

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

VYGANDAS PLIASAS

VERSLAS VERSLUI UŽSAKYMŲ VALDYMO SISTEMŲ
ARCHITEKTŪRINIŲ SPRENDIMŲ TYRIMAS

Magistro baigiamasis darbas

Darbo vadovas
doc. dr. T. Blažauskas

KAUNAS, 2013

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

VYGANDAS PLIASAS

VERSLAS VERSLUI UŽSAKYMŲ VALDYMO SISTEMŲ
ARCHITEKTŪRINIŲ SPRENDIMŲ TYRIMAS

Magistro baigiamasis darbas

Darbo vadovas
doc. dr. T. Blažauskas
2013-05-16

Recenzentas
doc. dr. L. Motiejūnas
2013-05-16

Atliko
IFM-1/2 gr. studentas
Vygandas Pliasas
2013-05-16

KAUNAS, 2013

AUTENTIŠKUMO PATVIRTINIMAS

Patvirtinu, kad įteikiamas baigiamasis darbas „Verslas verslui užsakymų valdymo sistemų architektūrinių sprendimų tyrimas“:

1. autoriaus atliktas savarankiškai, jame nėra pateikta kitų autorių medžiagos kaip savos, nenurodant tikrojo šaltinio;
2. nebuvo to paties autoriaus pristatytas ir gintas kitoje mokymo įstaigoje Lietuvoje ar užsienyje;
3. nepateikia nuorodų į kitus darbus, jeigu jų medžiaga nėra naudota darbe;
4. pateikia visą naudotos literatūros sąrašą.

Vygandas Pliastas

(STUDENTO VARDAS, PAVARDĖ)

2013-05-16

(DATA)

(PARAŠAS)

SANTRAUKA

Tai yra verslas verslui užsakymų valdymo architektūrinių sprendimų tyrimas. Sukurtoje platformoje kompanijos gali komunikuoti tarpusavyje, sudaryti užsakymus, keistis bylomis bei dokumentais, ieškoti kitų paslaugų tiekėjų ir gauti informaciją apie juos. Pasitikėjimas tarp įmonių yra labai svarbus, o ši platforma padeda tai pasiekti. Bendravimas su klientais ir tiekėjais paprastai užima daug laiko ir reikalauja naudoti daug skirtingų įrankių, pavyzdžiui, el. pašta, telefoninius skambučius, *Skype*, ryšių su klientais valdymo sistemas ir t. t. Šio sprendimo idėja yra į vieną vietą sutraukti visus svarbiausius įrankius ir palengvinti kasdienį vadybininkų darbą. Aukštos kokybės paslaugų užtikrinimui reikalinga integracija su kitomis išorinėmis verslo valdymo sistemomis, o kad tai būtų pasiekta, reikalingos saityno tarnybos. Siekiant išsiaiškinti, kokios tarnybos ir jų konfigūracijos geriausiai tinka šiam tikslui pasiekti, buvo atliktas tyrimas ir eksperimentas, kurie aprašyti šiame dokumente.

Šį dokumentą sudaro trys pagrindinės dalys. Pirmojoje dalyje aprašoma galimų sprendimų analizė ir saityno tarnybų savybių tyrimas. Antrojoje dalyje aprašoma sukurta platforma, jos aktualumas ir funkcionalumas. Taip pat pateikiamos vartotojo sąsajos, panaudojimo atvejų schemos ir klasių diagrama. Šioje dalyje atsakoma į klausimus, kodėl vartotojai turėtų pasirinkti būtent šią platformą ir kokias problemas ji sukurta spręsti. Paskutiniuoju dalyje aprašomas eksperimentas, kuriuo buvo nustatomas labiausiai tinkantis protokolas ir duomenų formatas, pateikiami eksperimento rezultatai.

Siekiant nustatyti tinkamiausią saityno tarnybos konfigūracijos pasirinkimą, buvo sukurtos dvi papildomos programos – saityno tarnyba ir saityno tarnybos klientas. Kliento programa siuntė užklausas saityno tarnybai ir apdorojo gautus rezultatus. Gautiems duomenims buvo pritaikytas Gauso skirstinys ir iš grafikų matyti, kad visais eksperimento atvejais SOAP protokolas ir XML duomenų formatas buvo efektyviau apdorojamas.

SUMMARY

RESEARCH ON ARCHITECTURAL SOLUTIONS OF BUSINESS TO BUSINESS ORDERS MANAGEMENT SYSTEMS

This is a research on architectural solutions of business to business orders management systems. It's a B2B platform where different companies can communicate, exchange orders and documents and search for other providers. Trust building is important for all companies and this platform can help to make it happen. Communication between companies requires a lot of time and use of different tools, e.g. email, phone calls, Skype, file storage places, CRMs and so on. Idea is to implement all necessary equipment in one place and let managers to do their daily work easier. To provide high quality services it's important to ensure good communication between platform and outside business management systems. There was made a research for which are the best fitting solutions to be integrated as web services.

This document is consisted of three major parts. First part describes analysis of available solutions and research of web services technology. Second part describes developed project and its features. There is user interface, use case, data structures and competitors review. This is whole about project and why it would be good choice for our targeted audience. The last part describes the research that was done during this project. It was essential to determine which protocol and data models are most efficient.

For this project there were created two additional applications - one as web service and other one as a client. Client application sent requests to service and calculated its efficiency. For collected data there were calculated Gaussian distributions. By having those results we can evaluate efficiency of which protocol and data model is better treated on server side. Regardless the speed of internet connection SOAP protocol and XML data format was treated better all times in experiments.

TURINYS

Lentelių sąrašas	8
Paveikslų sąrašas	9
Terminų ir santrumpų žodynas	10
Įvadas	11
1. SAITYNO tarnybų analizė.....	12
1.1. Analizės tikslas	12
1.2. Saityno tarnybos.....	12
1.2.1. Saityno tarnybų savybės	13
1.2.2. Saityno tarnybų privalumai ir trūkumai	14
1.2.3. Kaip veikia saityno tarnybos.....	14
1.2.4. Saityno tarnybų standartai	16
1.2.5. Alternatyvūs duomenų perdavimo formatai	20
1.2.6. Saugumas	21
1.2.7. Saityno tarnybų tarpininkas	22
1.2.8. Saityno tarnybos verslo valdymo sistemose	22
1.3. Analizės išvados.....	23
2. UŽSAKYMŲ VALDYMO PLATFORMOS projektas	24
2.1. Įvadas	24
2.2. Projekto tikslas.....	24
2.3. Problemos sprendimas pasaulyje	26
2.4. Programų sistemos funkcijos	27
2.5. Bendri apribojimai	29
2.6. Projektavimo metodologijos ir technologijų analizė	29
2.7. Projekto apribojimai.....	29
2.7.1. Įpareigojantys apribojimai	29
2.8. Funkciniai reikalavimai	30
2.8.1. Veiklos sritis	30
2.8.2. Sistemos sudėtis.....	31
2.9. Reikalavimai duomenims.....	35
2.10. Klasių diagrama	36
2.11. Apibendrinančios išvados	36
3. Saityno tarnybų protokolų savybių tyrimas	37
3.1. Tikslas.....	37
3.2. Tyrimo nuostatos	37
3.3. Tyrimo metodika.....	38
3.4. Rezultatai ir jų analizė	40

3.4.1. Rezultatai	40
3.5. Analizē	42
3.6. Apibendrinimas	44
4. Rezultātu apibendrinimas ir i š vados	46
5. Literatūra	47

LENTELIŲ SĄRAŠAS

Lentelė 1. PA „Redaguoti vartotojų informaciją“	32
Lentelė 2. PA „Redaguoti komentarus“	32
Lentelė 3. PA „Ieškoti įmonių ir tiekėjų“	32
Lentelė 4. PA „Peržiūrėti įmonės informaciją“	32
Lentelė 5. PA „Rašyti komentarą apie įmonę“	32
Lentelė 6. PA „Pateikti užsakymą“	33
Lentelė 7. PA „Dalyvauti derinant užsakymą“	33
Lentelė 8. PA „Patvirtinti užsakymą“	33
Lentelė 9. PA „Pildyti įmonės informaciją“	33
Lentelė 10. PA „Redaguoti savo profilį“	33
Lentelė 11. PA „Gauti užsakymą“	34
Lentelė 12. PA „Pažiūrėti informaciją apie klientą“	34
Lentelė 13. PA „Publikuoti savo įmonės puslapį“	34

PAVEIKSLŲ SĄRAŠAS

Paveikslas 1. Saityno tarnybos koncepcinė diagrama.....	13
Paveikslas 2. Saityno tarnybos dalys	15
Paveikslas 3. Saityno tarnybos naudojimo schema.....	16
Paveikslas 4. Saityno tarnybos tarpininko iliustracija	22
Paveikslas 5. Užsakymų valdymo platformos užsakymų peržiūros lango karkasinis modelis	27
Paveikslas 6. Sistemos veikėjai.....	28
Paveikslas 7. Kontekstinė diagrama.....	30
Paveikslas 8. Panaudos atvejų diagrama	31
Paveikslas 9. Duomenų bazės struktūra.....	35
Paveikslas 10. Sistemos klasių diagrama	36
Paveikslas 11. Kliento programos langas	38
Paveikslas 12. Proceso iliustracija	39
Paveikslas 13. Bandymo su 100 užklausų rezultatai.....	40
Paveikslas 14. Bandymo su 1000 užklausų rezultatai.....	40
Paveikslas 15. Bandymo su 10000 užklausų rezultatai.....	41
Paveikslas 16. Bandymo su 15x100 užklausų rezultatai.....	41
Paveikslas 17. Bandymo su 100 užklausų rezultatų Gauso skirstinys.....	42
Paveikslas 18. Bandymo su 1000 užklausų rezultatų Gauso skirstinys.....	42
Paveikslas 19. Bandymo su 10000 užklausų rezultatų Gauso skirstinys.....	43
Paveikslas 20. Bandymo su 15x100 užklausų rezultatų Gauso skirstiniai	43
Paveikslas 21. REST/JSON ir SOAP/XML rezultatų Gauso skirstiniai.....	44
Paveikslas 22. REST/JSON ir SOAP/XML rezultatų diagrama. Matomi trukmės šuoliai, kurie reiškia, kad sistema užtrunka labai ilgai apdorodama duomenis	45

TERMINŲ IR SANTRUMPŲ ŽODYNAS

- ADS (*Advertisement and Discovery of Services (IBM)*) – saityno tarnybų reklamavimo ir paieškos technologija;
- API (*Application Programming Interface*) – programinės įrangos programinė sąsaja;
- B2B (*Business-to-Business*) – verslo verslui sąveika;
- CORBA (*Common Object Request Broker Architecture*) – bendra objektų sąveikos architektūra;
- DISCO (*Discovery of Web Services (Microsoft)*) – saityno tarnybų paieškos technologija;
- HTML (*Hypertext Markup Language*) – hipertekstinių žymių kalba;
- HTTP (*Hypertext Transfer Protocol*) – hipertekstinių duomenų perdavimo protokolas;
- JAXM (*Java API for XML Messaging*) – Java programinė sąsaja XML žinučių apsikeitimui;
- LB – Lietuvos bankas;
- MS – Microsoft;
- QoS (*Quality-of-Service*) – kokybės užtikrinimas;
- RMI (*Remote Method Invocation*) – nutolusio metodo iškvietimas;
- RPC (*Remote Procedure Call*) – nutolusios procedūros iškvietimo metodas;
- SOAP (*Simple Object Access Protocol*) – paprastas objektų sujungimo protokolas;
- UDDI (*Universal Description, Discovery and Integration*) – universali saityno tarnybų aprašymo, suradimo ir integravimo technologija;
- W3C (*World Wide Web Consortium*) – pasaulinio tinklo konsorciumas;
- WSDL (*Web Services Definition/Description Language*) – saityno tarnybų aprašymo kalba;
- WSIL (*Web Services Inspection Language*) – saityno tarnybų suradimo ir integravimo technologija;
- WWW (*World Wide Web*) – pasaulinis tinklas;
- XML (*Extensible Markup Language*) – išplėsta žymių kalba.

IVADAS

Užsakymų valdymo platformos uždavinys – optimizuoti užsakymų valdymo procesą, taupyti laiką ir resursus derinant užsakymus tarp klientų ir tiekėjų. Trumpai tariant, sistema perkelia popierinius dokumentus į elektroninę erdvę, komunikaciją padaro patogesnę, padeda darbuotojams lengviau surasti aktualią informaciją apie užsakymus, taupo įmonės resursus dokumentų, *media* elementų ir kitų bylų archyvavimui bei padeda analizuoti nukrypimus nuo terminų ir, žinoma, pardavimų efektyvumą.

Kadangi šios sistemos vartotojai yra įmonės, o jos dažniausiai turi įsidiegusios kokius nors verslo valdymo sprendimus, būtina išspręsti užsakymų valdymo sistemos integraciją su jau esamomis kliento sistemomis.

Taip pat tikėtina, kad bus kuriamos papildomos trečios šalies aplikacijos šiai užsakymų valdymo platformai. Tam reikalinga saityno tarnyba, kuri leistų pasiekti reikiamą užsakymų valdymo platformos informaciją iš išorės, t. y. nenaudojant internetinio tinklapio sąsajos.

Integraciją patogiausia atlikti naudojant saityno tarnybas (angl. *web services*), kurių pagalba visiškai skirtingos sistemos gali komunikuoti ir keistis duomenimis.

Saityno tarnyba – tai tarsi atskira programa, kuri nėra priklausoma nuo paties projekto. Ji gali būti parašyta naudojant kitą programavimo kalbą ir veikti kitokioje operacinėje sistemoje. Svarbiausia, kad būtų nustatytas vienodas komunikacijos protokolas besikreipiančioje sistemoje ir saityno tarnyboje.

Mainai tarp saityno tarnybų tinka įvairioms verslas verslui (angl. *B2B*) sistemoms, visoms vienodai reikalingas saugus bendravimas, dinamiškumas ir stabili integracija. Kompanijos turi priimti bendrą sprendimą, kuris numatytų technologijas, naudojamas įmonių viduje, biudžetą ir sinchronizacijos lygį, reikalingą verslo veiksniumi. Šie sprendimai nulems technologijas, kurių reikės procesams įgyvendinti.

Šiame darbe aprašoma kuriama sistema, jos poreikis rinkai, komponentai, taip pat atliekamas tyrimas, parodantis, koks protokolas ir duomenų formatas leidžia pasiekti geriausių rezultatų integruojant verslo sistemas.

1. SAITYNO TARNYBŲ ANALIZĖ

Kuriant projektą pastebėta, kad įvairios verslo valdymo sistemos gali būti glaudžiai susiejamos naudojant saityno tarnybas, todėl nuspręsta projekto architektūrą grįsti saityno tarnybomis. Tai svarbu šio tipo platformai, nes toks išdėstymas, kai atskiros projekto dalys ir funkcionalumas iškeliami į saityno tarnybas, leis lengviau tobulinti ir vystyti projektą ir suteiks palankesnes galimybes jį integruoti su kitomis sistemomis, skirtomis verslo valdymui, ryšiams su klientais arba apskaitai.

Šiame skyriuje analizuojamos saityno tarnybos ir jų veikimo principas. Kad tyrimas būtų atliktas tinkamai, privalu tiksliai žinoti, su kuo reikės dirbti.

1.1. Analizės tikslas

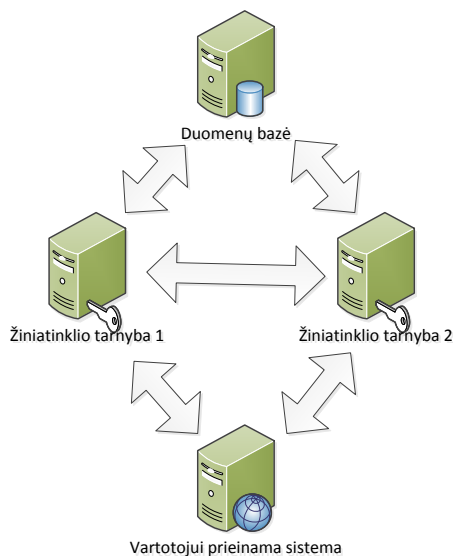
Nusprendus saityno tarnybas integruoti į užsakymo valdymo platformą, kilo poreikis nustatyti, kokios saityno tarnybų savybės ir kokia konfigūracija yra tinkamiausia norint tikėtis geriausių rezultatų. Šiam tikslui pasiekti visų pirma reikia įsigilinti į tai, kaip veikia pačios saityno tarnybos, kokias funkcijas galima atlikti ir kokių rezultatų galima tikėtis.

1.2. Saityno tarnybos

Saityno tarnybos yra interneto taikomųjų programų rūšis. Saityno tarnybos gali vykdyti tiek paprastas užklausas, tiek sudėtingas operacijas, susijusias su verslo valdymu. Saityno paslaugos kuriant internetines programas sparčiai plinta. Tokias saityno tarnybas galima trumpai apibūdinti kaip funkcijas ar objektus, veikiančius internete, kuriuos galima naudoti savo programose panašiai kaip įprastines funkcijas ar objektus [1].

Saityno tarnybos yra programos, skirtos atskirų sistemų komunikavimui užtikrinti ir duomenų mainams atlikti. Tarnybos gali būti naudojamos duomenų gavimui, saugojimui arba kokiems nors darbams atlikti juos pasiekiant per nuotolį. Saityno tarnybos naudojamos vis dažniau, nes jos gali būti plačiau pritaikomos ir užtikrinamas didesnis saugumas dėl jau integruoto saugumo mechanizmo. Praktikoje pasitaiko, kad kuriant internetinį portalą verslo logika iškeliamą į saityno tarnybą, o pats portalas turi tik duomenų gavimo, saugojimo ir atvaizdavimo funkcionalumą.

Toliau pateikiama saityno tarnybų koncepcinė diagrama, kurioje vaizduojama, kaip tokios sistemos gali sąveikauti tarpusavyje ir su skirtingomis sistemomis (pav. 1):



Paveikslas 1. Saityno tarnybos koncepcinė diagrama

Paveikslėlyje (pav. 1) pavaizduotos dvi saityno tarnybos, duomenų bazė ir vartotojo naudojama sistema. Sistema kreipiasi į kelias tarnybas pagal reikalingus duomenis arba pageidaujamas operacijas, tuo tarpu tarnybos gali komunikuoti tarpusavyje ir operuoti duomenimis duomenų bazėje.

1.2.1. Saityno tarnybų savybės

Saityno tarnybos gali būti daug kartų naudojamos tiek tam pačiam projektui, tiek ir daugeliui skirtingų projektų, kuriems reikalingas panašus funkcionalumas ir duomenys. Programuotojams nereikia rašyti daugybės skirtingų ir naujų informacijos apdorojimo tarnybų arba metodų. Tokios tarnybos leidžia greičiau ir kokybiškiau įgyvendinti reikiamas užduotis.

Programiniai komponentai nėra tvirtai susieti vieni su kitais, o tai leidžia juos lengvai valdyti ir koordinuoti. Programinį nesusietumą suprantame, kai tarnyba ir projektas nėra būtinais kuriami naudojant tą pačią programavimo kalbą arba jie nebūtinai turi būti talpinami toje pačioje operacinėje sistemoje. Pagrindinis projektas gali būti kuriamas .NET pagrindu ir talpinamas Windows serveryje, tuo tarpu mums reikalinga saityno tarnyba galėtų būti parašyta naudojant PHP programavimo kalbą ir patalpinta Linux šeimos operacinėje sistemoje.

Saityno tarnybos gali būti prieinamos viešai arba privačiai. Viešos tarnybos pavyzdys galėtų būti Lietuvos banko valiutų kursų saityno tarnyba, kur visi gali sužinoti dabartinį kursą. Privati tarnyba skirta kokio nors konkretaus ar kelių projektų naudojimui, prieinama tik VPN klientams ir konkrečioms IP adresams. Saityno tarnybos dažniausiai pasiekiamos naudojant šiuos protokolus: HTTP/HTTPS, SMTP, FTP/SFTP, TCP/UDP, *pipe* (komunikacija tarp procesų).

1.2.2. Saityno tarnybų privalumai ir trūkumai

Saityno tarnybų naudojimas yra santykinai paprastas, nes pagrįstas saityno technologijomis ir nereikalauja atskiro komunikacijos protokolo kūrimo. HTTP protokolas leidžia lengvai pasiekti saityno tarnybas ir patikrinti jų veikimą. Tai galima padaryti, pavyzdžiui, į naršyklę įdiegus įskiepi, skirtą saityno tarnyboms kviesti. Saityno tarnybomis galima naudotis visur, kur yra interneto prieiga.

Saityno tarnybos nėra priklausomos nuo naudojamos platformos, operacinės sistemos ir programavimo kalbos. Dažniausiai pasitaikantis duomenų formatavimas – XML. Tereikia išlaikyti vienodą paketų formatą, o projekto įgyvendinimo kalbos gali būti bet kokios, pavyzdžiui, C, C++, C#, PHP, Ruby On Rails, Python, Perl ir t. t. [2]

Tokia projekto skaidymo koncepcija yra artimesnė objektinio programavimo supratimui, nes atskiriami objektai ne tik sistemos viduje, bet ir pati sistema suskaidoma į atskiras nepriklausomas dalis.

Saityno tarnybas palaiko praktiškai visos populiarios verslo valdymo sistemos. Akivaizdu, kad tokia paslauga yra reikalinga ir kompanijos dažnai naudoja keletą tokio tipo sistemų. Pavyzdžiui, viena VVS įdiegta seniau, tada atsiranda naujesnė taikomoji sistema ir jas abi reikia integruoti, sinchronizuoti duomenis ir kuo labiau sumažinti rankinį darbą.

Kai saityno tarnybos yra kuriamos verslo valdymo ir operacijų funkcionalumui automatizuoti, viena iš aktualiausių problemų yra jų saugumas. Kadangi duomenys keliauja internetu, jie gali būti prieinami programišiams, taigi reikalingas duomenų šifravimas.

Sistemų veikimo sparta, kai naudojamos saityno tarnybos, yra tiesiogiai proporcinga interneto greitaveikai. Tai reiškia, kad turint lėtą internetą teks ilgai laukti duomenų arba kol bus įvykdytas numatytas darbas.

Projektą išskaidžius, jo dalys dažnai būna talpinamos skirtingose stotyse, o tai reiškia, kad reikia prižiūrėti daugiau serverių. Vienam jų sustojus gali sutrikti sistemos funkcionalumas. Kita vertus, sistema iš dalies vis dar funkcionuos.

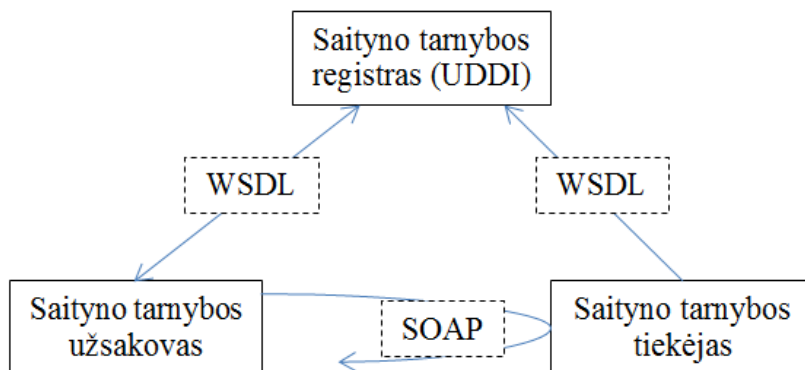
1.2.3. Kaip veikia saityno tarnybos

Saityno tarnybą sudaro šios dalys:

- saityno tarnybos tiekėjas – publikuoja tam tikras saityno tarnybas ir užtikrina jų funkcionalumą;
- saityno tarnybos registras – saugo informaciją apie saityno tarnybas bei tarpininkauja tarp saityno tarnybų tiekėjų ir saityno tarnybų užsakovų;
- saityno tarnybos užsakovas – bendradarbiauja su saityno tarnybų registrais, ieškodamas tinkamų saityno tarnybų bei jas integruodamas su savo turimomis.

Saityno tarnybų tiekėjai registruojasi kaip verslo vienetai, publikuoja savo saityno tarnybą registre ir reikalui esant atnaujina tiek verslo, tiek konkrečios saityno tarnybos informaciją. Tarnybų užsakovai tinkamos saityno tarnybos ieško registre. Kai nustatoma tinkama saityno tarnyba, parenkami būdai, kaip bus su ja bendraujama ir apsiukeičiama reikalinga informacija.

Toliau pateikiama schema, kuri iliustruoja saityno tarnybos veikimą (pav. 2) [1]:



Paveikslas 2. Saityno tarnybos dalys

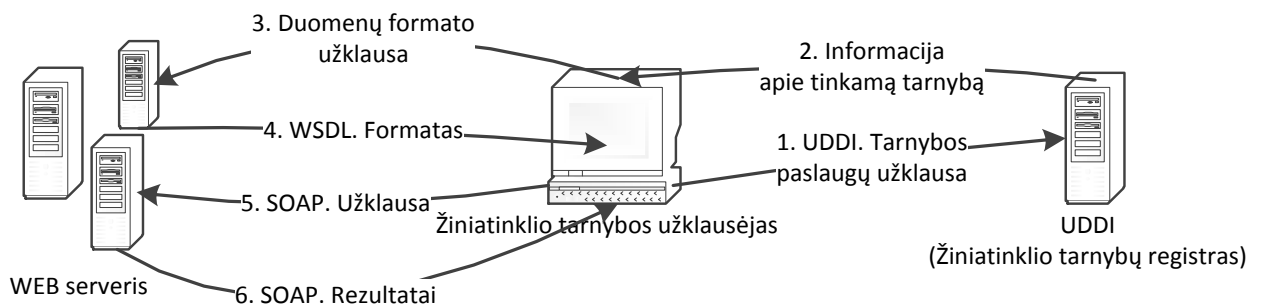
Saityno tarnybos trys pagrindinės funkcijos:

- Publikavimas / Pašalinimas
 - Tai yra saityno tarnybų įrašymas į saityno tarnybų registrą.
 - Tai daroma UDDI – universalios saityno tarnybų aprašymo, paieškos bei su ja susijusių verslo procesų integruotumo užtikrinimo technologijos pagalba. Jie taip pat gali būti pašalinti iš registro. Saityno tarnybų publikavimą ar šalinimą užsako saityno tarnybų struktūrinis elementas – tiekėjas, o veiksmus atlieka tarpininkas.
- Paieška
 - Šią operaciją atlieka saityno tarnybų užsakovai ir saityno tarnybų registrai.
- Sujungimas
 - Operacija yra atliekama tarp užsakovo ir tiekėjo. Saityno tarnybos užsakovas gali prisijungti prie tiekėjo siūlomos tarnybos ar jos dalies, naudodamasis tam tikromis sujungimo taisyklėmis.

Šioms operacijoms atlikti naudojamos skirtingos technologijos.

- SOAP – vokas, į kurį dedami ir transportuojami duomenys. Tai yra svarbiausia dalis, nes apjungia tarnybas;
- WSDL – saityno tarnybos sąsajos aprašas;
- UDDI – vieta, kur saugoma informacija apie saityno tarnybą.

Toliau pavaizduota, kaip vykdoma komunikacija naudojant prieš tai minėtas technologijas (pav. 3):



Paveikslas 3. Saityno tarnybos naudojimo schema

1.2.4. Saityno tarnybų standartai

1.2.4.1. UDDI

UDDI technologijos pagalba galima surinkti ir saugoti informaciją apie saityno tarnybas. Organizacijoms siūloma sistema, suteikianti galimybę aprašyti savo verslą ir nuosavas saityno tarnybas bei surasti kitų organizacijų siūlomas tarnybas. UDDI registras – tai tokia pati saityno tarnyba, teikianti struktūrinę informaciją apie kitas saityno tarnybas. UDDI registro realizacija turėtų leisti prisijungti prie jo dviem skirtingais būdais.

1. Paprasčiausias prisijungimas, kai atidaroma nuoroda interneto naršyklėje. Toks prisijungimas yra gana paprastas būdas, leidžiantis tarnybos tiekėjui publikuoti informaciją apie jo verslą ir teikiamą tarnybą, o tarnybos vartotojui leidžiantis lengvai surasti publikuojamą informaciją.
2. Programinis prisijungimas naudojant UDDI API specifikaciją. Šis metodas leidžia prisijungti prie UDDI registro kaip prie saityno tarnybos – tuomet informacijos publikavimas arba paieška vyksta dinamiškai, realiu laiku. Programinio prisijungimo metu, norint susisiekti su UDDI registru ir atlikti publikavimo ar paieškos operacijas, naudojami atitinkami UDDI API komandiniai pranešimai.

UDDI specifikacija sudaryta iš kelių susijusių dokumentų ir WSDL kalba parašytos aprašomosios schemas, kur yra apibrėžiamos SOAP protokolu paremtos saityno tarnybų registracijos ir atradimo taisyklės.

UDDI registras yra logiškai centralizuota ir fiziškai paskirstyta saityno tarnyba su daugybe šakninių elementų, kurie tarpusavyje pasikeičia duomenimis. Kai tik verslo elementas užregistruoja savo informaciją registre, ji yra automatiškai paskirstoma tarp atskirų šakninių elementų ir tampa visiems plačiai prieinama. UDDI registruose saugoma informacija susideda iš trijų pagrindinių komponentų:

- baltųjų puslapių – juose pateikiama kompanijos kontaktinė informacija;
- geltonųjų puslapių – juose kompanijų veikla yra suskirstyta pagal standartinę sistematiką;
- žaliųjų puslapių – juose pateikiama konkrečių tarnybų techninė dokumentacija.

Yra du pagrindiniai modeliai, nusakantys UDDI registrų panaudojimą:

- vieši registrai;
- privatūs registrai.

Tiek viešų, tiek privačių UDDI registrų vidinė informacijos struktūra tokia pati, skiriasi tik prieigos prie registro lygis ir apsaugos priemonių panaudojimas. Egzistuoja keletas privačių UDDI registrų tipų, kurie gali būti naudojami priklausomai nuo kompanijos ir saityno tarnybų pritaikymo tipo.

- Partnerių katalogo UDDI registras. Saityno tarnybos, kurios yra skirtos tam tikros kompanijos naudojimui, gali būti publikuojamos partnerių katalogo UDDI registre.
- Įmonės vidinių saityno tarnybų UDDI registras. Saityno tarnybos, naudojamos įmonės vidaus užduotims ar taikomųjų programų integravimui, turėtų būti publikuojamos atitinkamos rūšies UDDI registre. Šio registro veiklos sritis gali būti tiek atskira taikomoji programa, tiek viso departamento, kompanijos ar verslo procesų seka. Šio tipo registrai neapsaugoti ugniasiene ir tarnybų publikuotojams leidžia labiau kontroliuoti jų tarnybų registraciją ir pasiekiamumą.
- El. prekyviečių UDDI registras. Saityno tarnybų, kurios konkuruoja rinkoje, aprašymai turėtų būti skelbiami e-prekyviečių UDDI registre arba UDDI Verslo registre. Tokio tipo UDDI registrai daugiausia kuriami ir prižiūrimi atskirų verslo šakų standartų tvirtinimo organizacijų ar konsorciumų ir saugo detalią tarnybas teikiančių verslo vienetų informaciją. Šio tipo registrai garantuoja tam tikrą bendrai nustatytų standartų griežtą laikymąsi, tam tikrą saityno tarnybų sąsają, metaduomenimis grįstos paieškos galimybes ir užtikrina bendrą saugomų tarnybų kokybę.
- Portalo UDDI registras. Saityno tarnybos, skirtos kompanijos išoriniams partneriams, turi būti publikuojamos portalo tipo UDDI registre. Portalo UDDI registrai egzistuoja arba tiekėjo ugniasienės išorėje, arba DMZ zonoje (tarp dviejų ugniasienių). Ši privačių UDDI registrų rūšis talpina tik tų tarnybų aprašymus, kuriuos kompanija nori teikti išorės partnerių tarnybų užklausejams. Tai leidžia kompanijoms kontroliuoti tarnybų aprašymus, priėjimą prie UDDI registrų ir tarnybų kokybę. Be to, įmonė gali riboti ir tarnybų aprašymų matomumą partneriams, kuriems suteikta teisė prieiti prie jų.

Privatus UDDI registras yra uždaresnio tipo, skirtas daugiau įmonės vidaus reikmėms arba tam tikros verslo šakos įmonių grupei. Kita vertus, tokiu būdu jis garantuoja jame saugomos

informacijos patikimumą, saityno tarnybos kokybę ir veiksmų, atliekamų su registru, geresnį valdymą, lyginant su viešu UDDI registru [3].

1.2.4.2. WSDL

Saityno tarnybų aprašymo kalba WSDL yra pagrįsta XML technologija ir leidžia sudaryti katalogus bei aprašyti saityno tarnybas. WSDL yra naudojama komunikavimo proceso tarp saityno tarnybų automatizavimui. WSDL aprašo saityno tarnybos sąsają, prisijungimo prie saityno tarnybos protokolą (SOAP) ir galutinę tarnybos buvimo vietą [4].

WSDL aprašymų dokumentas gali saugoti tiek visą saityno tarnybą, tiek jo atskirą dalį aprašančią informaciją. Dokumente operacijos ir žinutės yra aprašomos abstrakčiai ir susiejamos su konkrečiu tinklo protokolu ir žinutės formatu. Tarpusavyje susiję komponentai apjungiami į abstraktesnius elementus. WSDL turi išplėstas atskirų tarnybų bei jų žinučių aprašymo galimybes atsižvelgiant į tai, kokie žinučių formatai ir tinklo protokolai naudojami. Dažniausiai WSDL kalba naudojama kartu su SOAP ir HTTP protokolais.

WSDL dokumentai skirstomi į du tipus – tarnybos sąsajos ir tarnybos įdiegimo. WSDL tarnybos sąsajos dokumente yra saugomas tarnybos aprašymas, kuris bus naudojamas įdiegiant vieną ar daugiau tarnybų. Tai yra abstraktus saityno tarnybos aprašymas, kuris naudojamas aprašant specifinį tarnybos tipą. Serviso sąsajos dokumentas gali nukreipti į kitą tarnybos sąsajos dokumentą naudodamas *import* elementą. Tarnybos diegimo dokumente gali būti nuorodos ne į vieną tarnybos sąsajos dokumentą. Tarnybos sąsajos dokumentas yra kuriamas ir publikuojamas tarnybos sąsajos tiekėjo. Tarnybos diegimo dokumentą kuria ir publikuoja tarnybos tiekėjas [5].

WSDL dokumentai saitynų tarnybų aprašymui naudoja šiuos elementus:

- **types** – konteineriai, skirti duomenų tipų aprašymams, naudojant tam tikrą sistemą;
- **message** – abstraktus komunikuojančių duomenų aprašymas;
- **operation** – abstraktus saityno tarnybos atliekamų veiksmų aprašymas (įėjimo, išėjimo parametrai);
- **portType** – operacijų, kurias sugeba atlikti vienas ar kitas baigtinis tarnybos komponentas, abstrakčių aprašymų aibė;
- **binding** – konkreti protokolo ar duomenų formato specifikacija tam tikram prieigos tipui (**portType**);
- **port** – atskiras baigtinis tarnybos komponentas, apibūdinamas kaip sąryšio ir tinklo adreso kombinacija;
- **service** – aprašo saityno tarnybą kaip **port** ar **endpoint** elementų kolekciją.

1.2.4.3. SOAP

SOAP pranešimą sudaro trys pagrindinės dalys: vokas, antraštė ir pranešimo vidinė informacinė (pagrindinė) dalis, apgaubta atitinkamo tinklo protokolo pranešimo su atitinkama protokolo antrašte.

Protokolo antraštė – tai standartinė transporto protokolo antraštė, naudojama SOAP pranešimo pristatymo informacijos saugojimui. Kokia bus ši antraštė, priklauso nuo konkretaus transporto protokolo. Pavyzdžiui, jeigu SOAP pranešimas siunčiamas per HTTP, tada *Content-Type* laukas nustatomas į „*text/xml*“, o *SOAPAction* laukas saugo saityno tarnybos, kurios pranešimas perduodamas, URL adresą. Laukas *SOAPAction* leidžia pranešimus priimančiai saityno tarnybai užfiksuoti įeinantį SOAP pranešimą ir jį nukreipti į reikiamą saityno tarnybą.

SOAP vokas yra pagrindinis ir būtinas kiekvieno SOAP pranešimo elementas. Jis apibrėžia likusioje pranešimo dalyje naudojamų vardų aibę. Voke būtinai turi būti voko kodavimo taisyklių rinkinį nurodantis elementas *xmlns:SOAP-ENV* (SOAP Voko vardų aibė), taip pat *xmlns:xsi* (XML Schemas reikalavimų vardų aibė) ir *xmlns:xsd* (XML Schemas duomenų tipų vardų aibė).

SOAP antraštė yra neprivalomas elementas, kuris reikalingas komunikuojančių programų specifinės pagalbinės informacijos apsikeitimui. Taigi galimi antraštės elementai neapibrėžiami SOAP specifikacijoje. Jeigu SOAP pranešime yra antraštė, tai ji seka iš karto po voko. Antraštėje taip pat gali būti *mustUnderstand* atributas, kuris leidžia gavėjui nusiųsti klaidos pranešimą, jeigu gavėjo programos nepalaiko vienos ar daugiau antraštėje nurodytų charakteristikų.

SOAP informacinė dalis XML formatu atvaizduoja informaciją, kuria keičiamasi. Informacinė dalis yra privaloma SOAP pranešime. Jei nėra antraštės, ši dalis seka iš karto po SOAP voko.

Pagrindinis reikalavimas saityno tarnybų tiekėjui ir tarnybos užsakovui yra gebėjimas kurti ir sintaksiškai analizuoti SOAP pranešimus bei komunikuoti tinklo pagalba. Paprastai SOAP serveris, veikiantis saityno programų serveryje, atlieka minėtas funkcijas. Gali būti naudojama tam tikros programinės kalbos biblioteka, užtikrinanti minėtas funkcijas [6].

Komunikacija tarp programų naudojant SOAP protokolą vyksta 4 etapais.

1. Tarnybos užsakovo kliento taikomoji programa naudoja vietinį XML sintaksės analizatorių ir tarnybos *proxy* programą XML dokumento sukūrimui ir SOAP pranešimo generavimui. Šis pranešimas yra užklausa, kuri iškviečia tarnybos tiekėjo tiekiamą saityno tarnybą. *Proxy* programa šį pranešimą kartu su tarnybos tiekėjo adresu persiunčia SOAP kliento aplinkai. SOAP klientas atsakymo pranešimą grąžina užklausejui.
2. Tinklo infrastruktūra pristato pranešimą tarnybos tiekėjo SOAP palaikančiam serveriui. SOAP serveris gautą pranešimą persiunčia adresatui. SOAP aplinka yra atsakinga už XML tipo pranešimų konvertavimą į programinės kalbos objektus pagal kliento programos

reikalavimus. Šis konvertavimas atliekamas remiantis kodavimo schemomis, kurios yra SOAP pranešime.

3. Saityno tarnybos programinė įranga, atsakinga už atsakymo į SOAP pranešimo užklauso generavimą. Šis pranešimas yra persiunčiamas saityno tarnybos užsakovui.
4. Kai tarnybos užsakovo *proxy* programa gauna atsakymo pranešimą, jis yra nukreipiamas į SOAP serverį tam, kad būtų išanalizuotas atsakymo XML dokumentas. Galiausiai atsakymo pranešimas pristatomas kliento programai.

1.2.5. Alternatyvūs duomenų perdavimo formatai

JSON – tekstinis, žmogui suprantamas duomenų formatas, atvaizduojantis duomenų struktūras ir asociatyvius masyvus. JSON dažniausiai naudojamas kartu su *Ajax* technologija duomenims perduoti ir yra naudojamas kaip XML alternatyva [7].

Pats paprasčiausias pavyzdys:

```
{
  "vardas": "Jonas Jonaitis",
  "adresas": {
    "gatve": "Savanorių g. 3 - 5",
    "miestas": "Kaunas",
    "salis": "Lietuva",
  },
  "telefonas": [
    "8 624 15000",
    "8 672 34928"
  ]
}
```

JSON viduje gali būti atvaizduoti paprasti kintamieji, asociatyvus masyvas, paprastas masyvas.

Toliau pateikiamas JSON ir XML to paties objekto aprašymas:

JSON:

```
{
  "vardas": "Jonas Jonaitis",
  "adresas": {
    "gatve": "Savanorių g. 3 - 5",
    "miestas": "Kaunas",
    "salis": "Lietuva",
  },
  "telefonas": [
    "8 624 15000",
    "8 672 34928"
  ]
}
```

```
XML:
<pagrindinis>
<vardas> Jonas Jonaitis </vardas>
<adresas>
  <gatve> Savanorių g. 3 - 5</gatve>
  <miestas>Kaunas</miestas>
  <salis>Lietuva</salis>
</adresas>
<telefonas>8 624 15000</telefonas>
<telefonas>8 672 34928</telefonas>
</pagrindinis>
```

1.2.6. Saugumas

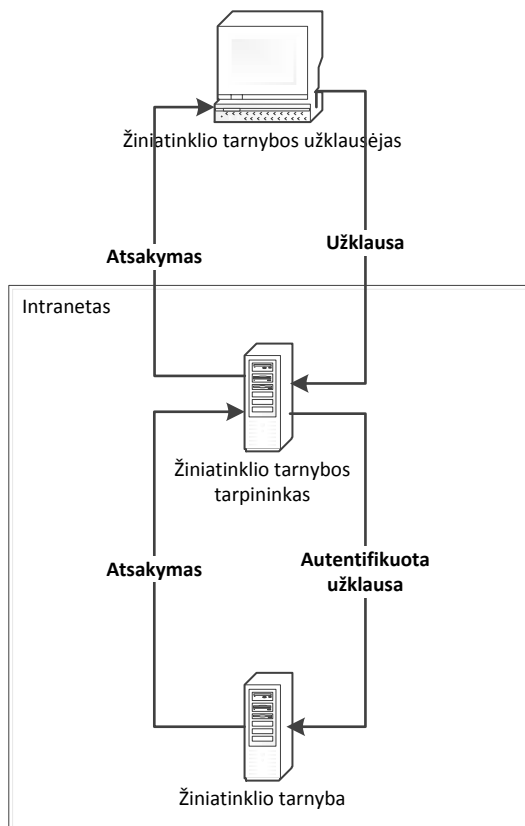
Vienas iš ryškiausių saityno tarnybų minusų – nesugebėjimas užtikrinti tarnybų kokybės (*Quality-of-Service QoS*) reikalavimų, kas yra ypatingai svarbu mainų procesui, ypatingai verslas verslui kontekste. *QoS* yra pabrėžiamos tokios savybės kaip saugumas, patikimumas, kontrolės galimybė.

Saugumą lemia daugelis aspektų:

- autentifikacija yra saugumo elementas, kurio dėka galima nustatyti komunikacijos dalyvius arba patikrinti dokumento kilmę. Paprastai vartotojai autentifikuojami pagal vartotojo prisijungimo vardą ir slaptažodį;
- autorizacija – vartotojo prisijungimo teisių prie tam tikrų resursų patikrinimas. Vyksta po to, kai vartotojas sėkmingai autentifikuotas. Kiekvienam vartotojui gali būti priskirti skirtingi priėjimo prie programų ar kitų informacijos resursų lygiai;
- nepertraukiamumas užtikrina transakcijos garantuota įvykdymą;
- konfidencialumas saityno tarnybose reiškia, kad informacija skirta tam tikram asmeniui negali būti skaitoma ar kitaip pasiekama asmeniui, kuriam ji nėra skirta.

1.2.7. Saityno tarnybų tarpininkas

Tai komponentas, esantis tarp užsakovo ir saityno tarnybos tiekėjo (pav. 4). Jo panaudojimo atvejis galėtų būti vartotojo identifikavimas ir nustatymas, ar galima priimti ir perduoti informaciją.



Paveikslas 4. Saityno tarnybos tarpininko iliustracija

Tokiu tarpininku dažniausiai būna kita saityno tarnyba, kuri atlieka kelias tam tikras operacijas. Dažniausiai norint išvengti didesnio atskirų tarnybų kiekio, tokios funkcijos įdiegiamos pagrindinėje tarnyboje.

1.2.8. Saityno tarnybos verslo valdymo sistemose

Egzistuoja įvairūs verslo modeliai. Vienas iš jų – verslas verslui, arba kitaip žinomas kaip sutrumpintas variantas *B2B (Business to business)*. Kad būtų lengviau suprasti, *B2B* galima lyginti su *B2C (Business to customers)* – verslas klientams.

Kadangi įmonės dažnai turi verslo valdymo programinę įrangą, tenka sukurti integraciją, jeigu atsiranda dar viena, papildoma sistema. Norint užtikrinti saugumą ir integruotumą šiam tikslui naudojamos saityno tarnybos.

Taip pat naudojama ir SDK (angl. *Software development kit*). Tai yra funkcijų ir metodų rinkinys, kuris leidžia atlikti norimas operacijas esamoje sistemoje iš išorinės sistemos.

Per verslo ir techninių procesų integraciją, kompanijos sugeba sustiprinti santykius su partneriais ir klientais, pasiekti vientisą integraciją iš išorės ir įmonės viduje, pasiekti klientų atsiskaitymų matymą realiu laiku, padidinti produktyvumą ir sumažinti kainas.

1.3. Analizės išvados

Nustatyta, jog saityno tarnybos naudoja SOAP protokolą, o šiuo protokolu siunčiami duomenys yra pateikiami XML formatu. UDDI technologijos pagalba galima surinkti ir saugoti informaciją apie saityno tarnybas. Organizacijoms siūloma sistema, suteikianti galimybę aprašyti savo verslą ir nuosavas saityno tarnybas bei surasti kitų organizacijų siūlomas tarnybas. UDDI registras, tai tokia pat saityno tarnyba, teikianti struktūrinę informaciją apie kitas saityno tarnybas. Saityno tarnybų aprašymo kalba WSDL yra pagrįsta XML technologija ir leidžia sudaryti katalogus bei aprašyti saityno tarnybas. WSDL yra naudojama komunikavimo proceso tarp saityno tarnybų automatizavimui. WSDL aprašo saityno tarnybos sąsają, prisijungimo prie saityno tarnybos protokolą (SOAP) ir galutinę tarnybos būvimo vietą. SOAP pranešimą sudaro trys pagrindinės dalys: vokas, antraštė ir pranešimo vidinė informacinė (pagrindinė) dalis, apgaubta atitinkamo tinklo protokolo pranešimo su atitinkama protokolo antrašte. JSON – tekstinis, žmogui suprantamas duomenų formatas, atvaizduojantis duomenų struktūras ir asociatyvius masyvus. JSON dažniausiai naudojamas kartu su *Ajax* technologija duomenims perduoti ir yra naudojamas kaip XML alternatyva.

2. UŽSAKYMŲ VALDYMO PLATFORMOS PROJEKTAS

2.1. Įvadas

Užsakymų valdymo platformos uždavinys – optimizuoti užsakymų valdymo procesą, taupyti laiką ir resursus derinant užsakymus bei kiek įmanoma maksimaliai teisiškai padėti apsisaugoti tiekėjams ir klientams. Trumpai tariant, sistema perkelia popierinius dokumentus į elektroninę erdvę, padaro komunikaciją patogesnę, padeda darbuotojams lengviau surasti aktualią informaciją apie užsakymus, taupo įmonės resursus dokumentų, *media* elementų ir kitų bylų archyvavimui bei padeda analizuoti nukrypimus nuo terminų ir pardavimų efektyvumą.

Kuriamas produktas skirtas segmentui, kuriame įmonės užsiima eksportu, IT, marketingo ir kitomis, iš anksto neapmokamomis paslaugomis. Jis tinkamas Lietuvos ir užsienio rinkoms.

Įmonių darbuotojai atliktų daugiau darbų, jeigu turėtų daugiau laiko. Siūlomas sprendimas įmonėms gali padėti sutaupyti laiko, organizuoti vykstančius užsakymų valdymo procesus, didinti įmonės konkurencingumą ir kelti eksporto lygį.

Pirmieji užsakymų valdymo platformos pardavimai bus atliekami Lietuvoje ir palaipsniui plečiamasi į užsienio rinkas.

Įmonėms bus patogų, nes nereikės papildomai investuoti į individualių sistemų diegimą, jų talpinimą, apsaugą bei darbo su klientais procesų optimizavimą.

2.2. Projekto tikslas

Projekto tikslas – optimizuoti darbą, susijusį su užsakymais ir jų apdorojimu. Įmonėms taps lengviau ir pigiau derėtis su klientais, priiminėti jų užsakymus ir patiems orientotis tarp susitarimo detalių bei terminų.

Užsakymų valdymo platforma – tai įrankis, kuriame integruoti svarbiausi ir dažniausiai naudojami įrankiai dirbant su klientais.

Įmonė, dirbanti su šia platforma, gali sutaupyti laiko ruošdama komercinius pasiūlymus klientams, o vadybininkai galės apdoroti žymiai didesnę informacijos kiekį.

Kiekvienos sutarties vykdymui reikalingas užsakymo priedas, o įvykdžius užsakymą ir perduodant prekes ar paslaugas užsakovui – paslaugų priėmimo-perdavimo aktas. Šiems dokumentams paruošti ir suderinti užtrunka daug laiko. Sumažinus laiko sąnaudas ir resursų poreikį tokiems ir panašaus pobūdžio darbams, personalas spėtų atlikti žymiai daugiau darbo. Tai reiškia, kad dokumentų tvarkymas ir derinimas tiekėjo įmonei kainuotų daug pigiau.

Platforma turėtų būti naudinga įmonėms, nes:

- sutaupyti laiką derinant užsakymus ir juos vykdant;
- sumažėtų reklamai reikalingi kaštai;
- suteiktų informacijos apie tiekėjo patikimumą iš kitų įmonių paliktų atsiliepimų ir vertinimų;
- perkelia popierinius dokumentus į elektroninę erdvę;
- nereikia diegti savo atskiros sistemos;
- nereikia skirti lėšų duomenų apsaugai ir archyvavimui;
- apsaugo tiekėją ir klientą dėl iškreiptai suprastų detalių;
- leidžia parsisiųsti detalias ataskaitas;
- leidžia analizuoti vykdymo procesą ir nukrypimus nuo sutartų terminų;
- dėl platformoje integruotos efektyvios paslaugų tiekėjų paieškos įmonėms didėja galimybė tapti pastebėtomis globalioje rinkoje. Tai itin padėtų mažoms ir vidutinėms įmonėms bei suteiktų galimybę naujoms įmonėms būti pastebėtomis.

Visada komunikacija su klientais užima daug laiko. Neretai pasitaiko, kad kalbėdamas klientas supranta, jog jam kaina yra nepriimtina arba iš kitos pusės – tiekėjas – jog jam norimas užsakyti projektas yra per mažos apimties ir jį vykdyti būtų nuostolinga. Galiausiai projekto atsisakyta, o laiko išėikvota daug.

Dirbant su užsakymais, visuomet sudėtinga susigaudyti tarp gausybės elektroninių laiškų. Vienuose būna užsakymo informacija, kituose klausimai, dar kituose pavienės bylos, susijusios su projektu, ir įvairūs komentarai. Esant dideliame jų kiekiui nesunku pamiršti, ką konkrečiai klientas norėjo padaryti, ir užtrunka laiko surasti, kur tam tikros bylos ir kokia sutarta kaina.

Viena iš dažnai iškylančių problemų – tai nežinojimas, ar mūsų pasirinktas paslaugų teikėjas yra patikimas ir kaip jaučiasi jo buvę ar esami klientai. Kadangi galime rasti tikrai skundus, pavyzdžiui, internetinėje svetainėje „skundai.lt“, kur gali rašyti bet kas, teisingą nuomonę susidaryti sunku. Ieškoti gerų paslaugų teikėjų per savo pažįstamų ir draugų ratą užtrunka taip pat nemažai laiko.

Kadangi didelės kompanijos yra investavusios daug pinigų į savo vardo reklamą ir žinomumą, potencialūs klientai dažniausiai renkasi jas. Dažnai smulkių klientų poreikiai yra ne tokio didelio masto, kaip tų kompanijų, į kurias jie kreipiasi įprastai vykdomi, taigi jų laiškai lieka neatsakyti ir atmesti. Kitais atvejais didžiosios kompanijos tokius užsakymus perduoda savo pažįstamiems ar draugams, kurie galėtų užsiimti tokiais darbais. Taip gaunama ilga grandinė ir yra didelė tikimybė, jog užsakymas bus atliktas prastai, užvėlintas terminas arba netgi išvis neįvykdytas. Būna tokių atvejų, kad susidaro situacija, jog galbūt užsakovui net nežinant užsakymą vykdo „rangovo subrangovo subrangovas“.

Tiek paslaugų teikėjams, tiek ir paslaugų gavėjams pasitaiko patirti nuostolių dėl kurios nors šalies nekompetentingumo, netiksliai įgyvendintų darbų ir vėlavimų. Šiuo atveju turima omenyje, kad teikėjai ir gavėjai yra nedidelės įmonės, kurios neturi didelio kapitalo. Paprasčiausias pavyzdys būtų, kai klientas užsisako paslaugas iš laisvai samdomo specialisto, aptaria sąlygas ir detales telefonu, po to elektroniniu paštu ir dar vėliau pokalbių programa „Skype“, po to, tikėdamasis gerų rezultatų, užsakovas investuoja į reklamą ir kitas reikalingas sritis savo pinigus, o kai reikia, laisvai samdomas specialistas arba neatlieka tinkamai darbų, arba jų išvis neatlieka ir „pradingsta“. Žvelgiant iš kitos pusės, viską teisingai ir laiku atlikęs laisvai samdomas specialistas gali nesulaukti atlyginimo ir piktų priekaištų dėl atliktų darbų, siekiant išvengti arba nutolinti atsiskaitymą už paslaugas.

2.3. Problemos sprendimas pasaulyje

Kadangi produktas yra tinkamas daugelyje šalių atliekamoms užsakymų valdymo operacijoms optimizuoti, planuojama, jog susidomėjimas šiuo produktu bus didelis. Dėl sistemos pobūdžio užsienio rinką pasiekti bus nesunku. Tam užtenka, kad Lietuvoje esanti eksportuojanti įmonė priimtų užsakymą iš užsienio kompanijos naudodamasi užsakymo valdymo platforma, o užsienio kompanija taip pat norėtų užsakymų priėmimą ir valdymą perkelti į sistemą.

Kompanija gali įsidiegti projektų valdymo sistemą ir jos pagalba valdyti reikiamas užduotis. Šiam poreikiui įgyvendinti yra daug pasirinkimo variantų, tiek nemokamų tiek ir mokamų, tačiau toks sprendimas reikalauja kliento apmokymo naudotis sistema, šios sistemos dažniausiai būna griozdiškos, sunku apmokyti ir įpratinti komandą teisingai naudotis tokia sistema, o jos funkcionalumas pritaikytas tik konkrečios srities projektų vykdymui. Čia praleidžiamas paties užsakymo ir susitarimo momentas, visa komunikacija vis tiek vykdoma elektroniniu paštu, o tokiai sistemai palaikyti reikia pinigų ir resursų.

Komunikacijos priemonės kaip *Gmail* ar *Skype* plačiai naudojamos bendravimui su klientu, tačiau dirbant daugiau nei su vienu arba tiesiog ilgainiui atsiranda problema, kai reikia rasti reikiamą informaciją, yra tikimybė, kad duomenys dings.

Kompanijos Adobe sukurta *EchoSign* dokumentų pasirašymo sistema leidžia sudaryti sutartis ir jas pasirašyti, tačiau viskas tuo ir apsiriboja. Taip pat Lietuvoje tokiu būdu pasirašyta sutartis nėra laikoma galiojančia.

Taip pat apžvelgti projektai: *BusyFlow.com*, *Avilys*, *BMP Solutions*, *DocLogix* ir *Kantorius*.

Šiuo metu naudojamos priemonės yra elektroninis paštas (pvz. *Gmail*), pokalbių programos (pvz. *Skype*), telefono skambučiai, dokumentų pasirašymo sistemos (pvz. *EchoSign*) ir projektų valdymo sistemos (pvz. *Redmine*). Kuriamos platformos išskirtinumas tas, kad joje bus patogiai

sujungta ir suvienyta dauguma šių sistemų principų ir klientams nebereikės atskirų įrankių tam pačiam rezultatui pasiekti.

2.4. Programų sistemos funkcijos

Išanalizavus poreikius ir kylančias problemas yra aišku, jog būtinas sprendimas, kuris sumažintų resursų ir laiko sąnaudas komunikacijai tarp šalių, supaprastintų ir sukongretintų užsakymų valdymo procesus ir suteiktų daugiau garantijų būti teisiškai apsaugotiems jei iškiltų nesutarimai.

Pagrindinės platformos funkcijos:

- užsakymų konkretizavimas ir maksimaliai supaprastinta komunikacija;
- matomumo galimybė paslaugų teikėjams ir pardavėjams;
- teisinė apsauga;
- reikalingų resursų mažinimas norint pasiekti kokybiškus rezultatus ir matomumą;
- perkelti popierinius dokumentų šablonus į elektroninę erdvę.

Į užsakymų valdymo platformą įeina keturi pagrindiniai komponentai:

- užsakymų valdymo modulis;
- ataskaitų generavimo modulis;
- įmonės profilio modulis;
- atsiliepimų ir įvertinimų modulis.

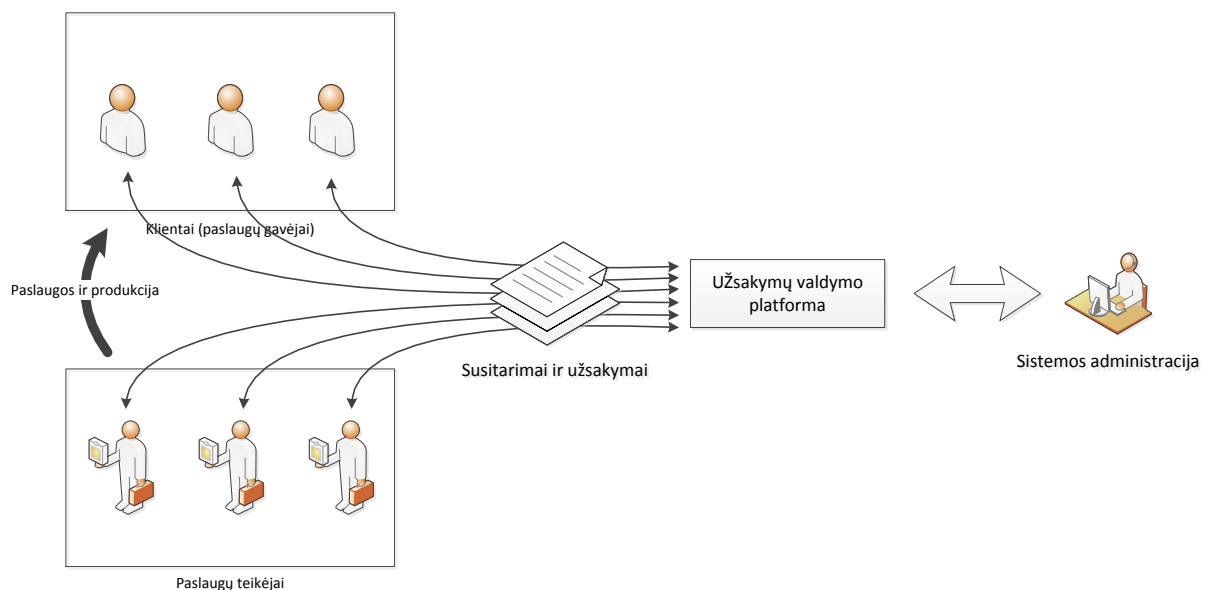
Toliau pateikiamas užsakymo valdymo platformos lango eskizas (pav. 5):

Įmonė X	Kitų įmonių paieška	Vadybininkas Jonas
Pagrindiniai	Užsakymas Nr. W00016-2 (iš įmonės Y)	Sutartis
Užklauso (naujų 5) Užsakymai (naujų 2) Vykdomi užsakymai Komunikacija (naujų 6) Kontaktai (nauj. pakv. 1)	<p>2012-11-25 2012-12-05</p> <p>Užsakymas (PIRMINIS) Būsena: Patvirtinta</p> <p>Reikia sukurti reklaminį skydelį Vykdoma</p> <p>Pokalbiai (15) Failai (2) Detalės (1)</p> <p>Užsakyta 2012-11-25 / Atlikti iki 2012-12-05 / Trukmė 11d. / Kaina 500 LT / Delspinigiai 0,004%/d. (0.00 LT)</p>	Failai
Nustatymai	Užsakymas 2 (papildymas) Būsena: Nepatvirtinta	Dokumentai Specifikacijos Media elementai Navigacija Komunikacija Užsakymų papildymai Sutartys Pateikimai Ataskaitos
Kompanijos puslapis Darbuotojai Formos Statistika Ataskaitos	Reikalinga JavaScript integracija svetainei	Apskaita
	Pokalbiai (8) Failai Detalės (4)	Išankstinės s/f PVM s/f Užsakymo vertė
	Užsakyta 2012-12-01 / Atlikti iki 2012-12-06 / Trukmė 6d. / Pasiūlyta kaina 100 LT	

Paveikslas 5. Užsakymų valdymo platformos užsakymų peržiūros lango karkasinis modelis

Paveikslo (pav. 5) kairėje pusėje matome meniu punktus, kurie leidžia pasiekti užsakymus pagal jų dabartinę būseną. Taip pat yra nustatymų skiltis, kurioje vartotojas gali sistemos sąsają pritaikyti pagal savo poreikius. Dešinėje pusėje yra meniu, kuris leidžia pasiekti susijusius dokumentus ir bylas su užsakymais. Taip pat yra nuorodos, leidžiančios pasiekti informaciją apie susirašinėjimą su klientu, gauti ataskaitas bei preliminarios apskaitos duomenis. Vidurinėje skiltyje pateikiamas visų užsakymų sąrašas.

Sistemoje egzistuoja keli vartotojų tipai. Juos iliustruoja žemiau pateiktas paveikslėlis (pav. 6).



Paveikslas 6. Sistemos veikėjai

Čia (pav. 6) pateikiama koncepcija, kaip turėtų sąveikauti skirtingi sistemos veikėjai:

- teikėjas – paslaugas teikianti arba prekes tiekianti įmonė;
- klientas – užsakymą teikianti įmonė arba interesantas;
- administracija – sistemos administracija, prižiūrinti, kad sistemoje užregistruotos įmonės atitiktų registrų centre registruotas įmones, nebūtų sukčiavimo atvejų, atsakinėjanti į klientų (sistemos vartotojų) klausimus bei stebinti, kad sistema veiktų korektiškai;
- nauji, neregistruoti asmenys, kurie ieško teikėjų – iš kitų šaltinių atvykę asmenys, kurie ieško paslaugų teikėjų.

2.5. Bendri apribojimai

Kadangi projekto vartotojų bazės augimas gali būti remiamas virusinės plėtros efektu, tikėtina, jog gali nepakakti turimų serverių resursų ir žmogiškųjų išteklių informacijai apdoroti ir sistemai prižiūrėti.

2.6. Projektavimo metodologijos ir technologijų analizė

Pastebėta, kad daugelyje kompanijų vienas iš sąlyginai ilgai trunkančių procesų yra užsakymų ir sutarčių derinimas tarp dalyvaujančių šalių. Yra sukurta įvairių įrankių šiam procesui palengvinti, tačiau kažkodėl nei vienas įrankis nėra sukonkretizuotas būtent tokio pobūdžio problemų pilnam ir visapusiškam sprendimui.

Praktikoje pasitaiko, jog klientai atsisako paslaugų arba išeina su jau įgytomis žiniomis (angl. *know-how*) pas konkurentus. Nors darbai jau pradėti vykdyti, nėra už ko užsikabinti, siekiant atlyginti žalą. Pagal LR įstatymus sudaryta sutartis internete ir taip pat pasirašyta – nėra laikoma galiojančia, tačiau užsakymai laikomi galiojančiais ir šalis apsaugo taip pat, kaip toks užsakymas būtų sudarytas raštiškai.

2.7. Projekto apribojimai

2.7.1. Įpareigojantys apribojimai

2.7.1.1. Apribojimai sprendimui

Pagrindinis apribojimas ir įpareigojimas – sistema turi būti pritaikyta darbui visomis kalbomis ir turi būti palaikomos bent jau pagrindinės valiutos bei formatai.

2.7.1.2. Diegimo aplinka

Platforma bus diegiama į Windows serverį. Taip pat kartu bus diegiamas .NET 4, .NET 4.5 ir MVC 4.

2.7.1.3. Bendradarbiaujančios sistemos

Serveryje bus įdiegtos papildomai sukurtos saitynų tarnybos, kurios atliks atsarginių kopijų darymo funkciją, naujienlaiškių siuntimą, integraciją su išorinėmis sistemomis ir kitų pranešimų apdorojimą.

2.7.1.4. Komerciniai specializuoti programų paketai

Kaip testavimo įrankio dalis bus naudojamas grafikų piešimo modulis. Šis modulis bus perkamas arba bus ieškoma nemokamo pakaitalo. Nemokamo grafikų piešimo modulio pavyzdys – *Google Chart* komponentas. Jo sąsaja ir atvaizdavimo būdas yra paprasti. Tačiau kuriant, išaugus poreikiams, gali tekti ieškoti mokamo, daugiau funkcijų turinčio modulio.

Taip pat gali prireikti mokamo tinklelio (angl. *Grid*) gražiam duomenų atvaizdavimui.

2.7.1.5. Numatoma darbo vietos aplinka

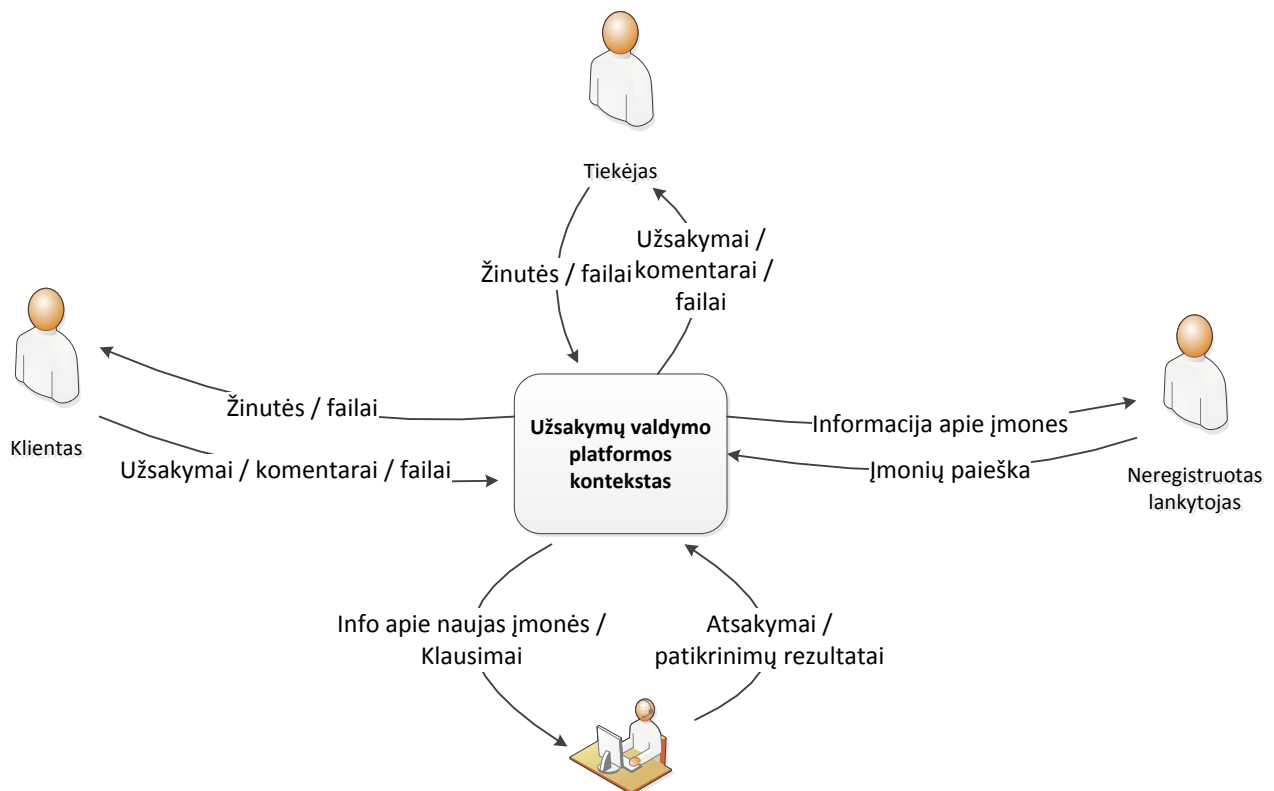
Kadangi įrankį naudos įmonių vadovai, projektų vadovai ir vadybininkai, kurie dirba arba namuose, arba įmonės biure, darbo vieta nesiskirs nuo standartinių. Tad ir aplinka nedarys jokios įtakos efektyviam sistemos darbui.

2.8. Funkciniai reikalavimai

2.8.1. Veiklos sritis

2.8.1.1. Veiklos kontekstas

Projektų valdymo platformos kontekstas pavaizduotas paveikslėlyje (pav. 7).



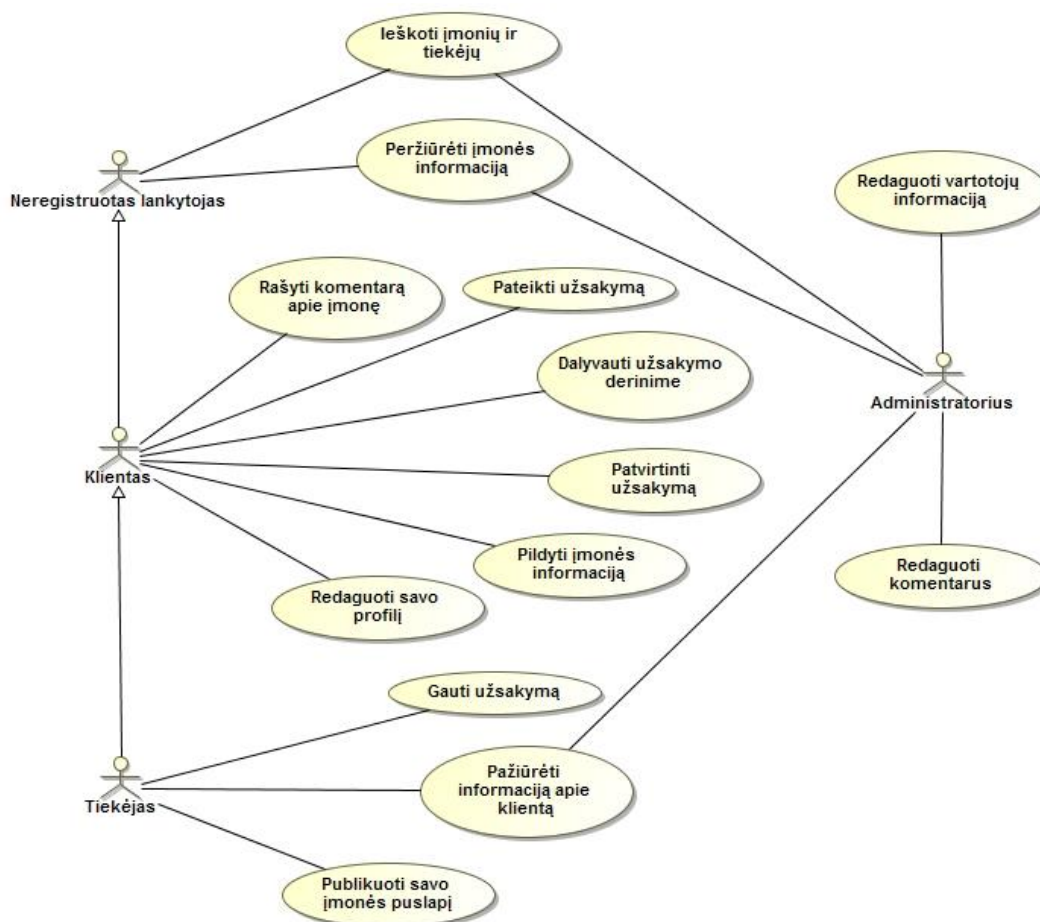
Paveikslas 7. Kontekstinė diagrama

7 paveikslėlyje pavaizduota, kaip sąveikauja skirtingi veikėjai. Matome, kad yra du registruoti vartotojai – klientas ir tiekėjas – taip pat neregistruotas lankytojas (kurių gali būtų 0..N). Taip pat privalomas administratorius, kuris prižiūri sklandų sistemos veikimą ir padeda vartotojams spręsti kilusius klausimus.

2.8.2. Sistemos sudėtis

2.8.2.1. Sistemos ribos

Užsakymų valdymo platformos panaudojimo atvejų diagrama pateikiama 8 paveikslėlyje. Diagramoje matyti keturi aktoriai: administratorius, registruoti vartotojai (klientas ir tiekėjas) ir neregistruotas lankytojas. Neregistruotas lankytojas gali būti tiek žmogus, tiek, pavyzdžiui, *Google* robotas.



Paveikslas 8. Panaudos atvejų diagrama

Šioje diagramoje (pav. 8) pavaizduota, kokius veiksmus gali atlikti klientai – tiekėjas ir klientas, neregistruotas lankytojas bei sistemos administratorius. Tiekėjas turi daugiausia galimybių, nes jo vartotojo paketas yra brangiausias ir tuo pačiu leidžia atlikti daugiau dalykų.

Toliau pateikiami pagrindinių panaudos atvejų aprašymai.

Lentelė 1. PA „Redaguoti vartotojų informaciją“

14 Panaudojimo atvejis	Panaudojimo atvejis	Redaguoti vartotojų informaciją
Aktoriai		Administratorius
Aprašas		Redaguoti vartotojo registracijos duomenis, t. y. patikslinti arba papildyti
Prieš sąlygos		Pasirinktas vartotojas
Sužadinimo sąlygos		Paspaudžiamas įrašymo mygtukas
Po sąlygos		Atnaujinami vartotojo duomenys

Lentelė 2. PA „Redaguoti komentarus“

15 Panaudojimo atvejis	Panaudojimo atvejis	Redaguoti komentarus
Aktoriai		Administratorius
Aprašas		Pašalinti komentarus, jeigu jie netinkami
Prieš sąlygos		Pasirinktas įmonės profilis
Sužadinimo sąlygos		Paspaudžiamas trynimo mygtukas
Po sąlygos		Panaikintas komentaras

Lentelė 3. PA „Ieškoti įmonių ir tiekėjų“

16 Panaudojimo atvejis	Panaudojimo atvejis	Ieškoti įmonių ir tiekėjų
Aktoriai		Administratorius, neregistruotas lankytojas
Aprašas		Įmonių paieška
Prieš sąlygos		Atidarytas paieškos langas
Sužadinimo sąlygos		Paspaudžiamas paieškos mygtukas
Po sąlygos		Pateiktas įmonių sąrašas

Lentelė 4. PA „Peržiūrėti įmonės informaciją“

17 Panaudojimo atvejis	Panaudojimo atvejis	Peržiūrėti įmonės informaciją
Aktoriai		Administratorius, neregistruotas lankytojas
Aprašas		Peržiūrima įmonės informacija, komentarai, vertinimai
Prieš sąlygos		Pasirinktas įmonės profilis
Sužadinimo sąlygos		Pasirenkama įmonė – paspaudžiamas mygtukas eiti į įmonę
Po sąlygos		Pateikiama informacija

Lentelė 5. PA „Rašyti komentarą apie įmonę“

18 Panaudojimo atvejis	Panaudojimo atvejis	Rašyti komentarą apie įmonę
Aktoriai		Klientas, tiekėjas
Aprašas		Komentuojama įmonė
Prieš sąlygos		Pasirinktas įmonės profilis
Sužadinimo sąlygos		Paspaudžiamas įrašymo mygtukas
Po sąlygos		Įrašomas komentaras

Lentelė 6. PA „Pateikti užsakymą“

19 Panaudojimo atvejis	Pateikti užsakymą
Aktoriai	Klientas, tiekėjas
Aprašas	Pateikiamas užsakymas įmonei
Prieš sąlygos	Pasirenkama įmonė
Sužadinimo sąlygos	Paspaudžiamas pateikimo mygtukas
Po sąlygos	Pateiktas užsakymas

Lentelė 7. PA „Dalyvauti derinant užsakymą“

20 Panaudojimo atvejis	Dalyvauti derinant užsakymą
Aktoriai	Klientas, tiekėjas
Aprašas	Redaguojamas ir pateikiamas užsakymas
Prieš sąlygos	Pasirenkamas užsakymas
Sužadinimo sąlygos	Paspaudžiamas pateikimo mygtukas
Po sąlygos	Pateikta nauja dokumento versija

Lentelė 8. PA „Patvirtinti užsakymą“

21 Panaudojimo atvejis	Patvirtinti užsakymą
Aktoriai	Klientas, tiekėjas
Aprašas	Patvirtinamas („pasirašomas“) užsakymas
Prieš sąlygos	Pasirinktas užsakymas
Sužadinimo sąlygos	Paspaudžiamas patvirtinimo mygtukas
Po sąlygos	Užsakymas patvirtintas

Lentelė 9. PA „Pildyti įmonės informaciją“

22 Panaudojimo atvejis	Pildyti įmonės informaciją
Aktoriai	Klientas, tiekėjas
Aprašas	Pildoma įmonės informacija
Prieš sąlygos	Atidarytas įmonės redagavimo puslapis
Sužadinimo sąlygos	Paspaudžiamas įrašymo mygtukas
Po sąlygos	Atnaujinta įmonės informacija

Lentelė 10. PA „Redaguoti savo profilį“

23 Panaudojimo atvejis	Redaguoti savo profilį
Aktoriai	Klientas, tiekėjas
Aprašas	Vartotojo profilio redagavimas
Prieš sąlygos	Atidaryta profilio redagavimo forma
Sužadinimo sąlygos	Paspaudžiamas įrašymo mygtukas
Po sąlygos	Atnaujintas profilis

Lentelė 11. PA „Gauti užsakymą“

24 Panaudojimo atvejis	Gauti užsakymą
Aktoriai	Tiekėjas
Aprašas	Tiekėjas turi teisę gauti užsakymą. Gaunamas pranešimas apie naują užsakymą
Prieš sąlygos	
Sužadinimo sąlygos	(Klientas pateikia užsakymą)
Po sąlygos	Tiekėjas gauna SMS ir el. laišką su pranešimu

Lentelė 12. PA „Pažiūrėti informaciją apie klientą“

25 Panaudojimo atvejis	Pažiūrėti informaciją apie klientą
Aktoriai	Administratorius, tiekėjas
Aprašas	Kliento, pateikusio užsakymą, informacijos peržiūra
Prieš sąlygos	Pasirinktas kliento profilis
Sužadinimo sąlygos	Paspaudžiamas atidarymo mygtukas
Po sąlygos	Matoma informacija

Lentelė 13. PA „Publikuoti savo įmonės puslapį“

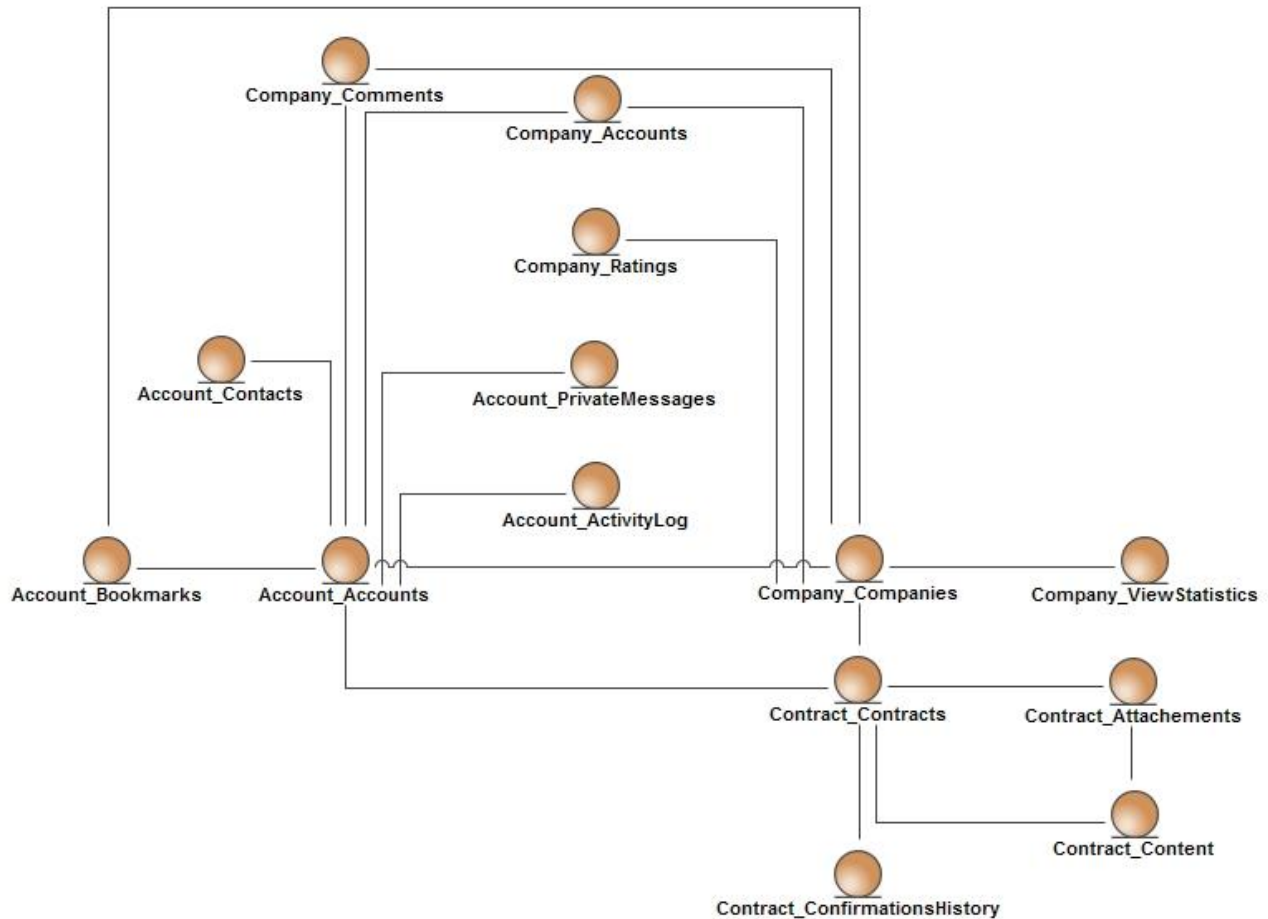
26 Panaudojimo atvejis	Publikuoti savo įmonės puslapį
Aktoriai	Tiekėjas
Aprašas	Publikuoti viešą įmonės puslapį
Prieš sąlygos	Pasirinkti nustatymai
Sužadinimo sąlygos	Paspaudžiamas publikavimo mygtukas
Po sąlygos	Įmonės puslapis viešai prieinamas

Aktoriai:

- administratorius – sistemą ir jos vartotojus prižiūrintis asmuo;
- klientas – vartotojas, kuris gali pateikti užsakymus;
- tiekėjas – vartotojas, kuris gali pateikti ir gauti užsakymus;
- neregistruotas lankytojas – vartotojas, kuris gali tik peržiūrėti informaciją apie įmones.

2.9. Reikalavimai duomenims

Kaip pavaizduota esybių diagramoje (pav. 9), duomenų bazės struktūra yra analogiška. Duomenų laukai pildosi nuolatos pagal klientų pageidavimus ir poreikius.

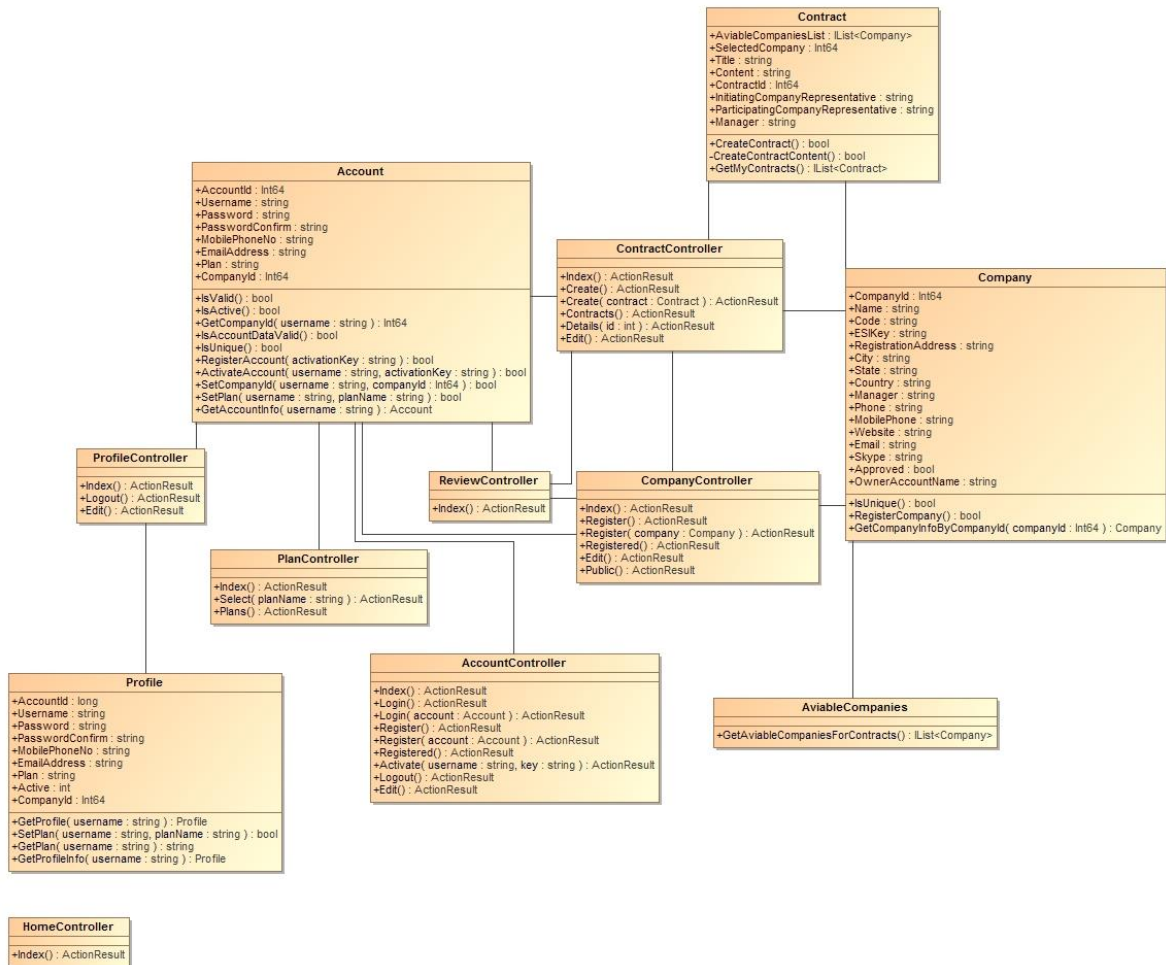


Paveikslas 9. Duomenų bazės struktūra

Ši schema (pav. 9) vaizduoja pagrindinių duomenų bazės lentelių išsidėstymą ir sąryšius tarp jų. Tokia schema pateikia mažiau informacijos, tačiau padeda lengviau orientuotis, kur turėtume ieškoti tam tikrų duomenų.

2.10. Klasių diagrama

Toliau pateikiama sistemos klasių diagrama (pav. 10). Tai yra pačios sistemos komponentai, kuriems padedant galima išsaugoti, gauti ir panaudoti esamus duomenis.



Paveikslas 10. Sistemos klasių diagrama

Pateikta klasių diagrama (pav. 10) vaizduoja pagrindines portalo klases ir jų metodus, kurie yra reikalingi sistemos veikimui. Pateikiamos vartotojo sąskaitos klasės, vartotojo profilio ir jo redagavimo klasės, įmonės profilio ir jo redagavimo klasės bei užsakymų tarp įmonių valdymo klasės.

2.11. Apibendrinančios išvados

Sistemoje yra keturi vartotojų tipai: neregistruotas lankytojas, klientas, tiekėjas ir administratorius, kurie atitinkamai gali atlikti skirtingus veiksmus sistemoje. Sistema susideda iš vartotojo sąskaitos, profilio, įmonės puslapio ir užsakymų valdymo bloką. Pasirinkta programavimo aplinka – Windows, .NET 4 ir 4.5 ir MVC architektūra.

3. SAITYNO TARNYBŲ PROTOKOLŲ SAVYBIŲ TYRIMAS

3.1. Tikslas

Šiuo tyrimu siekiama išsiaiškinti duomenų protokolų ir juos palaikančių formatų apdorojimo charakteristikas, pagrindinį dėmesį kreipiant į našumą ir stabilumą.

Buvo nuspręsta atlikti eksperimentą, kurio rezultatai parodytų, kuris protokolas ir duomenų formatas yra greičiausiai ir stabiliausiai apdorojamas. Laikoma, kad duomenų gavimo trukmė yra vienoda visais atvejais. Šiuo tyrimu norima nustatyti, kas greičiau apdorojamas – REST/JSON ar SOAP/XML formatas.

Ankstesniuose punktuose (žr. 1.2 punktą) nustatyta, kad SOAP yra patogesnė duomenų perdavimo forma, jei duomenų struktūra yra sudėtinga ir reikalingas didesnis saugumas. Tačiau jeigu norime dirbti su kiek paprastesnėmis struktūromis, mums gali pakakti ir JSON formato (žr. 1.2.5 punktą).

Naudojant JSON formato duomenis, programą galime sukurti šiek tiek greičiau, nes šiai struktūrai reikia sąlyginai mažiau kodo.

3.2. Tyrimo nuostatos

Tyrimui pasirinkta .NET platforma ir C# programavimo kalba. Saityno tarnyba sukurta naudojant *Microsoft WCF Service* [8]. Ši saityno tarnyba yra įdiegiama ir veikia kaip Microsoft Windows IIS 8 priedas.

Kompiuterio, kuriame atliekamas tyrimas, duomenys:

- Procesorius: Pentium(R) Dual-Core CPU T4400 @ 2.20 Ghz 2.20 Ghz
- Atmintis (RAM): 4Gb
- Sistemos tipas: x64
- OS: Windows 7 Ultimate

JSON tyrimui pasirinktas trečios šalies įskiepis „Newtonsoft.Json“ [9].

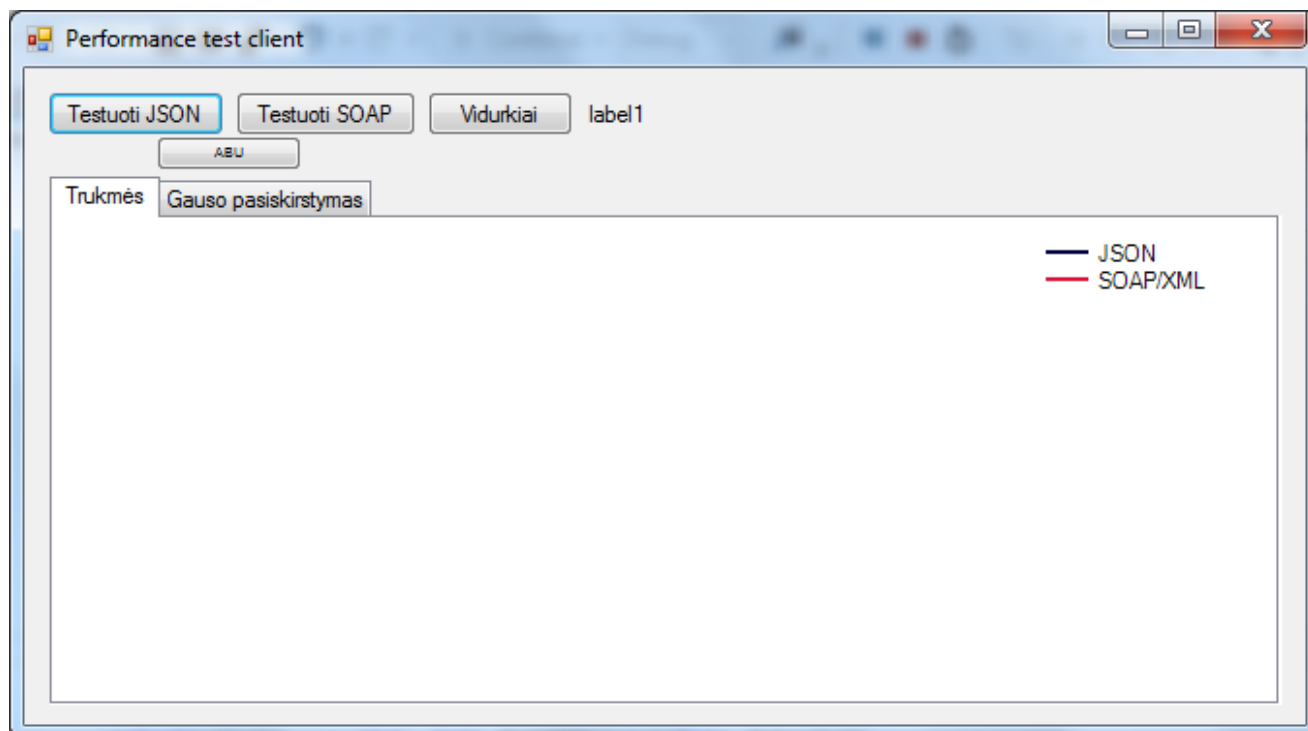
Tyrimui naudojamos papildomai sukurtos dvi programos. Viena atlieka saityno tarnybos vaidmenį, kita – kliento.

Programoje modeliams naudojami *DataContract* ir *DataMember* atributai, o operacijoms *OperationContract* ir *ServiceContract* atributai.

Klientinė programa vykdo užklausas saityno tarnybai tiek kartų, kiek nurodyta kintamajame *reqCount*. Programa vizualiai atvaizduoja trukmes, kiek laiko reikia gauti rezultatą iš saityno tarnybos. Taip pat pateikiamas Gauso skirstinys.

Užklauso vykdomos paeiliui, tuomet duomenys apibendrinami ir atvaizduojami.

Programos langas atrodo taip (pav. 11):



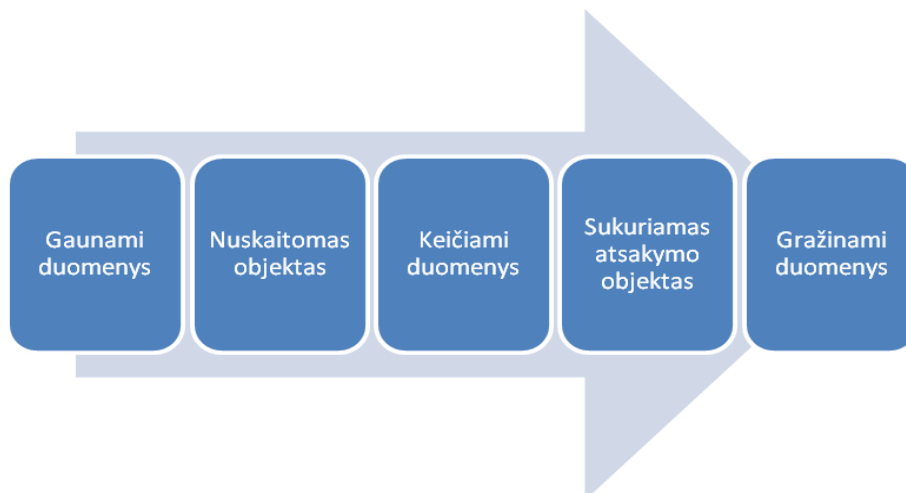
Paveikslas 11. Kliento programos langas

Matome (pav. 11), kad programos lange yra atvaizduojamas trukmės grafikas ir Gauso pasiskirstymas. Vartotojas turi galimybę pasirinkti, ar nori tirti vieną kurį nors protokolą, ar abu iš karto. Pasirinkti abu iš karto yra patogiu, kai nustatomas didelis kiekis duomenų, nes tokia operacija trunka gana ilgai.

3.3. Tyrimo metodika

Saityno tarnyboje yra du metodai, kurie priima užklausas iš kliento, atlieka duomenų pakeitimą ir grąžina atgal. Metodai yra vienodi siekiant neiškreipti rezultatų tikslumo. Tarnyba gali priimti SOAP arba JSON formato užklausas. Grąžina taip pat atitinkamo formato duomenis.

Saityno tarnyboje vykdomas toks procesas (pav. 12):



Paveikslas 12. Proceso iliustracija

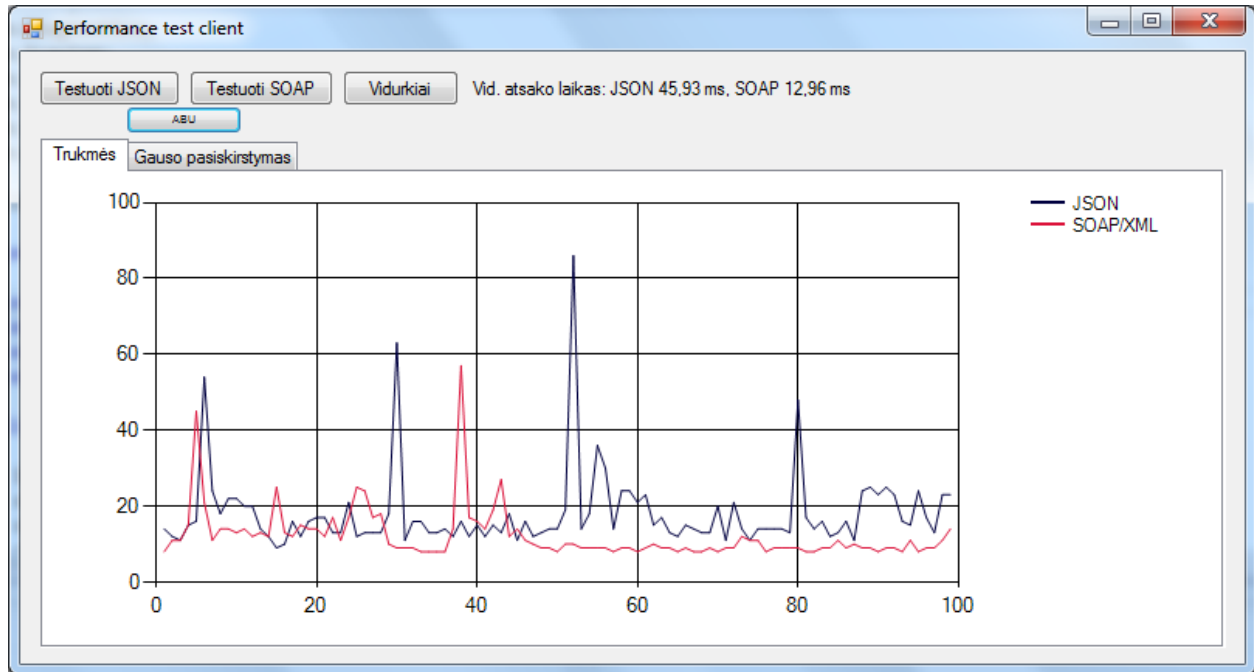
Vykstant šiam procesui (pav. 12) yra priimamas duomenų paketas, jis paverčiamas objektu, kuriuo gali operuoti sistema, tuomet atliekamas duomenų pakeitimas, objektas paverčiamas atgal siuntimui tinkamu formato objektu ir išsiunčiamas. Atliekant eksperimentą rezultatams įtakos neturi interneto greیتaveika, nes tiriamas tik objektų apdorojimo laikas ir stabilumas.

Tyrimo rezultatams apskaičiuoti bus taikomas Gauso skirstinys [10]. Jo pagalba galėsime vizualiai matyti, kuris protokolas ir duomenų formatas yra efektyvesnis.

3.4. Rezultatai ir jų analizė

3.4.1. Rezultatai

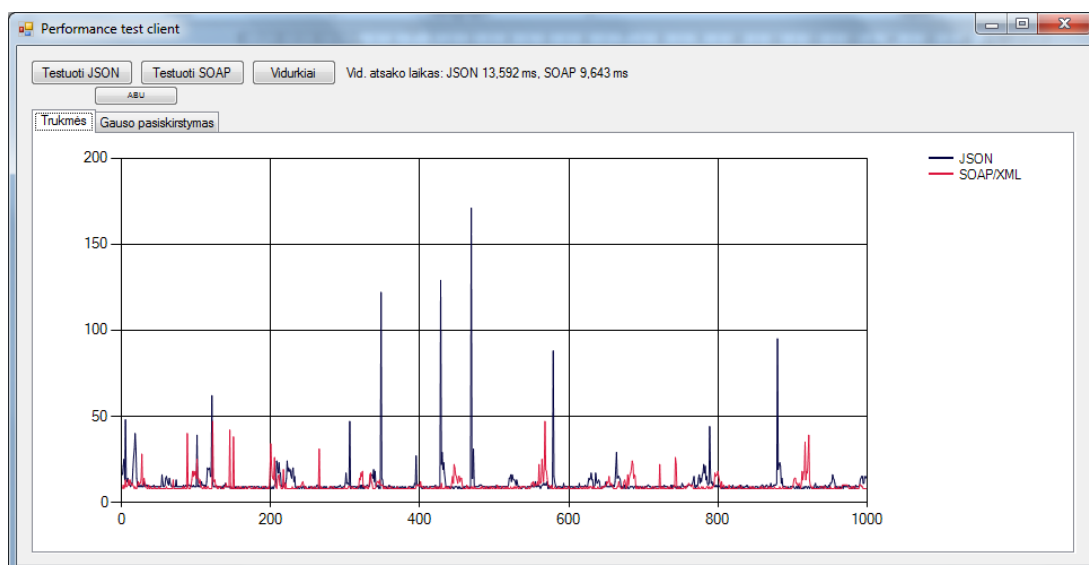
Atlikus skaičiavimus su 100 užklausų kiekvienam tiriamajam objektui, gaunami tokie grafikai:



Paveikslas 13. Bandyto su 100 užklausų rezultatai

Kaip matome iš grafiko (pav. 13) ir vidutinio laiko (paveikslas viršuje), REST/JSON užklausos trunka ilgiau. Kadangi sistema veikia lokaliaje terpėje, interneto greitimeika neturi įtakos. Pastebimi dažnesni pikai apdorojant JSON formato duomenis. Tam gali turėti įtakos kiti, kompiuteryje vykstantys procesai, bei tai, kad naudojamas komponentas yra trečios šalies.

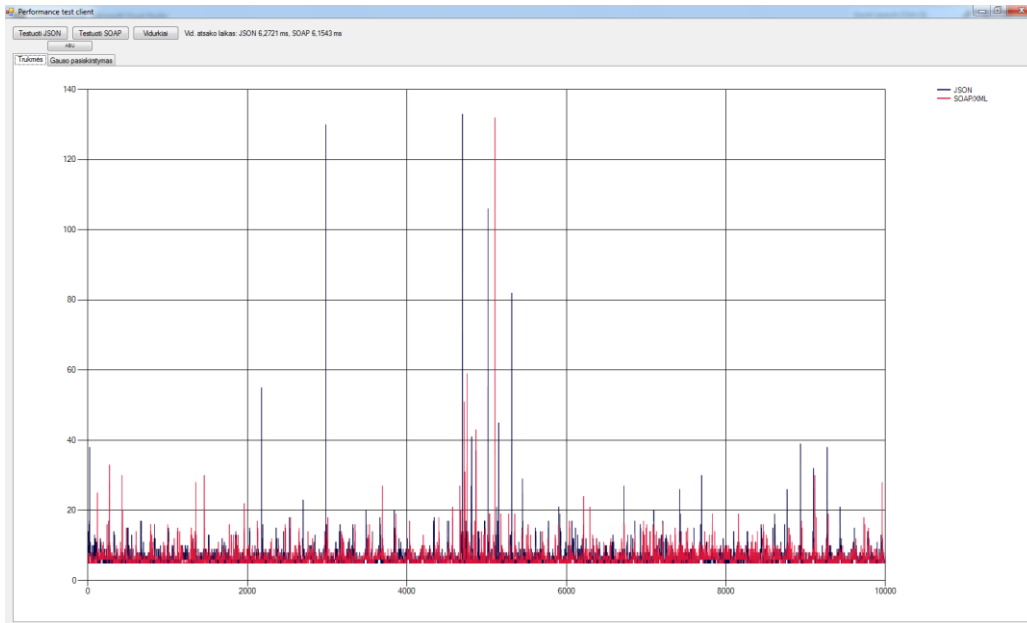
Padidiname užklausų kiekį iki 1000 ir gauname tokius rezultatus (pav. 14):



Paveikslas 14. Bandyto su 1000 užklausų rezultatai

Trukmės šuoliai (pav. 14) atsiranda dėl sistemos apkrovai įtaką darančių kitų faktorių, pavyzdžiui, kitų tarnybų, kaip *Skype*, interneto naršyklė, duomenų bazės stotys ir pan.

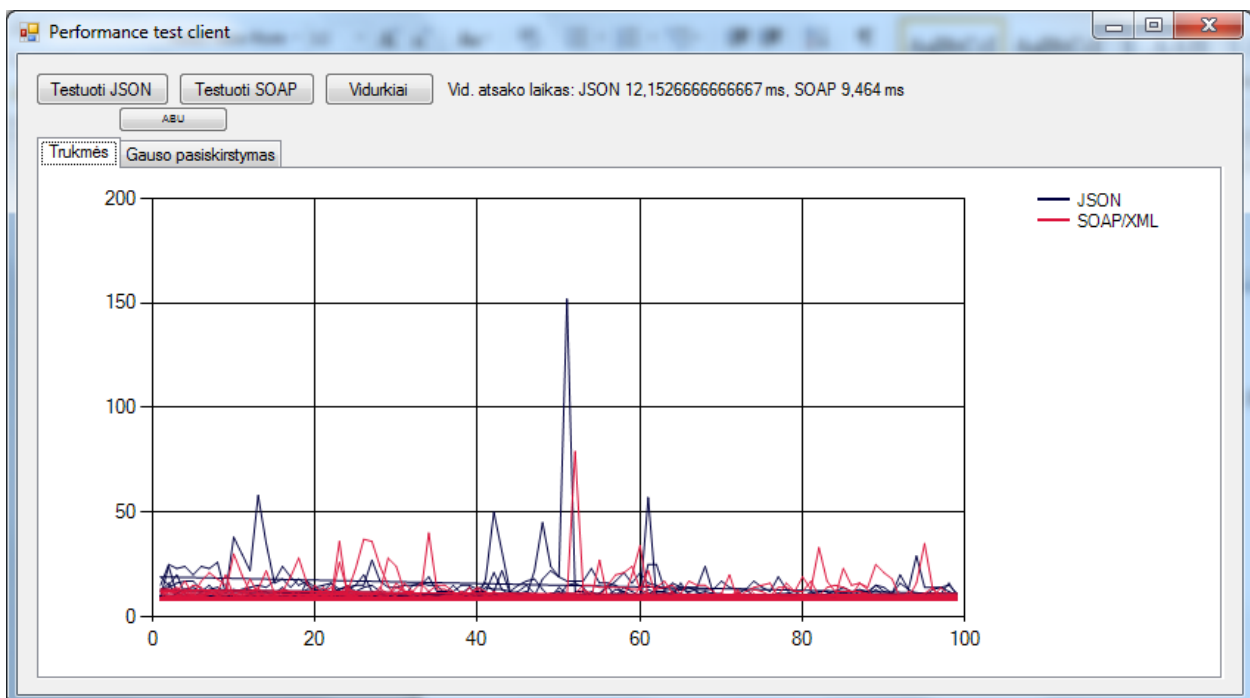
Toliau nustatomas 10000 užklausų kiekis ir gaunami tokie rezultatai:



Paveikslas 15. Bandymo su 10000 užklausų rezultatai

Šiuo atveju abu variantai beveik lygūs (pav. 15).

Dėl tikslesnių rezultatų atliekamas eksperimentas 15 kartų po 100 užklausų. Duomenys piešiami tuose pačiuose grafikuose (pav. 16):

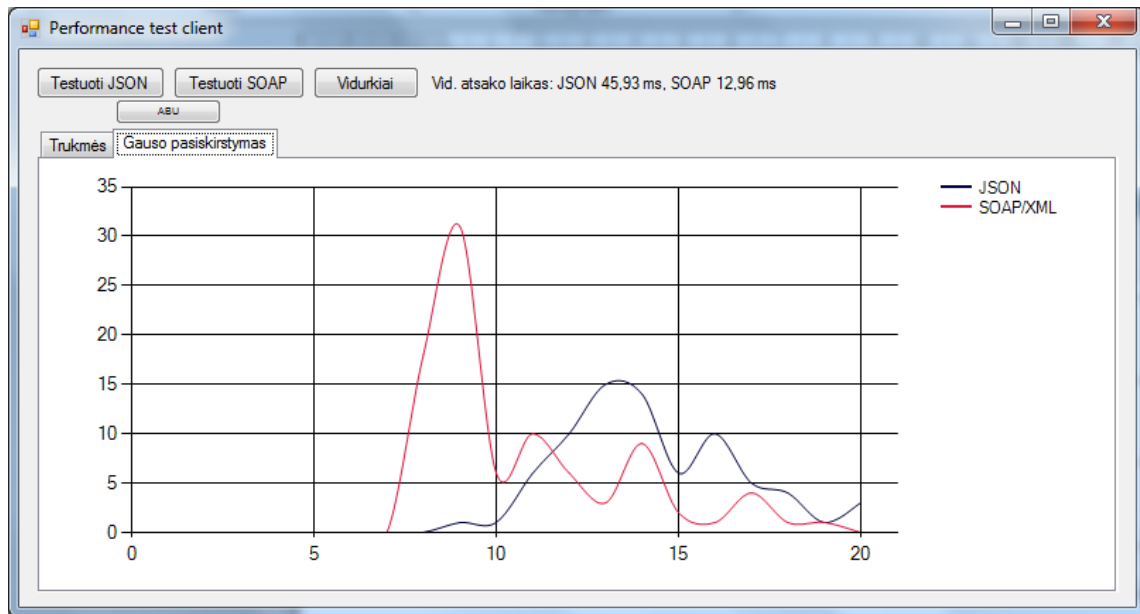


Paveikslas 16. Bandymo su 15x100 užklausų rezultatai

3.5. Analizė

Pagal eksperimento rezultatus suskaičiuoti Gauso skirstiniai. Grafiko X ašis yra laiko trukmė, kiek trunka užklausa, o Y ašis – kiek atitinkamos trukmės įvykdyta užklausų.

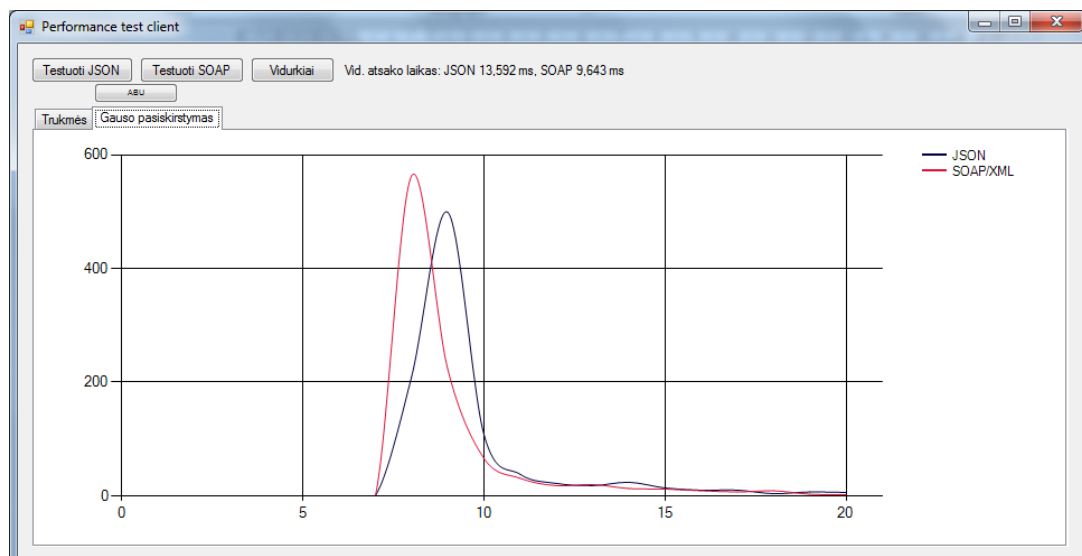
Ekspirimentas su 100 užklausų:



Paveikslas 17. Bandymo su 100 užklausų rezultatų Gauso skirstinys

Iš šio rezultatų grafiko (pav. 17) matome, kad SOAP/XML buvo apdorojamas greičiausiai ir lyginant su REST/JSON – stabiliau. JSON viršūnė pasislinkusi dešiniau, tai reiškia, kad užklauso atsakymas grįždavo vėliau. Lygindami abi viršūnes matome, kad vidutinis atsakymo laikas skiriasi per 16 ms.

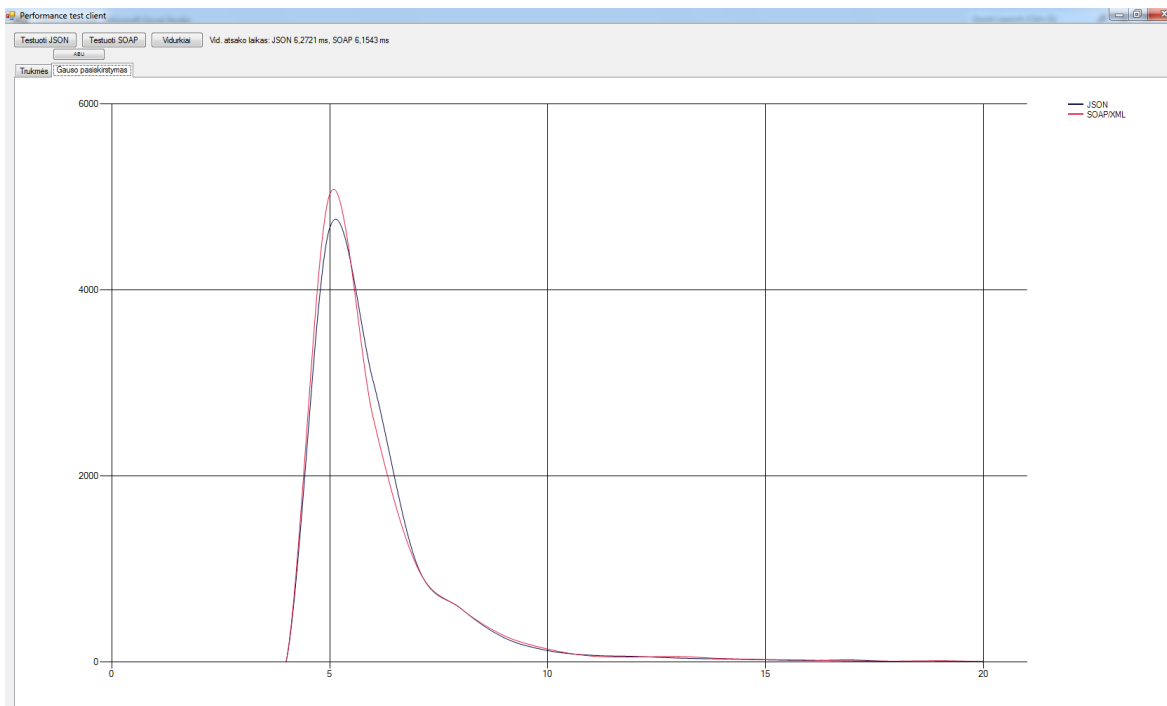
Ekspirimentas su 1000 užklausų:



Paveikslas 18. Bandymo su 1000 užklausų rezultatų Gauso skirstinys

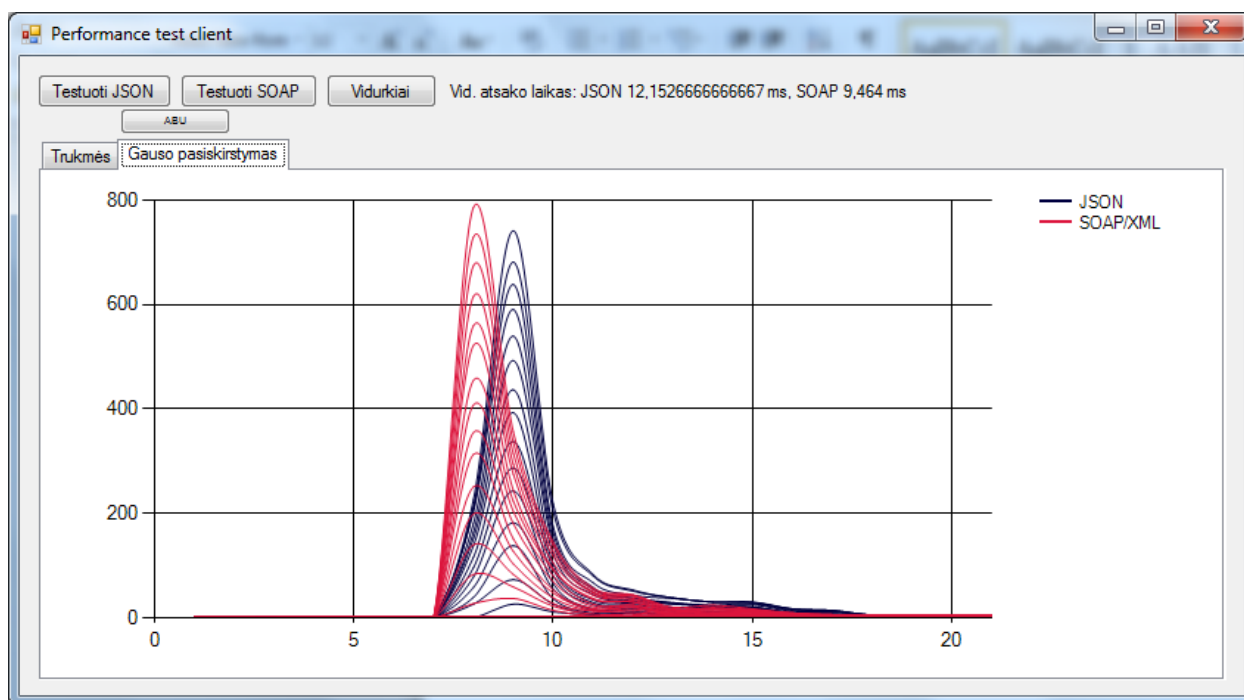
Pateikus daugiau užklausų, rezultatų grafikai (pav. 18) pasidaro gražesni ir statesni. Kuo statesnis grafikas – tuo stabilesnė sistema.

Ekspertas su 10000 užklausų (pav. 19):



Paveikslas 19. Bandymo su 10000 užklausų rezultatų Gauso skirstinys

Atlikus 15 kartų po 100 užklausų eksperimentą, gauti tokie skirstiniai:



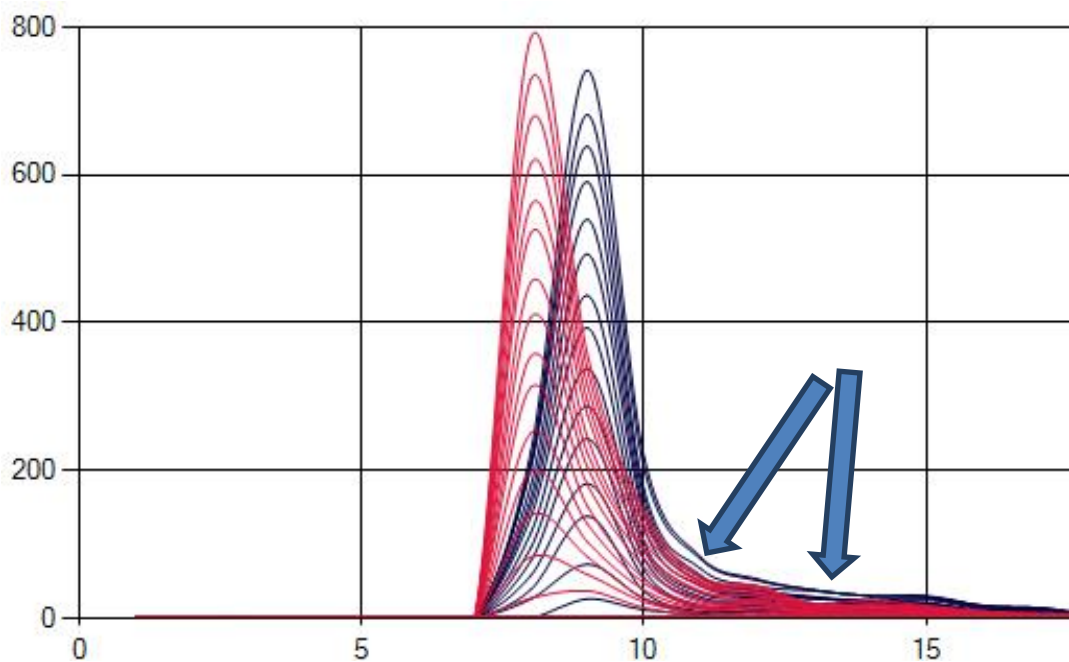
Paveikslas 20. Bandymo su 15x100 užklausų rezultatų Gauso skirstiniai

Šiame grafike (pav. 20) matome penkiolikos bandymų su šimtu užklausų rezultatus. Jie buvo piešiami ant to paties grafiko siekiant vaizdžiai parodyti rezultatus ir jų nuokrypius. Kaip matome, SOAP/XML visais atvejais pirmauja lyginant su REST/JSON.

3.6. Apibendrinimas

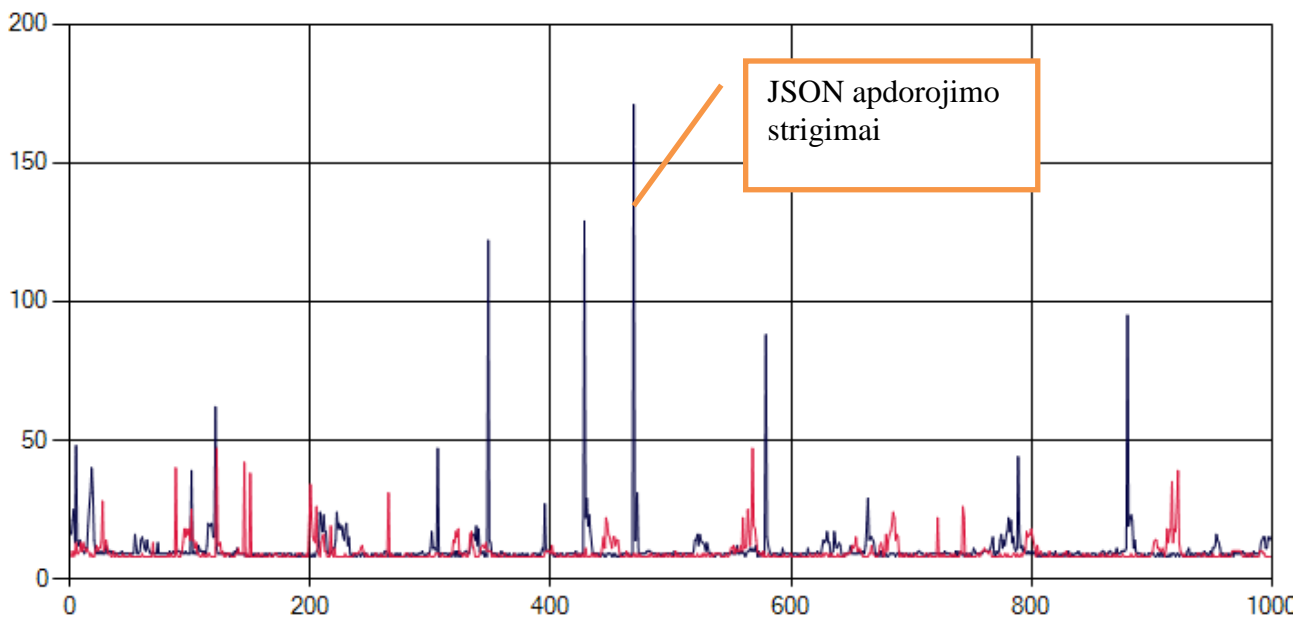
Iš Gauso pasiskirstymo grafiko matome (pav. 21), kad SOAP apdorojimas yra stabilesnis, dėl to jo apdorojamų duomenų paketų trukmės rezultatai geresni.

Iš paskutinio grafiko matome, kad JSON yra mažiau stabilus ir jo trukmė yra dauguma atvejų ilgesnė. Taip pat matome, kad JSON grafiko dešinioji pusė yra aukščiau negu SOAP/XML ir jos tolimesni taškai netolygiai yra iškilę, o tai reiškia, jog nors dauguma rezultatų sugrįžo greičiau, SOAP/XML duomenų perdavimo modelis apdorojamas stabiliau.



Paveikslas 21. REST/JSON ir SOAP/XML rezultatų Gauso skirstiniai

Pagal kitą grafiką (pav. 22) matome, kad iš SOAP varianto galime tikėtis didesnio stabilumo ir geresnio atsakymo laiko. Kreivių nuožulnumas rodo, kad tikimybė atsirasti strigimams didesnė REST/JSON (mėlyna).



Paveikslas 22. REST/JSON ir SOAP/XML rezultatų diagrama. Matomi trukmės šuoliai, kurie reiškia, kad sistema užtrunka labai ilgai apdorodama duomenis

Atsižvelgiant į programinį kodą, JSON yra paprasčiau aprašoma struktūra. Jeigu nėra kritiškai svarbi greitimeika, galima svarstyti JSON panaudojimą, taip sutaupant resursų projekto įgyvendinimui. Kita vertus, SOAP apibrėžia tikslią duomenų struktūrą ir dėl to ateityje išvengiama nesusipratimų ir yra lengviau kviesti reikiamus metodus.

JSON apdorojamas ilgiau, nes SOAP yra .NET vidinis komponentas ir greičiausiai parašytas naudojant žemiausio lygmens programavimo kalbą. Tuo tarpu JSON yra trečios šalies priedas, kurį reikia atskirai įdiegti į projektą, kad būtų galima naudotis šia biblioteka. Tikėtina, kad ji parašyta aukščiausio lygio kalba ir dėl to operacijų laikas išauga, o stabilumo rodikliai suprastėja.

4. REZULTATŲ APIBENDRINIMAS IR IŠVADOS

4.1 Šiame darbe buvo analizuojama, kaip veikia saityno tarnybos, kokie jų privalumai ir trūkumai. Apžvelgta, kokie duomenų perdavimo protokolai naudojami ir kaip formatuojami perduodami duomenys.

4.2 Išnagrinėti populiariausi protokolai ir jų duomenų formatai – REST/JSON ir SOAP/XML. JSON yra paprastai aprašoma duomenų struktūra, kuri užima nedaug vietos, lyginant su XML. Tuo tarpu XML yra aprašomas detaliau ir proporcingai užima daugiau vietos. Nustatyta, kad SOAP saityno tarnybos pateikia viešas instrukcijas, taigi jomis naudojantis yra lengviau pritaikyti kuriamą programinę įrangą pasirinktos tarnybos naudojimui. SOAP taip pat turi integruotą saugumo sistemą, taigi apsaugos klausimas tampa paprastesnis.

4.3 Buvo iškelti klausimai, kokį duomenų perdavimo protokolą ir duomenų formatą pasirinkti, norint sukurtą sistemą praplėsti saityno tarnybomis ir jas panaudoti kuo efektyviau; kuris formatas apdorojamas stabiliau ir greičiau Windows operacinėje sistemoje, .NET 4 aplinkoje.

4.4 Eksperimentiniams tyrimams atlikti buvo sukurtos dvi programos. Vienos paskirtis – priimti, apdoroti ir grąžinti duomenis, kitos – siųsti užklausas serverio programai bei apskaičiuoti rezultatus.

4.5 Nustatyta, kad stabiliau ir greičiau apdorojamas SOAP/XML. REST/JSON užtrunka šiek tiek ilgiau, grafike matyti žymūs duomenų apdorojimo sulėtėjimai. Manoma, kad tam įtakos turėjo kiti aplinkoje veikiantys procesai. Platformos funkcionalumo praplėtimui pasirinktas SOAP/XML.

4.6 Šiam tyrimui įtakos neturėjo interneto greitaveika, tačiau tai labai svarbus faktorius, kuris taip pat lemia protokolo pasirinkimą. SOAP yra didesnės apimties duomenų modelis, o JSON yra aprašomas minimaliai, taigi jis užima mažiau vietos. Jeigu atsižvelgiama į perduodamų duomenų kiekį, pavyzdžiui, mobiliajai programėlei komunikuojant su serveriu, vis dėlto geresnis variantas būtų JSON. Šios platformos praplėtimo atveju interneto greitaveika ir kaina nėra tokios svarbios lyginant su poreikiu apdoroti daug duomenų.

5. LITERATŪRA

- [1] [Tinkle]. Prieiga: http://en.wikipedia.org/wiki/Web_service. [Žiūrėta 01 04 2013].
- [2] [Tinkle]. Prieiga: http://www.tutorialspoint.com/webservices/web_services_characteristics.htm. [Žiūrėta 01 04 2013].
- [3] [Tinkle]. Prieiga: <http://www.baltic-amadeus.lt/lt/apie/naudinga/soa/uddi>. [Žiūrėta 01 04 2013].
- [4] [Tinkle]. Prieiga: <http://www.baltic-amadeus.lt/lt/apie/naudinga/soa/wsdl>. [Žiūrėta 01 04 2013].
- [5] [Tinkle]. Prieiga: <http://www.baltic-amadeus.lt/lt/apie/naudinga/soa/wsdl>. [Žiūrėta 01 04 2013].
- [6] [Tinkle]. Prieiga: <http://www.baltic-amadeus.lt/lt/apie/naudinga/soa/soap>. [Žiūrėta 01 04 2013].
- [7] [Tinkle]. Prieiga: <http://en.wikipedia.org/wiki/JSON>. [Žiūrėta 01 04 2013].
- [8] [Tinkle]. Prieiga: http://en.wikipedia.org/wiki/Windows_Communication_Foundation. [Žiūrėta 01 04 2013].
- [9] [Tinkle]. Prieiga: <http://james.newtonking.com/pages/json-net.aspx>. [Žiūrėta 01 04 2013].
- [10] [Tinkle]. Prieiga: https://en.wikipedia.org/wiki/Normal_distribution. [Žiūrėta 01 04 2013].