeistas jį perduodant ir užtikrinamas integralumas. Neišsigynimas failų apskaitime nėra užtikrinamas. Norint jį užtikrinti, galima papildomai naudoti užtikrinančią programinę įrangą, vadinamą valdoma failų perdavimu (angl. *Managed File Transfer, MFT*) programine įranga. Taigi apskaitime failais užtikrinamas integralumas ir konfidencialumas, o neišsigynimui užtikrinti reikia naudoti papildomą programinę įrangą.

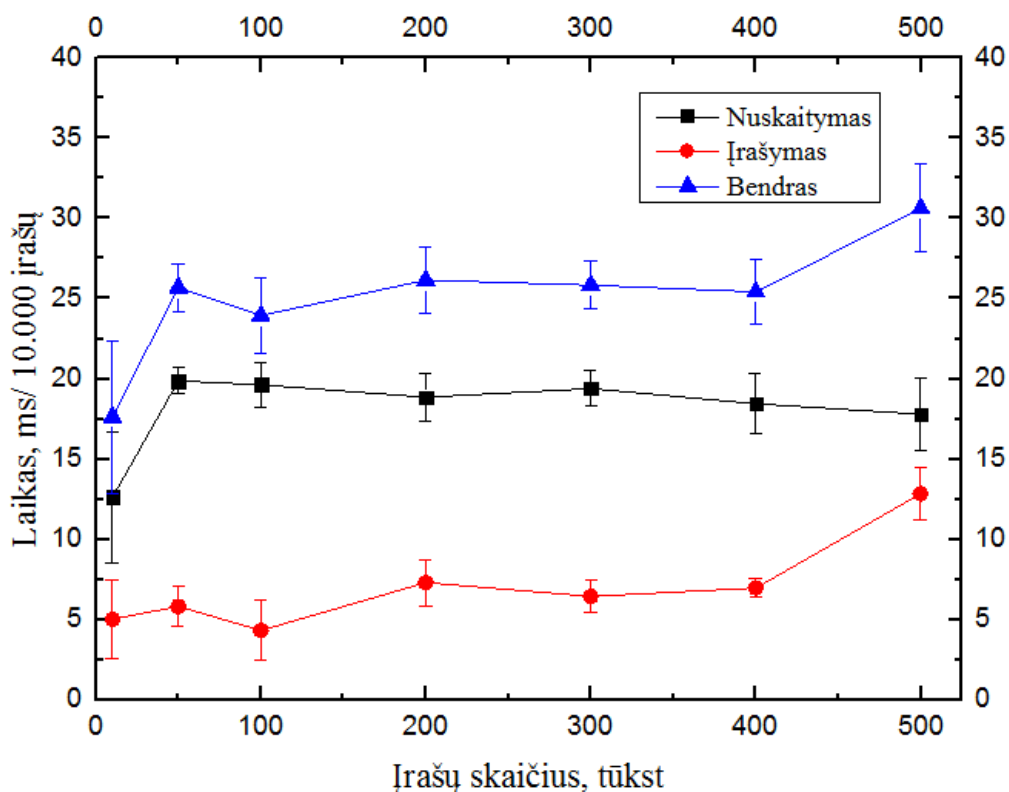
2 lentelė. Sistemų sukibimas apskaitime failais

Saugumo komponentas	Užtikrinamas
Autentifikavimas	1
Autorizacija	1
Konfidencialumas	1
Integralumas	1
Neišsigynimas	0
<b>Iš viso:</b>	<b>4</b>

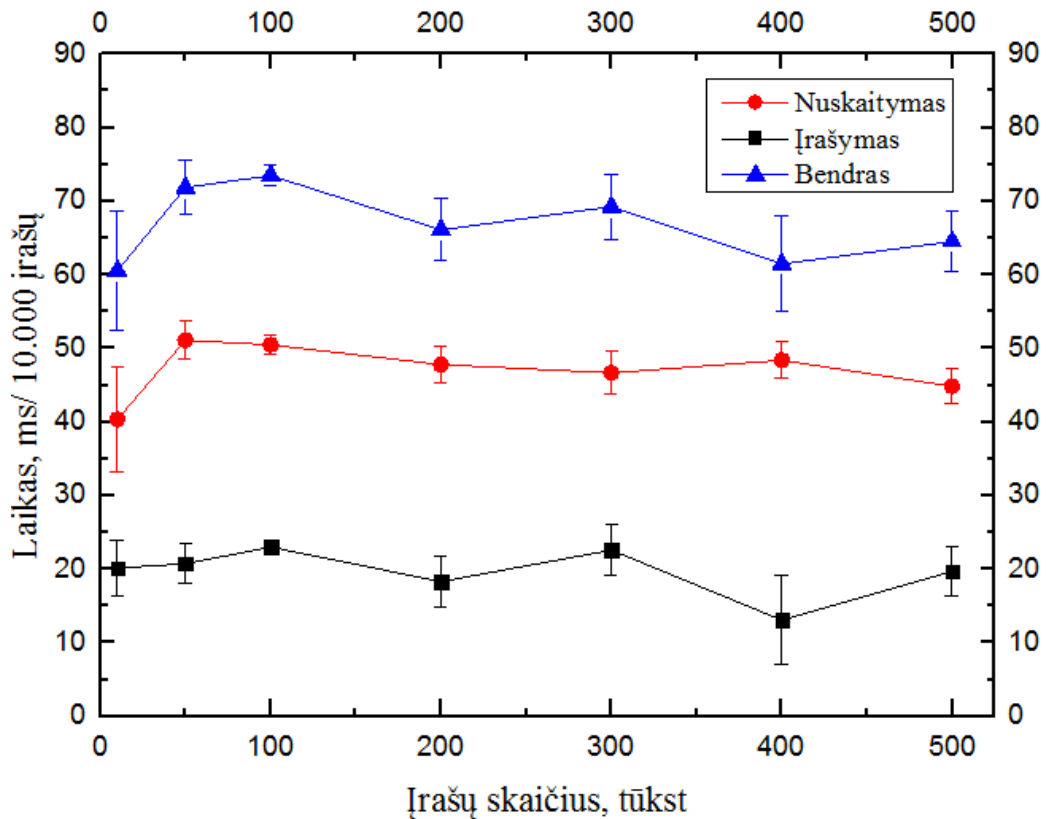
Apsikeitimo failais saugumas yra užtikrinamas 4 iš 5 saugumo komponentų. Kaip pavaizduota 2 lentelėje, apsikeitimas failais užtikrina sistemų autentifikavimą ir autorizaciją, konfidencialumą ir integralumą. Visų jų užtikrinamas priklauso nuo saugumo užtikrinimo failų talpykloje – anonimiškos prieigos ribojimo, sistemoms skirtų vartotojų sukūrimo, integruojamų sistemų failams saugoti skirtų direktorijų, failų šifravimo. Failų talpyklos naudojimas neleidžia užtikrinti neišsigynimo, tačiau jį užtikrinti galima naudojant tam skirtą programinę įrangą.

#### 2.2.1.4 Duomenų transliacijos laikas

Apsikeitimo failais duomenų transliacijos laikui matuoti pasirinktas panaudos atvejis, kai integruojanti sistema nuskaityto integruojamos sistemos duomenis, pateikiamus kableliais atskirtu formatu. 2.1 poskyryje nusakytomis sąlygomis atlikus eksperimentą, gauti rezultatai, kurie pavaizduoti 16 pav. ir 17 pav. Eksperimentiškai nustatyta, kad tirtame įrašų skaičiaus intervale, duomenų transliacijos laikas nepriklauso nuo įrašų skaičiaus faile. Tarp 50.000 ir 500.000 tekstinio tipo ir kompleksinių duomenų transliacijos laikas yra pastovus ir atitinkamai lygus  $25.95 \pm 2.45$  ms ir  $67.79 \pm 4.33$  ms. Nuskaitant 10.000 dydžio masyvą gauta mažesnė duomenų transliacija gali būti paveikta matavimo technikos tikslumo ir turi didelę paklaidą.



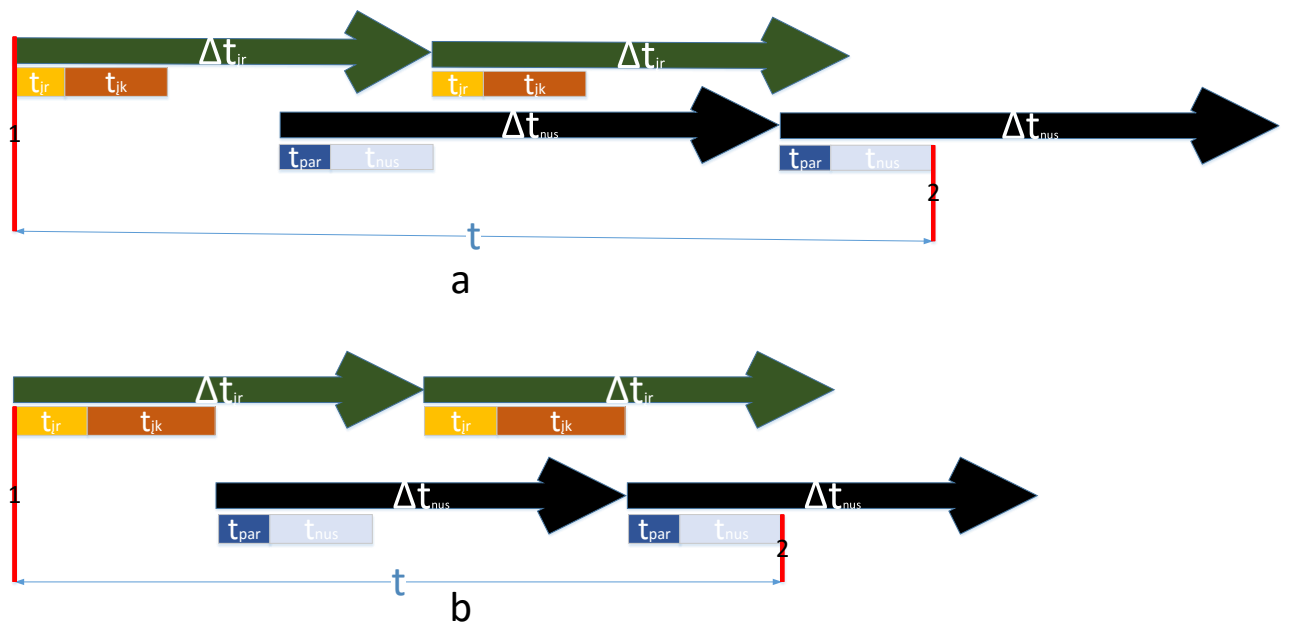
16 pav. Duomenų transliacijos laiko priklausomybė nuo įrašų skaičiaus apdorojant tekstinio tipo duomenų masyvą apsikeitimo failais



17 pav. Duomenų transliacijos laiko priklausomybė nuo įrašų skaičiaus apdorojant kompleksinius (tekstinio, skaičiaus ir loginės reikšmės rinkinio) duomenų masyvą apsikeitime failais

### 2.2.1.5 Duomenų delsa

Duomenų delsa apsikeitime failais atsiranda dėl to, kad sistemos duomenimis keičiasi fiksuotais intervalais, joms reikia autentifikuotis, įrašyti, nuskaityti ir ištrinti failus. Tarkime, kad nuskaitančioji sistema failus nuskaityti intervalais, kurių trukmė  $\Delta t_{nus}$ , įrašančioji sistema intervalais, kurių trukmė  $\Delta t_{ir}$ . Integracijos proceso dalys yra failo įrašymas, kurio trukmė yra  $t_{ir}$ , failo įkėlimas į saugyklą  $t_{ik}$ , failo parsisiuntimas, kurio trukmė yra  $t_{pars}$  ir failo nuskaitymas, kurio trukmė  $t_{nus}$ . Duomenų nuskaitymas ir įrašymas yra duomenų transliacijos laikas  $t_{tr} = t_{nus} + t_{ir}$ .



18 pav. Duomenų delsa apsikeitime failais. a – kai delsa maksimali; b – kai delsa minimali

Kaip pavaizduota 18 pav. duomenų delsa priklauso nuo sistemų suderinimo. Jei sistemos yra suderintos ( $\Delta t_{nus} = \Delta t_{ir}$ ), tai duomenų delsa yra pastovi. Jei sistemos yra nesuderintos duomenų delsa kinta. Tuo atveju, kai nuskaitančioji sistema, nuskaito duomenis prieš pat kitai sistemai juos įrašant (kaip pavaizduota 18a pav.), gaunama didžiausia duomenų delsa. Duomenų delsa tokiu atveju  $t = \Delta t_{ir} + \Delta t_{nus}$ . Tokiu atveju 1 laiko momentu įrašyti duomenys dar gali būti naudojami nuskaitančioje sistemoje momentu 2. Tuo atveju, kai nuskaitančioji sistema nuskaito duomenis iškart po jų įrašymo (kaip pavaizduota 18b pav.), gaunama mažiausia duomenų delsa. Jeigu sistemos veikia taip, kad  $\Delta t_{ir} = \Delta t_{nus} = t_{tr} + t_{pars} + t_{jk}$ , gaunama minimali delsa, kuri lygi  $t = 2(t_{tr} + t_{pars} + t_{jk})$ . Taigi gauname, kad duomenų delsa failų integravime yra:

$$2(t_{tr} + t_{pars} + t_{jk}) \leq t \leq \Delta t_{ir} + \Delta t_{nus} \quad (2.2.3)$$

Kai sistemose failais apsikeičiama dažnai, duomenų delsa nulemia duomenų transliacija ir failų įkėlimo bei parsisiuntimo sparta. Dėl šios priežasties mažesnė duomenų delsa gaunama tada, kai apsikeičiama mažesnio dydžio failais. Kai sistemose failais apsikeičiama retai, duomenų delsa nulemia apsikeitimo intervalas.

## 2.2.2. Bendra duomenų bazė

Bendros duomenų bazės integravimo būdas gali būti naudojamas atliekant duomenų arba procesų integraciją. Šiame darbe nagrinėjamos dvi galimos šio integravimo būdo realizacijos – duomenis nuskaityti ir įrašyti gali visos sistemos (kaip pavaizduota 5a pav.) ir duomenis įrašo viena sistema, o kitos sistemos juos tik nuskaityti. Pastaroji gali būti realizuota apribojant prieigos teises ir suteikiant prieigą prie realios duomenų bazės (kaip pavaizduota 5b pav.) arba atliekant duomenų kopijavimą (angl. *replication*) į kitą duomenų bazę, iš kurios duomenis nuskaityti integruojančios sistemos (kaip pavaizduota 5c pav.).

### 2.2.2.1 Programinio kodo sudėtingumas

Kai integruojama sistema duomenų saugojimui naudoja tą pačią duomenų bazę, kuri naudojama bendros duomenų bazės integracijai, duomenų įrašymui aprašyti nereikia papildomo programinio kodo. Nuskaitančioji sistema aprašo duomenų nuskaitymo programą, kurios pseudokodas pavaizduotas 19 pav. Programoje nėra atliekamas tikrinimas, ar sėkmingai pavyko prisijungti prie duomenų bazės ir įvykdyti užklausą, dėl to sumažėja klaidų tolerancija. Nuskaitant duomenis, kiekvienas užklausos rezultatas yra atvaizduojamas į sistemos įrašą.

```
(1) prisijungti prie duomenų bazės;  
(1) įvykdyti sql užklausą;  
  
for ((1)i=0;(2)i<eilučių skaičius;(4)i++){  
  (3) atvaizduoti irasa();  
}  
  
(5) iseiti;
```

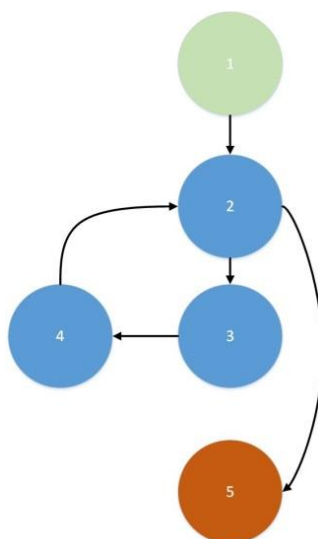
19 pav. Duomenų nuskaitymo programos pseudokodas integruojant bendros duomenų bazės integravimo būdu

Programos grafe, kuris pavaizduotas 20 pav., yra  $N=5$  viršūnės ir  $E=5$  briaunos. Šios programos ciklominis sudėtingumas remiantis 2.1.1 formule:

$$M_{nus} = 5 - 5 + 2 = 2.$$

Kadangi integruojama sistema nereikalauja papildomo programinio kodo, todėl integruojamos sistemos programinio kodo sudėtingumas:

$$M_{ir} = 1.$$



20 pav. Duomenų bazės nuskaitymo programos grafas

Remiantis 2.1.2 formule integracijos programinio kodo sudėtingumas yra:

$$M = M_{nus} + M_{ir} = 1 + 2 = 3.$$

### 2.2.2.2 Sistemų sukibimas

Integruojant sistemas 5 pav. pavaizduotais integravimo būdais, sistemų veikimas nepriklauso tarpusavyje. Sistemų veikimas gali sutrikti, kai sutrinka duomenų bazės veikla. Siekiant užtikrinti didelį prieinamumą, galima pridėti antrinę duomenų bazę, į kurią kreipiamasi, kai pagrindinė duomenų bazė nėra pasiekama. Stabili sąsaja integravimo būde yra užtikrinama, kadangi nepriklausomai nuo pasirinkto duomenų bazės tipo ar duomenų bazių valdymo sistemos elementariosios operacijos yra nekintamos. Tai – įrašyti (angl. *insert*), ištrinti (angl. *delete*), atnaujinti (angl. *update*) ir nuskaityti (angl. *select*). Taigi prieinamumas ir stabili sąsaja bendros duomenų bazės integravimo būde yra užtikrinami.

Papildomumas bendros duomenų bazės integravime nėra užtikrinamas. Duomenų bazės duomenų schemas pakeitimai turi būti derinami su visomis sistemomis, kurios naudoja tą duomenų bazę. Naujos struktūros pridėjimas neturi neigiamo poveikio sistemų veikimui, tačiau esamos struktūros pakeitimas (pvz.: duomenų tipo pakeitimas, lentelės pašalinimas) gali sukelti veikimo sutrikimus integruojamose sistemose. Be to, pridėjus naują struktūrą, reikia papildomo programavimo, kad ji būtų nuskaityta integruojančiose sistemoje. Taigi duomenų modelio pakeitimus bendros duomenų bazės integravimo būde privaloma derinti su integruojamomis sistemomis, todėl prieinamumas nėra užtikrinamas.



3 lentelė. Sistemų sukibimas bendros duomenų bazės integravimo būde

Sukibimo komponentas	Užtikrinamas
Prieinamumas	1
Stabili sąsaja	1
Papildomumas	0
<b>Iš viso:</b>	<b>2</b>

Atliekant bendros duomenų bazės integravimą sistemos yra sukibusios. Kaip pavaizduota 3 lentelėje, bendros duomenų bazės integravimo būde užtikrinamas prieinamumas, kadangi sistemų veikimas priklauso nuo duomenų bazės veikimo, tačiau nepriklauso nuo kitų sistemų veikimo. Integravimo būde garantuojama stabili sąsaja, kadangi elementarių funkcijų rinkinys yra apibrėžtas ir nekintantis. Papildomumas integravimo būde nėra užtikrinamas, kadangi pakitus duomenų struktūrai, integruojančių sistemų veikimas gali sutrikti arba jos duomenis gali nuskaityti neteisingai.

### 2.2.2.3 Saugumas

Atliekant sistemų integravimą, kaip pavaizduota 5 pav., prisijungimui prie duomenų bazės naudojama vartotojo vardo ir slaptažodžio pora, o anonimiška prieiga yra apribota, taip užtikrinant autentifikavimą. Sistemoms sukuriama vartotojai, kuriems suteikiamos tinkamos teisės, užtikrinant autorizaciją. 5a pav. pavaizduotame integravimo būde visos sistemos turi nuskaitymo ir įrašymo teises, 5b pav. ir 5c pav. pavaizduotuose integravimo būde – visos sistemos turi nuskaitymo teises, o sistema A turi ir įrašymo teises. Be to, vartotojams galima apriboti teises prieiti prie tam tikrų duomenų, duomenis, saugomus duomenų bazėje, galima šifruoti taip užtikrinant konfidencialumą. Taigi bendros duomenų bazės integravimo būde užtikrinama autorizacija, autentifikavimas ir konfidencialumas sukuriama sistemoms atskirus vartotojus ir suteikiant jiems reikiamas teises.

Integralumas bendros duomenų bazės integravimo būde nėra užtikrinamas. Atliekant integravimą, kaip pavaizduota 5a pav. sistemos gali atsidurti neapibrėžtoje būsenoje, kai dvi sistemos nori pakeisti tą patį duomenų bazės įrašą. Jei nėra taikomos papildomos apsaugos priemonės (pvz.: įrašo versijos saugojimas), duomenų bazėje bus išsaugotas įrašas, kurį įrašys vėliau įrašanti sistema. Jei tik viena sistema turi įrašymo teises (5b pav. ir 5c pav. pavaizduotiems integravimo būdams), duomenų integralumas yra užtikrinamas. Kaip minėta, duomenų bazėse vartotojams identifikuoti naudojama vartotojo vardo ir slaptažodžio pora. Trečioji šalis, žinanti vartotojo vardą ir slaptažodį, gali juo apsimesti, todėl neišsigynimas nėra užtikrinamas. Taigi

integralumas užtikrinamas tada, kai tik viena sistema turi įrašymo teises, o neišsigynimas nėra užtikrinamas.

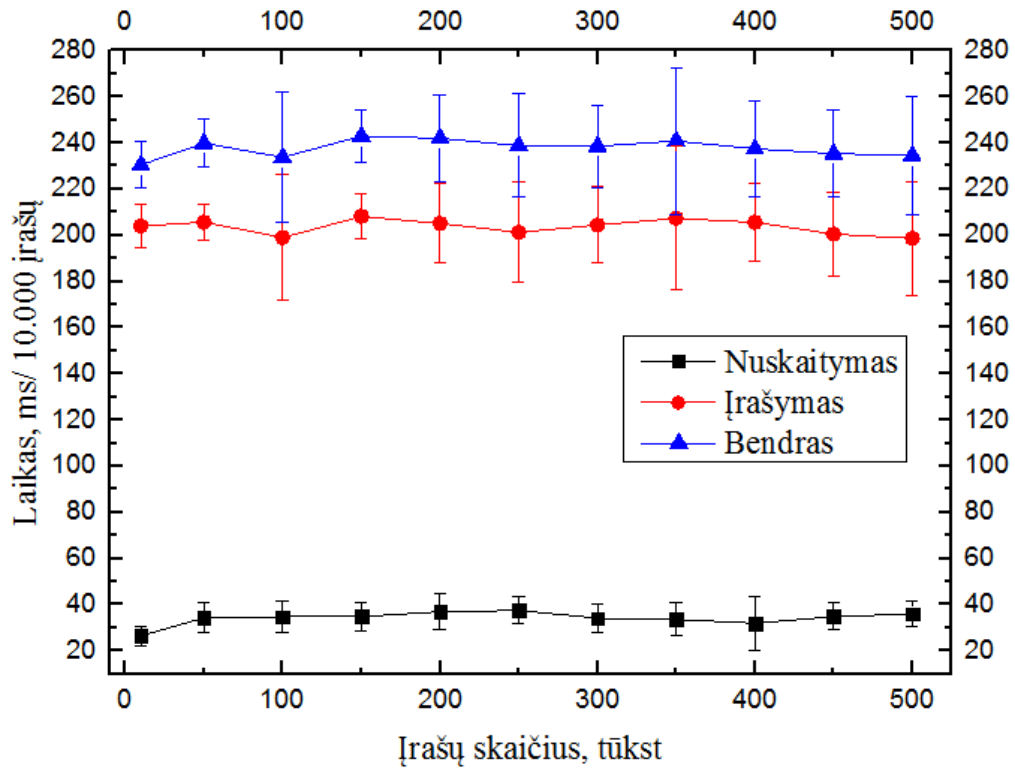
4 lentelė. Saugumas bendros duomenų bazės integravimo būde

Saugumo komponentas	Užtikrinamas
Autentifikavimas	1
Autorizacija	1
Konfidencialumas	1
Integralumas	0
Neišsigynimas	0
<b>Iš viso:</b>	<b>3</b>

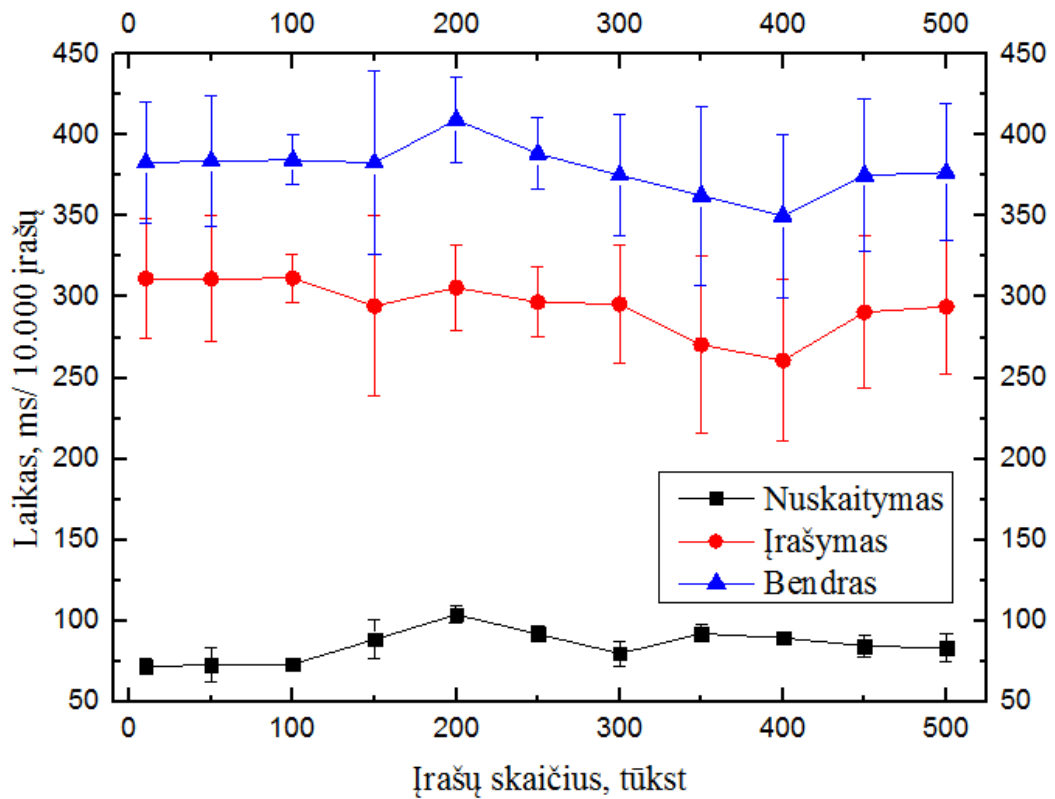
Bendros duomenų bazės integravimo būde užtikrinami 3 iš 5 saugumo komponentai. Kaip pavaizduota 4 lentelėje, integravimo būde užtikrinami autentifikavimas, autorizacija ir konfidencialumas. Vartotojo vardo ir slaptažodžio pora, naudojama bendros duomenų bazės integravime, neleidžia užtikrinti neišsigynimo. Integralumas užtikrinamas tik tada, kai viena sistema turi rašymo teises, taip apribojamos 5a pav. pavaizduoto integravimo būdo taikymo galimybės. 5b pav. pavaizduotame integravimo būde integralumas yra užtikrinamas, tačiau integracija tampa vienkryptė (integruojančios sistemos negali perduoti informacijos integruojamai sistemai). 5c pav. pavaizduotame integravimo būde integralumas taip pat užtikrinamas, tačiau šiame integravimo būde duomenų bazės kopijavimas padidina duomenų delną.

#### 2.2.2.4 Duomenų transliacijos laikas

Siekiant stabilaus sistemų veikimo, didėjant į duomenų bazę įrašomų duomenų kiekiui laikas, skirtas vienam įrašui apdoroti, turi išlikti pastovus. 2.1 poskyryje aprašytoje aplinkoje eksperimentiškai matuotas duomenų įrašymo ir nuskaitymo laikas į reliacinę duomenų bazę (PostgreSQL v9.0) įrašų skaičiui kintant nuo 10.000 iki 500.000. Prisijungimui prie duomenų bazės naudota 9.4-1203 JDBC 4 [Pos15] tvarkyklė. Prieš kiekvieną įrašymo matavimą duomenų bazės lentelė buvo išvaloma. Kiekvienam nuskaitymo matavimui duomenų bazės lentelėje buvo lygiai tiek įrašų, kiek reikia tam matavimui. Atlikus matavimus, išsaugoti duomenų įrašymo (nuskaitymo) pabaigos ir pradžios laikai. Apskaičiavus šių laikų skirtumą, gautas nuskaitymo ir įrašymo laikas, kurių suma yra duomenų transliacijos laikas.



21 pav. Duomenų transliacijos laiko priklausomybė nuo įrašų skaičiaus apdorojant tekstinius duomenis bendros duomenų bazės integravimo būdu



22 pav. Duomenų transliacijos laiko priklausomybė nuo įrašų skaičiaus apdorojant kompleksinius duomenis (tekstinio, skaičiaus ir loginės reikšmės rinkinio) bendroje duomenų bazėje

Atlikus eksperimentą nustatyta, kad duomenų transliacijos laikas intervale tarp 10.000 ir 500.000 yra pastovus. Apskaičiavus atliktų 1000 matavimų vidurkius, gautos duomenų transliacijos laiko priklausomybės, kurios pavaizduotos 21 pav. ir 22 pav. tirtame įrašų skaičiaus intervale duomenų transliacija nepriklauso nuo įrašų skaičiaus nei tekstiniams, nei kompleksiniams duomenims. Tarp 10.000 ir 500.000 tekstinio tipo duomenų transliacijos laikas yra pastovus ir vidutiniškai lygus  $237.70 \pm 21.79$  ms. Kompleksiniams duomenims tame pačiame intervale duomenų transliacijos laikas yra  $379.35 \pm 43.29$  ms.

### 2.2.2.5 Duomenų delsa

Duomenų delsa bendros duomenų bazės integravimo būde priklauso nuo pasirinktos realizacijos. 5a pav. ir 5b pav. pavaizduotiems integravimo būdams duomenų delsa  $t$  atsiranda dėl duomenų siuntimo iki duomenų bazės  $t_{ik}$ , duomenų siuntimo į integruojančią sistemą laiko  $t_{pars}$  ir duomenų įrašymo ir nuskaitymo laiko (transliacijos laiko)  $t_{tr}$ :

$$t = 2(t_{tr} + t_{pars} + t_{ik}) \quad (2.2.4)$$

Sistemas integruojant kaip pavaizduota 5c pav., be 2.2.4 formulėje esančių dėmenų, duomenų transliacijos laikas priklauso nuo intervalo  $\Delta t$ , po kurio sena duomenų bazės kopija pakeičiama atnaujinta:

$$t = \Delta t + t_{tr} + t_{pars} + t_{ik} \quad (2.2.5)$$

Didelėse duomenų bazėse duomenų kopijavimas yra labai ilgas ir todėl  $\Delta t$  išauga ( $\Delta t \gg t_{tr} + t_{pars} + t_{ik}$ ) ir nulemia delką:

$$t \approx \Delta t \quad (2.2.6)$$

Taigi integracijose, kuriose naudojama ta pati duomenų bazė, duomenų delką nulemia transliacijos laikas ir duomenų įrašymo ir nuskaitymo sparta, remiantis 2.2.4 formule, o integravimo būde, kai naudojama duomenų bazės kopija – kopijavimo intervalas.

### 2.2.3. Nuotolinio funkcionalumo iškvietimas

Nuotolinio funkcionalumo iškvietimas naudojamas atliekant duomenų, pranešimų arba procesų integraciją. Duomenų integracijos atveju aprašoma funkcija, kuri grąžina kitai sistemai reikalingus duomenis. Pranešimų integravime, integruojama sistema kaip funkcijos parametras priima pranešimą, kurį apdoroja ir grąžina integruojančiai sistemai atsaką. Procesų integravimas pasiekiamas sujungiant funkcijas į sekas ir aprašant jų kvietimo taisykles.

#### 2.2.3.1 Programinio kodo sudėtingumas

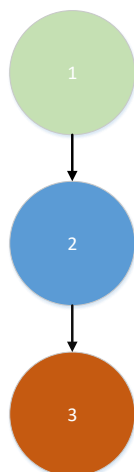
Jei integruojamoje sistemoje funkcija, kurią norima iškviešti jau egzistuoja, tai nereikalauja papildomo programinio kodo rašymo. Tokiu atveju reikia funkciją padaryti prieinamą integruojančioms sistemoms. Kai duomenų vaizdavimui naudojamos bibliotekos (pvz.: Gson, JAXB), programinio kodo kompleksškumas sumažėja, kadangi nereikia aprašyti duomenų vaizdavimo. Integruojančios ir integruojamų programų pseudokodas (pavaizduotas 23 pav. ir 24 pav.) yra paprastesnis jį lyginant su kodu, kurį reikia parašyti, kai vaizdavimas aprašomas programoje (pavaizduoti 8 pav. ir 10 pav.).

```
(1) gauti duomenis;  
(2) versti duomenis;  
(3) grąžinti rezultata;
```

23 pav. Integruojamos sistemos pseudokodas integruojant nuotolinio funkcionalumo iškvietimo integravimo būdu

```
(1) kviešti nuotolinę funkciją;  
(2) versti duomenis;  
(3) išeiti;
```

24 pav. Integruojančios sistemos programos pseudokodas integruojant nuotolinio funkcionalumo iškvietimo integravimo būdu



25 pav. Integruojančios ir integruojamos sistemų programos grafas integruojant nuotolinio funkcionalumo iškvietimo integravimo būdu

Tiek nuskaitančios, tiek įrašančios programos grafas yra identiškas, kaip pavaizduota 25 pav. Įrašančios ir nuskaitančios programų grafas turi  $N=3$  viršūnes ir  $E=2$  briaunas. Remiantis 2.2.1 formule, nuskaitančios ir įrašančios programų sudėtingumas yra:

$$M_{nus} = M_{įr} = 2 - 3 + 2 = 1.$$

Minėtu atveju, kai duomenų vaizdavimas atliekamas panaudojant trečiųjų šalių bibliotekas, integracijos programinio kodo sudėtingumas, remiantis 2.2.2 formule yra:

$$M = M_{nus} + M_{įr} = 2.$$

### 2.2.3.2 Sistemų sukibimas

Atliekant sistemų integravimą nuotolinio funkcionalumo iškvietimo būdu prieinamumas nėra užtikrinamas. Jei integruojama sistema nėra prieinama, integruojančioje sistemoje negalima užtikrinti, kad funkcionalumo iškvietimas bus įvykdytas. Kai funkcijos rezultatas negaunamas, integruojanti sistema yra neapibrėžtoje būsenoje ir nežino, ar iškviesta funkcija buvo įvykdyta. Sistema, nesulaukusi atsako, gali kviesti tą pačią funkciją dar kartą. Nors siekiama, kad integruojamoje sistemoje ši funkcija būtų įvykdyta tik kartą, ji gali būti įvykdyta kelis kartus. Todėl nuotolinio funkcionalumo iškvietime prieinamumas nėra užtikrinamas.

Kai sistemų integravimui naudojamos REST žiniatinklio paslaugos, užtikrinamas stabilus elementarių funkcijų rinkinys: PUT (atnaujinti), POST (įrašyti), PATCH (atnaujinti dalį), DELETE (ištrinti) ir GET (nuskaityti). Naudojant SOAP žiniatinklio paslaugas, stabili sąsaja nėra užtikrinama. Funkcijų aprašus pateikia integruojama sistema. Aprašytos funkcijos gali kisti, būti pašalintos, galima pridėti naujas funkcijas, todėl sistemų integravimo sąsaja gali kisti. Taigi stabili sąsaja yra užtikrinama REST, o neužtikrinamas SOAP žiniatinklio paslaugose.

Papildomumas nuotolinio funkcionalumo iškvietime yra užtikrinamas. Integruojama sistema gali laisvai pridėti naujas funkcijas, kurias integruojančios sistemos gali nuskaityti. REST žiniatinklio paslaugose keičiami prieinami resursai, kurių aprašas pateikiamas per HTTP OPTIONS metodą. SOAP žiniatinklio paslaugose integruojama sistema pateikia funkcijų aprašą naudodama WSDL. Šis aprašas gali būti versijuojamas, integruojamos sistemoms pasirenkant naudojamą versiją. Taigi papildomumas užtikrinamas REST ir SOAP žiniatinklio paslaugoms.

5 lentelė. Sistemų sukibimas nuotolinio funkcionalumo iškvietime

Sukibimo komponentas	Užtikrinamas
Prieinamumas	0
Stabili sąsaja	0-1
Papildomumas	1
<b>Iš viso:</b>	<b>1-2</b>

Atliekant nuotolinio funkcionalumo iškvietimo integravimą sistemos yra sukibusios, kadangi užtikrinamas tik 1 (REST žiniatinklio paslaugoms 2) iš 3 mažo sukibimo komponentų. Kaip pavaizduota 5 lentelėje, integravimo būde užtikrinamas tik papildomumas. Stabili sąsaja yra užtikrinamas tik naudojant REST žiniatinklio paslaugas, o prieinamumas nėra užtikrinamas ir sutrikus integruojamos sistemos veikimui, integruojanti sistema kviečiamą funkciją gali įvykdyti kelis kartus arba jos visai neįvykdyti.

### 2.2.3.3 Saugumas

Autentifikavimas nuotolinio funkcionalumo iškvietime yra užtikrinamas. Pradedant komunikaciją tarp sistemų abi sistemos autentifikuoja viena kitą. REST žiniatinklio paslaugose resursai prieinami, tik turint tam teisę, SOAP žiniatinklio paslaugose galima apriboti prieigą prie duomenų ir metodų. Taip užtikrinama autorizacija. Kai apsikeičiama šifruotais duomenimis, kuriuos gali skaityti tik tą galinti šalis (pavyzdžiui panaudojant viešojo rakto kriptografiją), užtikrinamas konfidencialumas. Taigi nuotolinio funkcionalumo iškvietime sistemų autorizacija ir autentifikavimas yra užtikrinami, o konfidencialumą galima užtikrinti panaudojant viešojo rakto kriptografiją.

Nuotolinio funkcionalumo iškvietime integralumas yra užtikrinamas. Integruojančios sistemos užklausoje pridedama kontrolinė suma, kurią patikrina integruojama sistema. Kai kontrolinė suma sutampa, integruojama sistema įvykdo kviečiamą funkciją, kitu atveju grąžina klaidą. Analogišku principu integruojanti sistema gali užtikrinti integruojamos sistemos atsako

integralumą. Kaip ir kituose integravimo būduose sistemų autentifikavimui naudojama vartotojo vardo ir slaptažodžio pora neužtikrina neišsigynimo.

6 lentelė. Saugumas nuotolinio funkcionalumo iškvietime

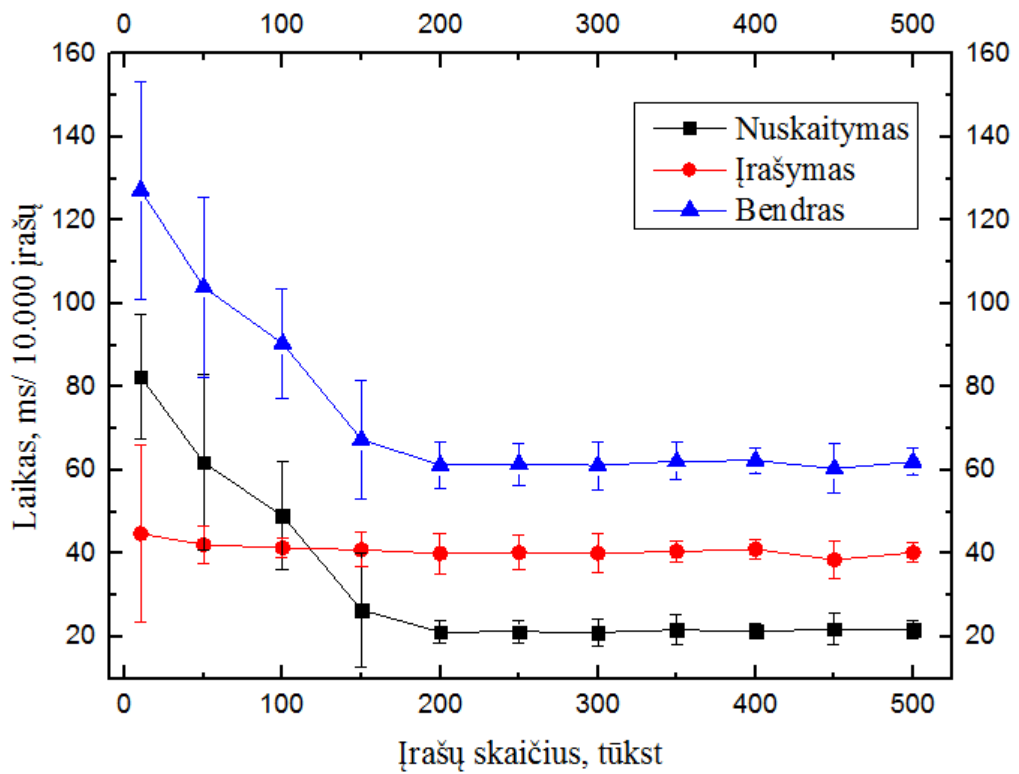
Saugumo komponentas	Užtikrinamas
Autentifikavimas	1
Autorizacija	1
Konfidencialumas	1
Integralumas	1
Neišsigynimas	0
<b>Iš viso:</b>	<b>4</b>

Nuotolinio funkcionalumo iškvietime saugumas yra užtikrinamas 4 iš 5 komponentų. Kaip pavaizduota 6 lentelėje, integravimo būde autentifikavimas ir autorizacija yra užtikrinami panaudojant vartotojo vardo ir slaptažodžio poras. Konfidencialumas yra užtikrinamas panaudojant viešojo rakto kriptografiją, o sistemų iškvietime ir atsake pridėjus kontrolines sumas užtikrinamas integralumas. Vartotojo vardo ir slaptažodžio pora, naudojama sistemų autentifikavimui neužtikrina neišsigynimo.

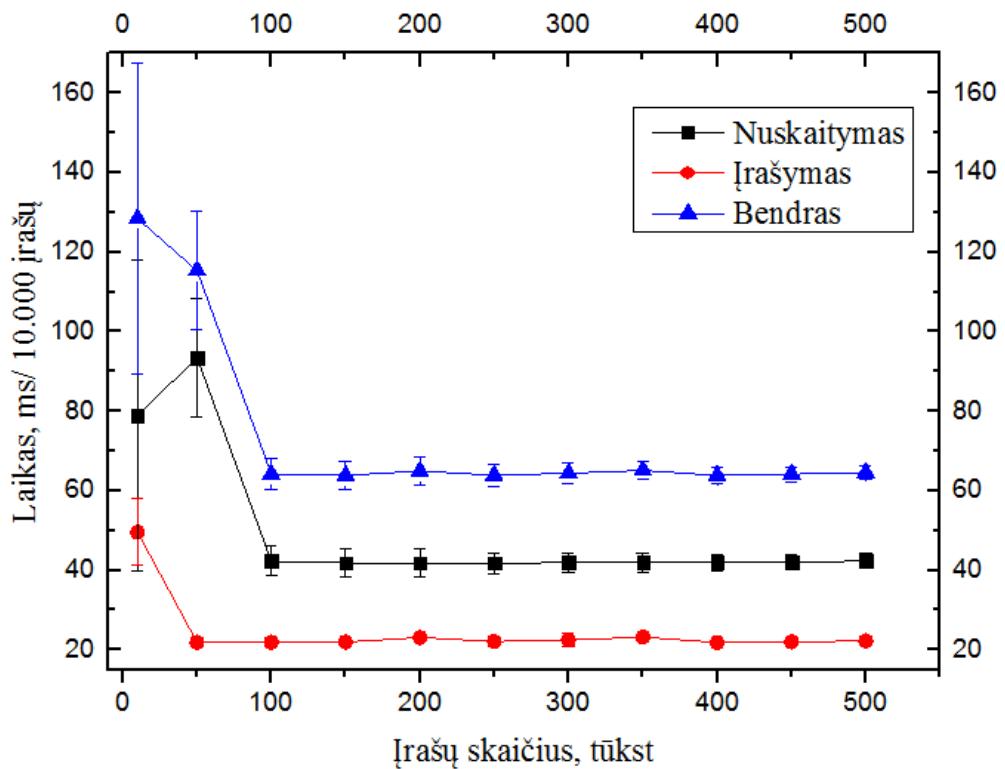
#### 2.2.3.4 Duomenų transliacijos laikas

Nuotolinio funkcionalumo iškvietime duomenų transliacijos laikas matuotas, kaip XML dokumento apdorojimas pasinaudojant JAXB klasė [JAX16] iš Java v1.8.0\_51. Transliacijos laikas matuotas įrašų viename dokumente skaičiui kintant nuo 10.000 iki 500.000. Matuotas duomenų įrašymo ir nuskaitymo laikas, gautas atėmus nuskaitymo ir įrašymo pradžios laiką iš pabaigos laiko. Šių laikų suma – duomenų transliacijos laikas. Atlikus 1000 matavimų gauti rezultatai, pavaizduoti 26 pav. ir 27 pav.





26 pav. Duomenų transliacijos laiko priklausomybė nuo įrašų skaičiaus apdorojant XML formato tekstinio tipo duomenis



27 pav. Duomenų transliacijos laiko priklausomybė nuo įrašų skaičiaus apdorojant XML formato kompleksinius (tekstinio, skaičiaus ir loginės reikšmės rinkinio) duomenis

Nustatyta, kad duomenų transliacijos ir duomenų nuskaitymo laikas nepriklauso nuo duomenų kompleksškumo. Didinant įrašų skaičių tarp 10.000 ir 150.000 nustatytas transliacijos laiko mažėjimas, kurį galėjo sąlygoti matavimo metodikos trūkumai. Tarp 150.000 ir 500.000 duomenų transliacijos laikas yra pastovus ir tekstiniams duomenims vidutiniškai lygus  $61.57 \pm 3.36$  ms. Kompleksiniams duomenims gauta, kad transliacijos laikas yra  $64.32 \pm 2.45$  ms. Nustatyta, kad duomenų transliacijos laikas nepriklauso nuo duomenų kompleksškumo, o tarp 150.000 ir 500.000 nepriklauso nuo duomenų kiekio.

### 2.2.3.5 Duomenų delsa

Nuotolinio funkcionalumo iškvietime duomenų delsa yra mažesnė lyginant su anksčiau minėtais integravimo būdais. Integravimo būde duomenų delsa atsiranda dėl duomenų transliacijos laiko  $t_{tr}$ , duomenų parsuntimo laiko  $t_{pars}$  ir duomenų įkėlimo laiko  $t_{ik}$ . Taigi bendru atveju duomenų delsa:

$$t = 2(t_{tr} + t_{pars} + t_{ik}) \quad (2.2.7)$$

Šio integravimo būdo delsa nepriklauso nuo procesų, kurie derinami tarp sistemų (failo įrašymo intervalų, duomenų bazės kopijavimo laiko). Dėl to duomenų delsa nuotolinio funkcionalumo iškvietime yra nedidesnė, nei anksčiau minėtuose integravimo būduose, kai duomenų dydis yra toks pats. Kai tinklai yra lėti (pvz.: toli geografiškai nutolusios sistemos), o duomenų transliacijos laikas mažas, duomenų delsa nulemia duomenų parsuntimo laikas:

$$t \approx 2(t_{pars} + t_{ik}) \quad (2.2.8)$$

## 2.2.4. Pranešimų integravimas

Pranešimų integravimas naudojamas atliekant duomenų, pranešimų arba procesų integraciją. Atliekant duomenų integraciją, integruojama sistema siunčia pranešimą su duomenimis integruojančios sistemoms. Siekiant atlikti pranešimų integravimą, integruojama sistema kitoms sistemoms skirtus pranešimus generuoja sistemoje įvykus įvykiams. Tarp sistemų suderinus pranešimų sekas, kaip ir nuotolinio funkcionalumo iškvietime, gaunamas procesų integravimas.

### 2.2.4.1 Programinio kodo sudėtingumas

Pranešimų integravimą galima realizuoti dviem būdais – nurodant pranešimo gavėją arba registracijos būdu. Tiek pranešimus siunčianti, tiek pranešimus nuskaityti sistemos aprašo paprastą programinį kodą, kaip pavaizduota 28 pav. ir 29 pav. Šiose programose nėra

atsižvelgiama į atvejį, kai prie pranešimų kanalo prisijungti nepavyksta. Be to, kaip ir nuotolinio funkcionalumo iškvietime laikoma, kad duomenų vertimui naudojamos trečiųjų šalių bibliotekos.

Programos grafe, kuris pavaizduotas 30 pav., yra  $N=3$  viršūnės ir  $E=2$  briaunos. Tiek nuskaitančios, tiek įrašančios programos sudėtingumas yra identiškas ir remiantis 5.1 formule lygus:

$$M_{nus} = M_{ir} = 2 - 3 + 2 = 1.$$

Remiantis 5.2 formule bendras integracijos programinio kodo sudėtingumas yra nuskaitančios ir įrašančios programų sudėtingumo suma:

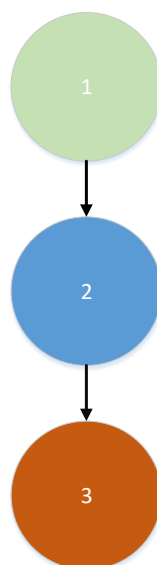
$$M = M_{nus} + M_{ir} = 2.$$

```
(1) prisijungti prie pranešimų kanalo();  
(2) nuskaityti pranešimą();  
(3) apdoroti pranešimą();
```

28 pav. Integruojamos sistemos pseudokodas pranešimų integravime

```
(1) vaizduoti pranešimo turinį();  
(2) prisijungti prie pranešimų kanalo();  
(3) patalpinti pranešimą();
```

29 pav. Integruojančios sistemos programos pseudokodas pranešimų integravime



30 pav. Integruojančios ir integruojamos sistemos programos grafas pranešimų integravime

### 2.2.4.2 Sistemų sukibimas

Pranešimų integravime prieinamumas yra užtikrinamas. Kai sutrinka pranešimus priimančios sistemos veikimas, nenuskaityti pranešimai kaupiasi ir yra apdorojami arba pašalinami šiai sistemai atstačius normalią veiklą. Tam gali būti naudojama strategija, kai užtikrinama, kad pranešimas bus gautas vieną kartą ir pranešimas bus gautas bent vieną kartą. Pranešimų siuntimas gali sutrinkti, sutrikus tarnybų magistralės veiklai. Siekiant to išvengti, galima naudoti antrinę tarnybų magistralę, kai pirminės veikla sutrinka. Taigi pranešimų integravime sistemų veikimas tarpusavyje nepriklauso, o sutrikimo metu prarasta informacija gali būti apdorota, todėl prieinamumas užtikrinamas.

Pranešimų integravimo elementarios funkcijų aibė yra apibrėžta ir nekintama. Integravimo būde yra dvi elementarios operacijos – išsiųsti pranešimą ir nuskaityti pranešimą. Pranešimų integravimo programinė įranga gali palaikyti ir kitas operacijas. ActiveMQ[ASF15], HornetQ[Jbo15] ir RabbitMQ[Rab15] taip pat palaiko visų pranešimų trynimą; OpenMQ[Jav15] ir ActiveMQ leidžia peržiūrėti pranešimus jų nenuskaitant. Nei viena programinė įranga neleidžia keisti elementarių funkcijų, todėl pranešimų integravime stabili sąsaja užtikrinama.

7 lentelė. Sistemų sukibimas pranešimų integravime

Sukibimo komponentas	Užtikrinamas
Prieinamumas	1
Stabili sąsaja	1
Papildomumas	1
<b>Iš viso:</b>	<b>3</b>

Pranešimų integravime sistemos nėra sukibusios, kadangi jame užtikrinami 3 iš 3 mažo sistemų sukibimo komponentai, kaip pavaizduota 7 lentelėje. Prieinamumas užtikrinamas tarnybų magistralei veikiant nepriklausomai nuo sistemų. Elementarių operacijų skaičius priklauso nuo pranešimų programinės įrangos, tačiau jų aprašas ir skaičius yra nekintantis. Papildomumas, kaip ir nuotolinio funkcionalumo iškvietime užtikrinamas per atnaujintą paslaugų aprašą.

### 2.2.4.3 Saugumas

Autentifikavimas pranešimų integravime yra užtikrinamas. Pranešimus siunčiančios ir priimančios sistemos autentifikuojasi naudodamos vartotojo vardo ir slaptažodžio porą. ActiveMQ, HornetQ ir RabbitMQ leidžia anoniminių vartotojų prisijungimą, tačiau siekiant užtikrinti autentifikavimą, svečių teisės turi būti apribotos. ActiveMQ, HornetQ ir RabbitMQ leidžia vartotojams suteikti nuskaitymo, įrašymo ir administratoriaus teises. Priklausomai nuo

naudojamos programinės įrangos vartotojai gali būti priskirti grupėms, kurioms priskiriamos teisės. Todėl pranešimų integravime autorizacija yra užtikrinama.

8 lentelė. Saugumas pranešimo integravime

Saugumo komponentas	Užtikrinamas
Autentifikavimas	1
Autorizacija	1
Konfidencialumas	1
Integralumas	1
Neišsigynimas	0
<b>Iš viso:</b>	<b>4</b>

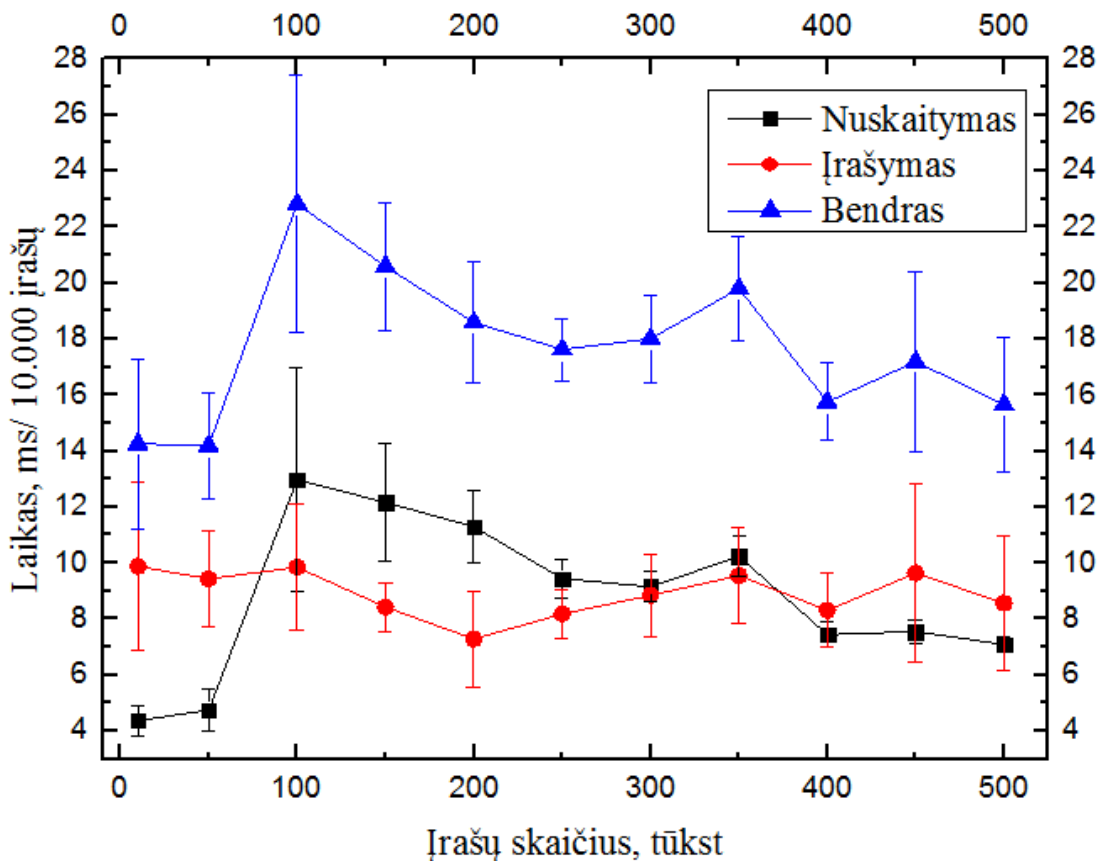
Pranešimų integravime konfidencialumas yra užtikrinamas. Kai pranešimų gavėjas gali būti bet kuri autentifikuota sistema, konfidencialumas užtikrinamas apribojant svečių teises. Kai pranešimas turi gavėją, užtikrinama, kad tik gavėjas gali perskaityti pranešimą. Integralumo užtikrinimas priklauso nuo pranešimų programinės įrangos. ActiveMQ integralumas užtikrinamas panaudojant SSL/TLS pranešimų magistralei autentifikuojantis į sistemas ir JAAS sistemoms autentifikuojantis į pranešimų magistralę. HornetQ integralumui užtikrinti remiasi AMQP[Oas12] standartu. Šį standartą galima taikyti naudojant ir kitą pranešimų programinę įrangą, taip užtikrinant integralumą. Neišsigynimas gali būti užtikrinamas pasinaudojant Morgan[MIP07] siūloma metodika, pranešimų perdavime panaudojant patikimas perdavimo stotis (angl. *reliable routing nodes*), kurios garantuoja pranešimų siuntėjo tapatybę ir tai, kad jis po to negalės išsižadėti savo veiksmo. Taigi pranešimų integravime užtikrinamas konfidencialumas ir integralumas, o neišsigynimą galima užtikrinti naudojant patikimas perdavimo stotis.

Pranešimų integravime saugumas yra užtikrinamas, kadangi užtikrinami 4 iš 5 saugumo komponentų. Saugumo užtikrinimas priklauso nuo naudojamos pranešimų programinės įrangos, ActiveMQ ir RabbitMQ reikia apriboti svečių prieigą prie pranešimų. Integralumas užtikrinamas HornetQ, naudojant kitą pranešimų programinę įrangą reikia laikytis AMQP standarto. Neišsigynimas pranešimų integravime nėra užtikrinamas, tačiau jį užtikrinti galima naudojant patikimas perdavimo stotis, kaip siūlo Morgan.

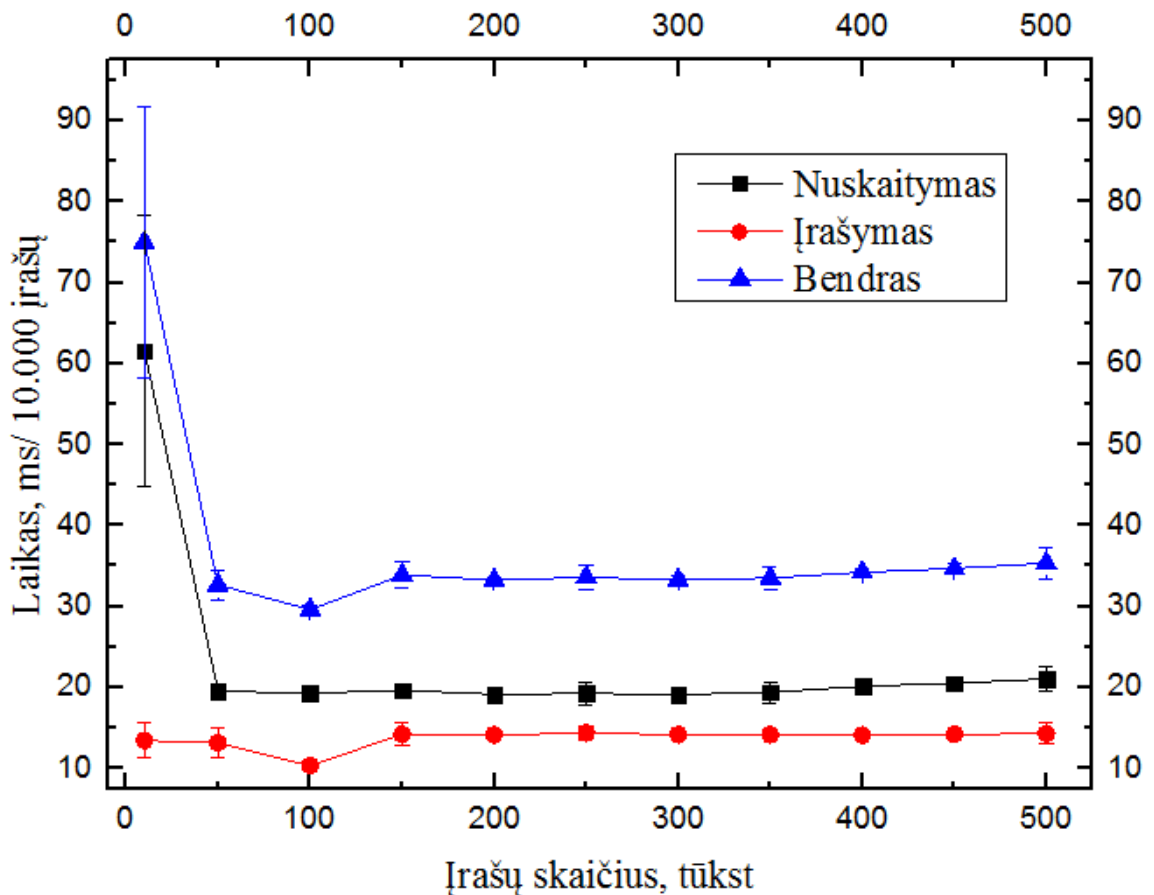
#### 2.2.4.4 Duomenų transliacijos laikas

Žinučių integravime 2.1 poskyryje aprašytoje aplinkoje duomenų transliacijos laikas matuotas, kaip JSON formato pranešimų apdorojimas pasinaudojant Gson v2.3.1[Gso16]

biblioteka. Transliacijos laikas matuotas intervale tarp 10.000 ir 500.000 įrašų viename dokumente. Išsaugotas laikas prieš nuskaitymą (įrašymą) ir laiką po jo, apskaičiuotas laikų skirtumas – nuskaitymo (įrašymo) laikas. Nuskaitymo ir įrašymo laikų suma – duomenų transliacijos laikas. Apskaičiavus atliktų 1000 matavimų rezultatų vidurkius, gautos priklausomybės, pavaizduotos 31 pav. ir 32 pav.



31 pav. Duomenų transliacijos laiko priklausomybė nuo įrašų skaičiaus apdorojant JSON formato tekstinio tipo duomenis



32 pav. Duomenų transliacijos laiko priklausomybė nuo įrašų skaičiaus apdorojant JSON formato kompleksinius (tekstinio, skaičiaus ir loginės reikšmės rinkinio) duomenis

Eksperimentiškai nustatyta, kad kompleksiniams duomenims duomenų transliacijos ir duomenų nuskaitymo laikas mažėja didinant įrašų skaičių tarp 10.000 ir 150.000. Tarp 150.000 ir 500.000 duomenų transliacijos laikas yra pastovus ir lygus  $33.49 \pm 1.22$  ms. Tekstiniams duomenims nenustatyta priklausomybė nuo įrašų skaičiaus. Intervale tarp 10.000 ir 500.000 vidutinis duomenų transliacijos laikas yra  $18.44 \pm 2.82$  ms. Duomenų transliacijos laikas priklauso nuo duomenų kompleksiskumo ir kompleksiniams duomenims jis beveik dvigubai didesnis. Lyginant su XML duomenų transliacija, JSON duomenų transliacijos laikas yra daugiau nei 3 kartus mažesnis tekstinio tipo duomenims ir beveik 2 kartus mažesnis kompleksiniams duomenims.

#### 2.2.4.5 Duomenų delsa

Pranešimų programinė įranga iš tarnybų magistralės pranešimus siunčia reikiamoms sistemoms. Šiuo aspektu pranešimų integravimas labiau primena nuotolinio funkcionalumo iškvietimą nei apsikeitimą failais, kadangi pastarajame integravimo būde sistema pati tikrina, ar failai nebuvo atnaujinti. Duomenų delsa nulemia duomenų transliacijos laikas  $t_{tr}$ , žinutės įkėlimo

laikas  $t_{ik}$  ir duomenų keliavimo tinklais laiko  $t_{pars}$ . Duomenų delsa pranešimų integravime lyginant jį su nuotolinio funkcionalumo iškvietimo yra didesnė dėl užlaikymo tarnybų magistralėje  $\Delta t$ . Bendriausiu atveju duomenų delsa:

$$t = 2(t_{tr} + t_{ik} + t_{pars}) + \Delta t \quad (2.2.9)$$

Kaip ir nuotolinio funkcionalumo iškvietime, jei tinklai yra lėti, duomenų transliacijos laikas ir užlaikymas tarnybų magistralėje yra maži, duomenų delsa priklauso nuo pranešimo parsisiuntimo ir įkėlimo laiko:

$$t = t \approx 2(t_{ik} + t_{pars}) \quad (2.2.10)$$



### 2.3. Integravimo būdų palyginimas

Integravimo būdų vertinimas, gautas 2.2 poskyryje, leidžia juos palyginti tarpusavyje. Kokybinės charakteristikos, saugumas ir sukibimas, lygintos pagal jų komponentų užtikrinimą kiekviename integravimo būdų. Kiekybinės charakteristikos, programinio kodo kompleksiskumas, plečiamumas, duomenų transliacijos laikas ir duomenų delsa, lyginamos pagal eksperimentiškai nustatytas absoliučias vertes. Charakteristikų vertinimo metu gauti rezultatai pavaizduoti 9 lentelėje.

9 lentelė. Integravimo būdų palyginimas

Integravimo būdas		Apsikeitimas failais	Bendra duomenų bazė	Nuotolinio funkcionalumo išskiriamas	Pranešimų integravimas
Charakteristika					
Programinio kodo sudėtingumas		7-9	3	2	2
Sistemų sukibimas	Prieinamumas	1	1	0	1
	Stabili sąsaja	0	1	0-1	1
	Papildomumas	0	0	1	1
Saugumas	Autentifikavimas	1	1	1	1
	Autorizacija	1	1	1	1
	Konfidencialumas	1	1	1	1
	Integralumas	1	0	1	1
	Neišsigynimas	0	0	0	0
Duomenų transliacijos laikas		25.95±2.45 67.79±4.33	237.70±21.79 379.35±43.29	61.57±3.36 64.32±2.45	18.44±2.82 33.49±1.22
Duomenų delsa		$\geq 2(t_{tr} + t_{pars} + t_{ik})$	$2(t_{tr} + t_{pars} + t_{ik})$	$2(t_{tr} + t_{pars} + t_{ik})$	$\geq 2(t_{tr} + t_{pars} + t_{ik})$

Gauti vertinimo rezultatai sunormuoti į vidurkį, kaip pavaizduota 10 lentelėje. Siekiant nuoseklumo, normuojant duomenų delsa, duomenų transliacijos laiką ir programinio kodo kompleksiskumą imtas atvirkštinis vidurkis. Tada normuotame vertinime, didesnė kiekvienos

charakteristikų vertė, reiškia geresnį integravimo būdo vertinimą. Kadangi duomenų delsa vertinimo rezultatas yra algebrinė išraiška, tai jis gali būti naudojamas tik tada, kai apskaitimas failais ar pranešimų integravimas lyginami su kitais integravimo būdais, kadangi jų duomenų delsa skiriasi nuo kitų integravimo būdų ir yra didesnė.

10 lentelė. Integravimo būdų sunormavus į vidurkį

Integravimo būdas Charakteristika	Apsikeitimas failais	Bendra duomenų bazė	Nuotolinio funkcionalumo iškvietimas	Pranešimų integravimas
Programinio kodo sudėtingumas	0.42-0.54	1.25	1.88	1.88
Sistemų sukibimas	0.5	1	1	1.5
Saugumas	1.07	0.8	1.07	1.07
Duomenų transliacijos laikas	3.31	0.36	1.40	4.66
	2.01	0.36	2.12	4.07
Duomenų delsa	$\leq 1$	1	1	$\leq 1$

Programinio kodo sudėtingumas integracijose priklauso nuo naudojamos programinės įrangos ir programavimo kalbų. Tirtu panaudos atveju, programinio kodo kompleksiškas didžiausias bendroje duomenų bazėje ir apskaitime failais. Apskaitime failais duomenų atvaizdavimui (angl. *mapping*) panaudojus trečiųjų šalių bibliotekas, programinio kodo sudėtingumas būtų analogiškas nuotolinio funkcionalumo iškvietimo ir pranešimų integravimo sudėtingumui.

Nagrinėti integravimo būdai sukuria skirtingą sukibimą tarp sistemų. Pranešimų integravimas sukuria mažiausią sukibimą tarp sistemų, kadangi užtikrina prieinamumą, plečiamumą ir stabilią sąsają. Nuotolinio funkcionalumo iškvietimas neužtikrina prieinamumo, kadangi sutrikus integruojamos sistemos veikimui, integruojančios sistemos kviečiama funkcija gali būti neįvykdyta arba įvykdyta kelis kartus. Bendroje duomenų bazėje duomenų modelio pakeitimai turi būti derinami su visomis sistemomis, todėl neužtikrinamas plečiamumas. Apskaitime failais nėra užtikrinama stabili sąsaja, kadangi apskaitimo failais procesas yra nestandartizuotas, o suderintas tarp sistemų. Taigi mažiausias sukibimas tarp sistemų yra pranešimų integravime.

Pranešimų integravime užtikrinami 4 iš 5 saugumo komponentų, tačiau saugumas pranešimų integravime nuo naudojamos pranešimų programinės įrangos. Pasinaudojus Morgan siūloma metodika ir naudojant patikimas tarnybines stotis, galima užtikrinti ir neišsigynimą. Neišsigynimas taip pat yra vienintelis neužtikrinamas komponentas nuotolinio funkcionalumo iškvietime. Bendroje duomenų bazėje neužtikrinamas neišsigynimas ir integralumas. Apsikeitime failais neišsigynimas gali būti užtikrintas tik panaudojus papildomą programinę įrangą. Taigi saugiausi integravimo būdai – nuotolinio funkcionalumo iškvietimas, apsikeitimas failais ir žinučių integravimas.

Duomenų transliacijos laikas priklauso nuo pasirinkto integravimo būdo ir duomenų formato. Eksperimentiškai nustatyta, kad didžiausia duomenų transliacija yra integruojant bendros duomenų integracijos būdu. Duomenų transliacija bendros duomenų bazės integravimo būde yra 11-13 kartų didesnė už transliaciją kitais nagrinėtais integravimo būdais. Naudojant JAXB XML duomenims apdoroti pastebėta, kad duomenų transliacijos laikas mažai priklauso nuo duomenų kompleksiskumo. Duomenų transliacija kompleksiškiems duomenims didesnė tik 1.04 karto, kai kitais tirtais atvejais transliacijos laikas pailgėjo nuo 1.6 iki 2.6 karto. Nepriklausomai nuo duomenų kompleksiskumo greičiausiai apdorojami JSON formato duomenys pranešimų integravime, nuotolinio funkcionalumo iškvietime arba failų integravime.

Duomenų delsa priklauso nuo pasirinkto integravimo būdo ir naudojamo duomenų formato. Kuo mažesnio dydžio yra failai, dalyvaujantys integravime, tuo mažesnė duomenų delsa. Apsikeitime failais, užtikrinant suderintą sistemų veikimą, kai iškart po failo įrašymo kita sistema failą nuskaity, duomenų delsa galima minimizuoti. Tai sukuria papildomą apkrovą sistemai, kadangi integruojanti sistema privalo nuolat tikrinti, ar failas nebuvo atnaujintas. Pranešimų integravime duomenų delsa padidina pranešimo užlaikymas tarnybų magistralėje. Bendros duomenų bazės ir nuotolinio funkcionalumo iškvietime duomenų delsa yra tokia pat. Kadangi bendros duomenų bazės integravimo būde duomenų transliacijos laikas yra didžiausias, jame didesnė ir duomenų delsa. Taigi mažiausia duomenų delsa yra nuotolinio funkcionalumo iškvietime.

#### **2.4. Integravimo scenarijų vertinimas**

Integravimo scenarijus, kaip ir integravimo būdus galima vertinti charakteristikomis. Izza[Izz09] darbe scenarijų vertinimui naudoja autonomiškumo, paskirstymo, lankstumo, dinamiškumo charakteristikas. Dinamiškumas – integracijos savybė greitai prisitaikyti prie pokyčių sistemose (pvz.: pridant naujas funkcijas ar plečiant duomenų modelį). Vertinant integravimo scenarijus dinamiškumas sujungtas su lankstumu – laiku, reikalingu pridėti vienetinei

funkcijai. Taigi integravimo scenarijų vertinimui šiame darbe naudojamos charakteristikos yra autonomiškumas, paskirstymas ir plečiamumas.

Vertinant integravimo būduose sukuriama sistemų sukibimą, vertintas pasiekiamumas. Vertinant integravimo scenarijus pasiekiamumas praplečiamas, įtraukiant stabilų sistemų veikimą ne tik sutrikus vienos iš sistemų veikimui, bet ir pasikeitus sistemoms. Tai reiškia, kad autonomiškas integravimo scenarijus yra tada, kai stabilus sistemų veikimas užtikrinamas praplėtus ar susiaurinus duomenų modelį (arba pakeitus palaikomas funkcijas). Taigi autonomiškas integravimo scenarijus užtikrina stabilų sistemų veikimą, pasikeitus vienai iš integruojamų sistemų.

Bendravimas tarp sistemų gali būti paskirstytas skirtingais būdais. Vertinant integravimo scenarijų paskirstymas (angl. *distribution*) nurodo, keliais skirtingais būdais galima perduoti integruojamus objektus tarp skirtingų sistemų. Integravimo objektai gali būti perduodami iš vienos sistemos į kitą gavus užklausa (angl. *request/response*) arba jie gali būti paskirstomi registracijos (angl. *publish/subscribe*) būdu. Taip pat objektai gali būti perduodami sinchroniškai ir asinchroniškai. Taigi integravimo scenarijus turi didesnę paskirstymą, jei integravimo objektų perdavimą leidžia užtikrinti daugiau skirtingų būdų.

Kai integruojamos sistemos po integracijos toliau vystomos, norint naudotis nauju funkcionalumu reikia vystyti ir integraciją. Dėl šios priežasties pasirinktas integravimo scenarijus turi palaikyti greitą naujo funkcionalumo pridėjimą esamoje integracijoje. Funkcionalumo pridėjimas apima naujos funkcijos pridėjimą integruojamos sistemos sąsajoje ir funkcijos panaudojimą integruojančioje sistemoje. Taigi lankstus integravimo scenarijus garantuoja greitą naujo funkcionalumo pridėjimą jau esamoje integracijoje.

#### **2.4.1. Vertinimo metodika**

Integravimo scenarijų vertinimas atliekamas subjektyviu autoriaus vertinimu. Charakteristikų vertinimui naudojama trijų lygių skalė (žemas, vidutinis, aukštas), kurią atitinka skaitinės vertės (0, 1, 2). Pasirinkti integravimo scenarijai yra vertinami atskirai kiekvienoje iš dimensijų pagal kiekvieną iš charakteristikų. Bendras integravimo scenarijaus vertinimas gaunamas sudėjus atskirus dimensijų vertinimus. Skaitinis vertinimas naudojamas integravimo scenarijui parinkti pagal integracijai keliamus reikalavimus.

Integravimo scenarijų vertinime lankstumas vertinamas kaip laikas, reikalingas pridėti vienetinį funkcionalumą (naują integravimo objektą – duomenis, pranešimą). Autonomiškumas vertinamas kaip stabilaus veikimo užtikrinimas pridėdant naują vienetinį funkcionalumą ar

pašalinant vienetinį funkcionalumą. Paskirstymas vertinamas kaip palaikomų integravimo objektų perdavimo būdų skaičius.

#### 2.4.2. Vertinimas

Pridėjus ar pašalinus vienetinį funkcionalumą semantinės integracijos atveju sistemų veikimas nepakinta, jei nepasikeičia ontologija. Jei reikia keisti ontologiją, stabilus integracijos veikimas nėra užtikrinamas iki ji bus pakeista. Dėl to semantinė integracija užtikrina vidutinį (1) autonomiškumą. Sintaksinės integracijos atveju, naujo funkcionalumo pridėjimas ar pašalinimas reikalauja pakeitimų abiejuose iš integruojamų sistemų, pridedant ar pašalinant funkcijas, keičiant duomenų modelį. Dėl to sintaksinės integracijos autonomiškumas yra žemas (0). Taigi sintaksinė integracija yra mažiau autonomiška nei semantinė integracija.

Vienetinio funkcionalumo pridėjimas duomenų integracijos atveju nedaro įtakos stabiliam sistemų veikimui. Tuo atveju, kai pašalinamas funkcionalumas iš integracijos, stabilus sistemų veikimas nebūtinai užtikrinamas, tačiau vienetinis pakeitimas duomenyse turi mažą įtaką sistemų stabilumui. Todėl duomenų integravimo stabilumas yra vidutinis (1). Pranešimų ir procesų integracijoje sistemų stabilumas priklauso nuo pranešimų apdorojimo ir procesų vykdymo. Todėl atliekant pranešimų (procesų) integraciją, vienetinio funkcionalumo pašalinimas iš integracijos turi didelę įtaką stabiliam sistemų veikimui. Pranešimų ir procesų integravimui autonomiškumas yra mažas (0). Taigi duomenų integracija yra autonomiškesnė negu pranešimų ir procesų integracija.

Paskirstymas priklauso nuo integravimo scenarijaus lygmens. Duomenų integracijos atveju, tiek sinchroniškai, tiek asinchroniškai galima perduoti duomenis pagal užklausą, tačiau nėra galimybės užtikrinti registracijos perdavimo būdą. Dėl to duomenų integracijos paskirstymas yra vidutinis (1). Procesų ir pranešimų integracijos atveju, sistemos apsieičia pranešimais, tada galima realizuoti sinchroninį ir asinchroninį užklausos ir registracijos perdavimą. Dėl to pranešimų ir procesų integravimas užtikrina aukštą (2) paskirstymą. Paskirstymas nepriklauso nuo integravimo lygio, dėl to laikysime, kad tiek semantinės, tiek sintaksinės integracijos paskirstymas yra aukštas (2). Taigi pranešimų ir procesų integracija yra geriau paskirstoma nei duomenų integracija.

Semantinės integracijos atveju sistemos yra lankstesnės, nes pasikeitimai sistemos duomenų modelyje neturi įtakos jų interpretavimui kitoje sistemoje. Nauji duomenys gali būti aptikti, jei tai nereikalauja pakeitimų ontologijoje. Dėl to semantinės integracijos lankstumas yra aukštas (2). Sintaksinės integracijos atveju, pridėta funkcija integruojamoje sistemoje gali būti pradėta naudoti

integruojančioje sistemoje tik po reikalingų pakeitimų. Dėl to sintaksinės integracijos lankstumas yra žemas (0). Taigi semantinė integracija yra lankstesnė negu sintaksinė integracija.

Lankstumas priklauso nuo integravimo scenarijaus lygmenis. Atliekant duomenų integraciją, pakeitimai, kuriuos reikia atlikti apima tik duomenų modelio pakeitimus. Pranešimų integracijos atveju, reikia suderinti ne tik pakeitimus duomenų modelyje, tačiau ir pranešimų nuskaitymo procedūrą. Procesų integravimas reikalauja pakeitimų abiejose sistemose, suderinant duomenų modelio pakeitimus, pranešimų nuskaitymo ir pranešimų sekų nuskaitymo procedūras. Lyginant integravimo lygmenis, gauname, kad duomenų integravimo lankstumas yra aukštas (2), pranešimų integravimo – vidutinis (1), o procesų integravimo – žemas (0).

11 lentelė. Integravimo scenarijų vertinimas

Charakteristika \ Scenarijus	Autonomiškumas	Paskirstymas	Lankstumas
Sintaksinė duomenų integracija	1 (0+1)	3 (2+1)	2 (0+2)
Sintaksinė pranešimų integracija	0 (0+0)	4 (2+2)	1 (0+1)
Sintaksinė procesų integracija	0 (0+0)	4 (2+2)	0 (0+0)
Semantinė duomenų integracija	2 (1+1)	3 (2+1)	4 (2+2)
Semantinė pranešimų integracija	1 (1+0)	4 (2+2)	3 (2+1)
Semantinė procesų integracija	1 (1+0)	4 (2+2)	2 (2+0)

Remiantis integravimo scenarijų vertinimu (pateiktu 11 lentelėje) galima parinkti integravimo scenarijų turint integracijai keliamus reikalavimus. Kai autonomiškumas, paskirstymas ir lankstumas yra vienodai svarbūs, semantinė duomenų integracija yra tinkamiausia. Kai paskirstymas yra svarbus, o lankstumas ir autonomiškumas nesvarbūs, tinkamiausia semantinė pranešimų integracija.

### 3. METODIKOS TAIKYMAS

#### 3.1. Reikalavimų siejimas su integravimo būdu ir scenarijumi

Atliktas integravimo scenarijų ir integravimo būdų vertinimas gali būti pritaikytas praktiškai. Pasinaudojant integracijai keliamais reikalavimais galima identifikuoti tinkamiausius integravimo būdus ir integravimo scenarijus. Reikalavimus vertinantis asmuo susieja kiekvieną integracijai keliamą reikalavimą su charakteristikomis. Reikalavimai, nedarantys įtakos integravimo būdo ar scenarijaus parinkimui, nėra siejami su nei viena charakteristika ir nėra naudojami vertinime. Kadangi reikalavimų siejimas su charakteristikomis yra subjektyvus ir priklauso nuo vertintojo, norint padidinti vertinimo tikslumą, jį gali atlikti daugiau negu vienas vertintojas, naudojant jų vertinimų vidurkį.

Atliekant reikalavimų ir charakteristikų siejimą pildoma reikalavimo ir charakteristikos ryšių matrica. Jeigu visi reikalavimai yra vienodo prioriteto (ar reikalavimų prioritetai nėra identifikuoti), laikoma, kad jis yra 1. Tuo atveju, kai reikalavimas yra siejamas su daugiau nei viena charakteristika, jo prioriteto vertė padalijama proporcingai charakteristikų skaičiui pagal formulę:

$$r = \frac{p}{n} \quad (3.1.1)$$

kur  $p$  – reikalavimo prioritetas,  $n$  – reikalavimo ryšių su charakteristikomis skaičius,  $r$  – charakteristikos ir reikalavimo ryšio koeficientas.

Atlikus integracijai keliamų reikalavimų siejimą su charakteristikomis, gaunamas kiekvienos charakteristikos svoris  $s$ , apskaičiuojamas pagal 9.2 formulę:

$$s = \sum_0^N r_i \quad (3.1.2)$$

kur  $N$  – reikalavimo skaičius,  $r_i$  –  $i$ -tojo reikalavimo ir charakteristikos ryšio koeficientas.

Charakteristikų svoriai toliau naudojami integravimo būdų ir scenarijų tinkamumui vertinti. Panaudojant integravimo būdo ir scenarijaus vertinimus (iš 10 lentelės arba 11 lentelės), gauname integravimo būdo ir scenarijaus tinkamumo koeficientus  $k_b$  ir  $k_s$ , apskaičiuojamus pagal formulę:

$$k = \sum_0^M s_i v_i \quad (3.1.3)$$

kur  $M$  – charakteristikų skaičius (būdams – 5, scenarijams – 3),  $s_i$  –  $i$ -tosios charakteristikos svoris,  $v_i$  –  $i$ -tosios charakteristikos įvertinimas (iš 10 lentelės arba 11 lentelės).

Pasinaudojant viešai prieinamais pirkimų duomenimis [VPK16] [VPK16a] [VPK16b], sistemų integravimui reikalavimai buvo apibendrinti ir sudarytas bendrų reikalavimų sąrašas.

Autorius, analizuodamas reikalavimus, identifikavo identiškus ir juos sujungė, sąrašą papildydamas tais, kurie buvo tik viename iš dokumentų. Gauti apibendrinti reikalavimai:

R1. Integracija turi palaikyti funkcinį ir technologinį išplėtimą iki bei suderinamumą su veiklos procesų valdymo sistemomis.

R2. Integracija turi realizuoti paslaugų maršrutizavimo funkciją.

R3. Integracija turi realizuoti tarpininkavimo funkcijas tarp integruojamų paslaugų.

R4. Integracija turi realizuoti paslaugų sąsajos (angl. *service gateway*) funkciją.

R5. Integracija turi palaikyti išorinius saugumo mechanizmus paslaugų autorizavimui bei autentifikavimui.

R6. Integracija turi palaikyti X (pvz.: HTTP, HTTPS, FTP, JMS) transporto protokolus.

R7. Integracija turi palaikyti asinchroninį, sinchroninį, registracijos transporto metodus.

R8. Integracija turi palaikyti federacinį darbo režimą su kitomis to paties gamintojo ir kitų gamintojų integracinėmis aplinkomis.

R9. Integracija turi palaikyti pranešimų maršrutizaciją pagal: iš anksto aprašytas taisykles, pranešimų turinį, siuntėją-gavėją.

R10. Turi būti galimybė vykdyti integracinius procesus lygiagrečiai.

R11. Integracija turi palaikyti SOA.

R12. Integracija turi būti lengvai plečiama papildant ją naujomis sistemomis.

R13. Duomenų integravimo komponento operatoriai ir administratoriai turi turėti galimybę stebėti ir kontroliuoti pasirinktų integracijų vykdymą.

R14. Turi būti galimybė perduoti struktūrizuotus ir nestruktūrizuotus duomenis.

R15. Turi būti galimybę informacija apsieisti tiek periodiškai, tiek pagal poreikį.

R16. Integracija turi būti realizuojama naudojant eilių mechanizmą.

R17. Žiniatinklio paslaugų funkcionalumai turi būti aprašyti WSDL.

R18. Integracija turi būti suderinama su XML ir XML žiniatinklio paslaugomis.

R19. Sistemos atsakymai turi būti pateikiami per ne ilgiau kaip X sekundžių.

R20. Integracijoje turi būti realizuotas veiksmų registravimo ir kontrolės mechanizmas (angl. *Audit trail*).

Šie reikalavimai susieti su charakteristikomis, kaip pavaizduota 12 lentelėje. Siejimą galima naudoti bet kokios integracijos reikalavimų vertinimui. Pirma, identifikuojami reikalavimai esantys pateiktame bendrame reikalavimų sąrašė ir integracijos reikalavimų specifikacijoje. Šiems reikalavimams iškart gaunamas siejimas su charakteristikomis. Antra, iš reikalavimų ir charakteristikų siejimo pašalinami reikalavimai, neegzistuojantys integracijos reikalavimų



specifikacijoje. Trečia, reikalavimų ir charakteristikų siejimo matrica papildoma apibendrintame sąraše neegzistavusiais reikalavimais ir atliekamas jų siejimas su charakteristikomis.

12 lentelė. Apibendrintų reikalavimų ir charakteristikų siejimo rezultatai. C1 – programinio kodo kompleksiškas, C2 – sukibimas tarp sistemų, C3 – saugumas, C4 – duomenų transliacijos laikas, C5 – duomenų delsa, C6 – autonomiškumas, C7 – paskirstymas, C8 – lankstumas.

Charakteristika \ Reikalavimas	C1	C2	C3	C4	C5	C6	C7	C8
	R1							
R2		$\frac{1}{2}$					$\frac{1}{2}$	
R3		$\frac{1}{2}$					$\frac{1}{2}$	
R4		$\frac{1}{3}$				$\frac{1}{3}$	$\frac{1}{3}$	
R5			1					
R6	$\frac{1}{2}$		$\frac{1}{2}$					
R7		$\frac{1}{3}$		$\frac{1}{3}$	$\frac{1}{3}$			
R8		$\frac{1}{2}$				$\frac{1}{2}$		
R9	$\frac{1}{2}$						$\frac{1}{2}$	
R10							1	
R11							1	
R12								1
R13		1						
R14	$\frac{1}{2}$							$\frac{1}{2}$
R15					1			
R16		$\frac{1}{2}$				$\frac{1}{2}$		
R17	1							
R18	1							
R19				$\frac{1}{2}$	$\frac{1}{2}$			
R20		1						

### 3.2. Reikalavimų ir charakteristikų siejimo pavyzdys

Siekiant parodyti praktinį atlikto vertinimo pritaikymą, pasirinkta integruota baudžiamojo proceso informacinei sistemai nustatyti tinkamą integravimo būdą ir integravimo scenarijų. Pasinaudojant sistemos reikalavimų specifikacija [VPK16] ir 12 lentelės duomenimis, atliktas integruotos baudžiamojo proceso informacinės sistemos reikalavimų vertinimas, siekiant nustatyti tinkamiausią integravimo būdą ir scenarijų. Reikalavimų specifikacija susiaurinta pasirenkant tik sistemų integravimui keliamus reikalavimus (kurie pateikti 1 priede). Gauti siejimo rezultatai pavaizduoti 13 lentelėje.

13 lentelė. Reikalavimų ir charakteristikų siejimo pavyzdys

Charakteristika Reikalavimas	C1	C2	C3	C4	C5	C6	C7	C8
R1								1
R2						$\frac{1}{2}$		
R3							1	
R4		$\frac{1}{2}$				$\frac{1}{2}$		
R5								
R6			$\frac{1}{2}$				$\frac{1}{2}$	
R7		1						
R8								
R9			1					
R10								
R11								
R12		$\frac{1}{3}$				$\frac{1}{3}$	$\frac{1}{3}$	
R13							1	
R14	1							
R15	$\frac{1}{2}$							$\frac{1}{2}$
R16								
R17		1						
R18	1							

13 lentelės tęsinys.

R19	1							
R20		1						
R21	1							
R22	1							
R23	1							
R24	1							
R25	1							
R26	1							
R27								
R28							1	
R29						1		

Pasinaudojant reikalavimų ir charakteristikų siejimo rezultatais, pavaizduotais 13 lentelėje, ir 3.1.2 formule apskaičiuoti charakteristikų svoriai integruotos baudžiamojo proceso informacinės sistemos integravimo reikalavimams. Gauti rezultatai pavaizduoti 14 lentelėje. Atlikus reikalavimų ir charakteristikų siejimą, nustatyta, kad svarbiausios charakteristikos – programinio kodo kompleksiskumas ir sistemų sukibimas, mažiausiai svarbios – duomenų transliacijos laikas ir duomenų delsa.

14 lentelė. Charakteristikų svoris

Identifikatorius	Charakteristika	Svoris
C1	Programinio kodo kompleksiskumas	9.5
C2	Sistemų sukibimas	4.33
C3	Saugumas	1.5
C4	Duomenų transliacijos laikas	0
C5	Duomenų delsa	0
C6	Autonomiškumas	2.33
C7	Paskirstymas	3.83
C8	Lankstumas	1.5

Gauti charakteristikų svoriai, pavaizduoti 14 lentelėje, panaudoti atliekant integravimo būdo ir integravimo scenarijaus tinkamumo vertinimą. Remiantis 3.1.3 formule apskaičiuotas

integravimo būdo tinkamumo koeficientas visiems integravimo būdams, kaip pavaizduota 15 lent. ir integravimo scenarijaus tinkamumo koeficientas, kaip pavaizduota 16 lentelėje.

15 lentelė. Integravimo būdų vertinimas. B1 – apsikeitimas failais, B2 – bendros duomenų bazės integravimas, B3 – nuotolinio funkcionalumo iškvietimas, B4 – pranešimų integravimas.

Būdas	Įvertinimas
B1	8.33
B2	17.41
B3	23.80
B4	25.96

16 lentelė. Integravimo scenarijų vertinimas. S1 – sintaksinė duomenų integracija, S2 – sintaksinė pranešimų integracija, S3 – sintaksinė procesų integracija, S4 – semantinė duomenų integracija, S5 – semantinė pranešimų integracija, S6 – semantinė procesų integracija

Scenarijus	Įvertinimas
S1	16.82
S2	16.82
S3	15.32
S4	22.15
S5	22.15
S6	20.65

Remiantis gautais rezultatais, nustatyta, kad integruoto baudžiamojo proceso informacinės sistemos integravimui tinkamiausias integravimo būdas pranešimų integravimas. Tai yra lauktas rezultatas, kadangi reikalavimų specifikacijoje R5 nurodo, kad integravimas turėtų būti atliekamas naudojant pranešimus. Tinkamiausias integravimo scenarijus – semantinė duomenų arba semantinė pranešimų integracija. Jei pasirenkamas integravimo būdas yra pranešimų, integravimo scenarijus turėtų būti semantinė pranešimų integracija.

# REZULTATAI IR IŠVADOS

## Rezultatai

1. Gauti integravimo būdų ir scenarijų vertinimai pagal pasirinktas charakteristikas.
2. Sukurta metodika integracijos reikalavimus susieti su integravimo būdų ir scenarijų vertinimo charakteristikomis.

## Išvados

1. Integravimo objektas, asinchroniškumas, specialistų susipažinimas su integravimo būdu, kaina, technologinis sudėtingumas yra netinkamos charakteristikos integravimo būdams vertinti, nes nepriklauso nuo integravimo būdo.
2. Atliekant išorinę sistemų integraciją, saugumas, sistemų sukibimas, duomenų delsa, duomenų transliacija ir programinio kodo sudėtingumas priklauso nuo integravimo būdo, todėl tinka jiems vertinti.
3. Atlikus integravimo būdų vertinimą nustatyta, kad išorinei sistemų integracijai:
  - a. apskaitimo failais integravimo būdas yra saugus, tačiau, lyginant su tirtais integravimo būdais, jame sukuriamą didžiausią duomenų delsa, rašomas sudėtingiausias programinis kodas, integruojamos sistemos stipriai sukibusios;
  - b. bendros duomenų bazės integravimo būdas, lyginant su tirtais integravimo būdais, sukuria mažesnę duomenų delšą, tačiau yra mažiau saugus, turi didžiausią duomenų transliacijos laiką ir sukuria stiprų sukibimą tarp integruojamų sistemų;
  - c. nuotolinio funkcionalumo iškvietimas, lyginant su tirtais integravimo būdais, yra saugus, reikalauja mažiau sudėtingo programinio kodo, tačiau integruojamos sistemos yra stipriai sukibusios;
  - d. pranešimų integravimas, lyginant su kitais integravimo būdais, yra saugus, reikalauja mažiau sudėtingo programinio kodo, sukuria silpną sukibimą tarp sistemų, tačiau turi didesnę duomenų delšą.
4. Atlikus integravimo scenarijų vertinimą nustatyta, kad, kai autonomiškumas, paskirstymas ir lankstumas yra vienodai svarbūs, tinkamiausia semantinė duomenų integracija; kai paskirstymas yra svarbus, o lankstumas ir autonomiškumas nesvarbūs, tinkamiausia semantinė pranešimų integracija.
5. Tinkamiausio integravimo būdo ir integravimo scenarijaus parinkimas tarpusavyje nepriklauso ir gali būti atliktas pagal reikalavimus, siejant juos su būdo ir scenarijaus vertinimo charakteristikomis.



## LITERATŪROS SĄRAŠAS

- [AM14] S. P. Ahuja ir N. Mupparaju, „Performance Evaluation and Comparison of Distributed Messaging Using Message Oriented Middleware“, *Computer and Information Science*, t. 7, nr. 4, pp. 9-19, 2014.
- [ASF15] „ActiveMQ“, Apache Software Foundation, [Tinkle]. Prieinamas: <http://activemq.apache.org/>. [Kreiptasi 28 05 2015].
- [Axe10] S. Axelsson, „Secure file transfer“. JAV Patentas US7707424 B2, 27 04 2010.
- [CGB02] K. Chiu, M. Govindaraju, R. Bramley. „Investigating the Limits of SOAP Performance for Scientific Computing“, *Proceedings of 11th. IEEE International Symposium on High Performance Distributed Computing HPDC-11 2002 (HPDC'02)*. Edinburgas, Škotija, pp. 246-254, 2002.
- [GY13] B. Glilai ir T. Yu, „B2B Integration: Bussiness Value and Adoption Trends“, *Global Suplly Chain Management Forum*, 2013.
- [Gso16] „Gson Java Doc“, Google Inc., [Tinkle]. Prieinamas: <https://google-gson.googlecode.com/svn/trunk/gson/docs/javadocs/com/google/gson/Gson.html> [Kreiptasi 05 01 2016]
- [HW00] G. Hohpe ir B. Woolf, *Enterprise Application Integration*, 2000.
- [Izz09] S. Izza, „Integration of industrial information systems: from syntactic to semantic integration approaches“, *Enterprise Information Systems*, t. 3, pp. 1-57, 2009.
- [Jav15] „Open Message Queue“, java.net, [Tinkle]. Available: <https://mq.java.net/>. [Kreiptasi 29 05 2015].
- [JAX16] „JAXB Java Doc“, Oracle, [Tinkle]. Prieinamas: <https://docs.oracle.com/javase/8/docs/api/javax/xml/bind/JAXB.html> [Kreiptasi 05 01 2016]
- [Jbo15] „HornetQ“, JBoss, [Tinkle]. Available: <http://hornetq.jboss.org/>. [Kreiptasi 29 05 2015].

- [KB12] E. Kaneshima ir R. T. V. Braga, „Patterns for Enterprise Application Integration,“ įtraukta *IX Latin America Conference of Pattern Languages of Programming*, 2012.
- [Kee02] W. Keech, „Secure file transfer method and system“. USA Patentas US20020095589, 02 07 2002.
- [KG14] R. Kukreja ir N. Garg, „Remote Procedure Call: Limitations and Drawbacks,“ *International Journal of Research*, t. 1, nr. 10, pp. 915-917, 2014.
- [Lin04] D. Linthicum, *Next generation application integration*, Boston: Addison-Wesley, 2004.
- [LMG+09] F. Lampataki, S. Mouzakitis, G. Gionis, Y. Charalabidis, D. Askounis, Business to Business interoperability: A current review of XML data standards, *Computer Standards and Interfaces*, nr. 31, 1045-1055 p., 2009.
- [LT03] B. Lublinsky ir D. Tyomkin, „Dissecting service oriented architectures,“ *Business Integration Journal*, pp. 52-58, 2003.
- [LVT+11] D. Litan, M. Velicanu, C. L., M. Teohari, A. M. Mocanu, I. Surugiu ir A. M. Raduta, „Business' New Requirement: Information Systems Integration - Methods and Technologies,“ *INTERNATIONAL JOURNAL OF COMPUTERS AND COMMUNICATIONS*, t. 5, nr. 3, pp. 133-145, 2011.
- [MB14] B. Medjahed ir B. Benatallah, „Business-to-business interactions: issues and enabling technologies,“ *The VLDB Journal*, pp. 59-85, 2003.
- [MB89] Th. J. McCabe, Ch. W. Butler, „Design complexity measurement and testing“, *ACM*, pp. 1415–1425, 1989.
- [Mic15a] „Microsoft RPC,“ Microsoft, [Tinkle]. Available: [https://msdn.microsoft.com/en-us/library/windows/desktop/ms691207\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms691207(v=vs.85).aspx). [Kreiptasi 15 05 2015].
- [Mic15b] „NET Remoting Overview,“ Microsoft, [Tinkle]. Available: [https://msdn.microsoft.com/en-us/library/kwdt6w2k\(v=vs.71\).aspx](https://msdn.microsoft.com/en-us/library/kwdt6w2k(v=vs.71).aspx). [Kreiptasi 15 05 2015].



- [MIP07] G.Morgan, D. Ingham, S. Parkin „A Message Oriented Middleware Solution Enabling Non-Repudiation Evidence Generation for Reliable Web Services,“ 4th International Service Availability Symposium Durham, NH, JAV, 2007.
- [Nur09] J. Nurmilaakso, „EDI-based and XML-based business-to-business integration: a statistical analysis,“ *International Journal of Business Information Systems*, t. 4, nr. 6, pp. 639-654, 2009.
- [Oas12] Oasis, Information technology — Advanced Message Queuing Protocol (AMQP) v1.0 Specification, 2012.
- [OMG15] „CORBA Directory,“ Object Management Group, [Tinkle]. Available: <http://corba-directory.omg.org/>. [Kreiptasi 15 05 2015].
- [Ora15] „Remote Method Invocation Home,“ Oracle, [Tinkle]. Available: <http://www.oracle.com/technetwork/articles/javaee/index-jsp-136424.html>. [Kreiptasi 15 05 2015].
- [Pos15] „PostgreSQL JDBC driver“, The PostgreSQL Global Development Group [Tinkle]. Prieinamas: <https://jdbc.postgresql.org/download.html> [Kreiptasi 2015-12-01]
- [PSK+12] P. K. Potti, A. S., U. K. ir P. Z., „Comparing Performance of Web Service Interaction Styles: SOAP vs. REST,“ įtraukta *Conference on Information Systems Applied Research* , New Orleans Louisiana, 2012.
- [PZL08] C. Pautasso, O. Zimmermann, F. Leymann „RESTful Web Services vs Big Web Services: Making the Right Architectural Decision,“ įtraukta *World Wide Web Conference* , Beijing China, 2008.
- [Rab15] „RabbitMQ documentation,“ Pivotal, [Tinkle]. Prieinamas: <https://www.rabbitmq.com/documentation.html> [Kreiptasi 2015-12-05].
- [SA09] R. M. Savola, H. Abie “Development of Measurable Security for a Distributed Messaging System”, *International Journal on Advances in Security*, t. 2 nr. 4, pp358-380, 2009.
- [Sin14] M. G. B. Singh, „E -Commerce Logistics: The New Wave,“ *International Journal of Multidisciplinary Approach*, pp. 105-111, 2014.

- [Spa15] „MANTARAY – OPEN SOURCE JMS,“ SPACEBUG, [Tinkle]. Available: [http://spacebug.com/mantaray\\_open\\_source\\_jms.html](http://spacebug.com/mantaray_open_source_jms.html). [Kreiptasi 24 08 2015].
- [Sto99] M. Stonebraker, „Integrating islands of information,“ *eAI Journal*,, p. 1–5, 1999.
- [VPK16] Integruota baudžiamojo proceso informacinė sistema, reikalavimų techninė specifikacija, viešojo pirkimo komisija, 2016.
- [VPK16a] Išmaniosios mokesčių administravimo sistemos elektroninio sąskaitų faktūrų ir elektroninių važtaraščių posistemių sukūrimo ir įdiegimo paslaugų techninė specifikacija, viešojo pirkimo komisija, 2016.
- [VPK16b] Informacinės sistemos „Avilys“ reikalavimų techninė specifikacija, viešojo pirkimo komisija, 2016.
- [WT12] K. Wagh ir R. Thool, „A Comparative Study of SOAP Vs REST Web Services Provisioning Techniques for Mobile Host,“ *Journal of Information Engineering and Applications*, t. 2, nr. 5, pp. 12-17, 2012.
- [Zor11] J. Zoroja, „INTERNET, E-commerce and e-government MEASURING THE GAP BETWEEN EUROPEAN DEVELOPED AND POST-communistic countries,“ *Interdisciplinary Description of Complex Systems*, pp. 119-133, 2011.

## PRIEDAI

1 priedas. Integruoto baudžiamojo proceso informacinės sistemos reikalavimai integracijai

- R1. Duomenų mainų platforma turi palaikyti funkcinį ir technologinį išplėtimą iki bei suderinamumą su veiklos procesų valdymo sistemomis.
- R2. Duomenų mainų platforma turi atlikti nepriklausomo nuo taikomųjų programų bei paslaugų transporto funkcijas.
- R3. Duomenų mainų platforma turi realizuoti paslaugų maršrutizavimo funkciją.
- R4. Duomenų mainų platforma turi realizuoti tarpininkavimo funkcijas tarp integruojamų paslaugų.
- R5. Duomenų mainų platforma turi realizuoti integraciją naudojant pranešimus.
- R6. Duomenų mainų platforma turi realizuoti paslaugų sąsajos (angl. *service gateway*) funkciją.
- R7. Duomenų mainų platforma turi realizuoti paslaugų virtualizacijos funkciją.
- R8. Duomenų mainų platforma turi palaikyti bylų apdorojimą ir perdavimą tarp skirtingų objektų.
- R9. Duomenų mainų platforma turi palaikyti išorinius saugumo mechanizmus paslaugų autorizavimui bei autentifikavimui.
- R10. Duomenų mainų platforma turi palaikyti HTTP, HTTPS, FTP, SNMP, SMTP, JMS transporto protokolus.
- R11. Duomenų mainų platforma turi palaikyti tiesioginį skaitymą ir rašymą iš/į DBVS.
- R12. Duomenų mainų platforma turi palaikyti asinchroninį, sinchroninį, registracijos (angl. *publish/subscribe*) transporto metodus.
- R13. Duomenų mainų platforma turi palaikyti federacinį darbo režimą su kitomis to paties gamintojo ir kitų gamintojų integracinėmis aplinkomis.
- R14. Kartu su duomenų mainų platforma turi būti pateikti elektroninio pašto, FTP, bylų, JDBC adapteriai.
- R15. Duomenų mainų platforma turi turėti galimybę ir priemones pačiam vartotojui kurti reikiamus adapterius.
- R16. Duomenų mainų platforma turi palaikyti pranešimų maršrutizaciją pagal: iš anksto aprašytas taisykles, pranešimų turinį, siuntėją-gavėją.
- R17. Duomenų mainų platforma turi palaikyti Service Component Architecture (SCA) standartą.
- R18. Duomenų mainų platforma turi palaikyti Service Data Object (SDO) standartą.
- R19. Duomenų mainų platforma turi palaikyti Java Conectivity Architecture (JCA) standartą.
- R20. Duomenų mainų platforma privalo užtikrinti ir garantuoti pranešimo pristatymą vieną ir tik vieną kartą (angl. *Once-and-only-once message delivery (XA)*).

- R21. Duomenų mainų platforma turi palaikyti XPath specifikaciją.
- R22. Duomenų mainų platforma turi palaikyti XSLT specifikaciją.
- R23. Duomenų mainų platforma turi palaikyti UDDI standartą.
- R24. Duomenų mainų platforma turi palaikyti Java EE 5, EJB 3.0, JDK 6 specifikacijas.
- R25. Duomenų mainų platforma privalo palaikyti XML, WSDL 1.1, XSD, SOAP 1.2, paslaugų specifikacijas.
- R26. Duomenų mainų platforma turi palaikyti WebServices standartus: WS-Security 1.1, WS-Notification, WS-Policy, WS-ReliableMessaging 1.1, WS-Trust 1.0, WS-Integration Basic Profile 2.0, WS-Integration Attachment Profile 1.1, WS-AtomicTransaction 1.0, WS-MetadataExchange, WS-Business Activity 1.0.
- R27. Duomenų mainų platforma turi palaikyti proxy funkcionalumą ir SOAP-HTTP spartinimą. Jei tam reikalinga papildoma programinė įranga ji turi būti pateikta.
- R28. Duomenų mainų platforma privalo palaikyti aukšto patikimumo priemonės bei dirbti lygiagrečiai keliuose tarnybinėse stotyse. Jei tam reikalinga papildoma programinė įranga, ji turi būti pateikta.
- R29. Duomenų mainų platforma turi palaikyti siūlomą virtualizacijos platformą tiek technologiškai, tiek pagal licencijavimo taisykles.