

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Robertas Mikaitis

**Programinė įranga mobiliojo ryšio paslaugų
operatoriams**

Magistro darbas

Vadovas:

Prof. K. Motiejūnas

Kaunas, 2010

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Robertas Mikaitis

**Programinė įranga mobiliojo ryšio paslaugų
operatoriams**

Magistro darbas

Recenzentas:		Vadovas:	
2010-05-31		2010-05-31	prof. K. Motiejūnas
	doc. D. Rubliauskas		
		Atliko:	
		2010-05-31	IFM-4/2 gr. stud.
			Robertas Mikaitis

Kaunas, 2010

TURINYS

ĮVADAS	7
1. ANALITINĖ DALIS	8
1.1 Literatūros apžvalga	8
1.1.1 JAIN SLEE architektūra	8
1.1.2 Kliento serverio architektūra	10
1.2 Virtualios telefoninės stotelės sistemos analizė	12
1.2.1 Probleminiai uždaviniai ir jų sprendimai	13
1.2.2 Realizavimo technologijos parinkimas	18
1.2.3 Egzistuojančios sistemos ir jų palyginimas	19
1.3 Analizės išvados	25
2. PROJEKTINĖ DALIS	26
2.1 Projekto tikslai	26
2.2 Sprendžiami uždaviniai	27
2.3 Priimti techniniai sprendimai	28
2.4 Diegimo aplinka	29
2.5 Veiklos kontekstas	32
2.6 Panaudojimo atvejai	33
2.7 Funkciniai reikalavimai	36
2.8 Nefunkciniai reikalavimai	37
2.9 Architektūros specifikacija	39
2.9.1 Bendras architektūros vaizdas	39
2.9.2 Sistemos dinaminis vaizdas	43
2.9.3 Duomenų vaizdas	50
2.10 Praktinis darbo rezultatas	52
3. EKSPERIMENTINĖ IR TIRIAMOJI DALIS	54
3.1 Platformų darbo patikimumas	54
3.2 Platformų darbo našumas	56
3.3 Kūrimo aplinkos	59
3.4 Egzistuojantys komponentai ir platformų savybės	62
3.5 Tyrimo išvados	64
IŠVADOS	65
LITERATŪRA	66
TERMINŲ IR SANTRUMPŲ ŽODYNAS	67

Software Solution For Mobile Network Operators

Summary

The objective of this project was to develop software for mobile network operators, which provides virtual private branch exchange functionality and has operator console for its control. During its execution, the analysis of design and technology solutions was performed. The architecture of the developed software is based mainly on the principle of client server design and JAIN SLEE specification. Quality and feature analysis of two competing platforms was performed – jNetX OCFS and Mobicents JAIN SLEE. Main objective of analysis was to find out if Mobicents JAIN SLEE is feasible replacement for jNetX OCFS, analysis showed that it is.

PAVEIKSLŲ SĄRAŠAS

1 pav. JAIN SLEE komponentų išsidėstymo schema.....	10
2 pav. Kelių lygių kliento serverio architektūra	11
3 pav. Galima jNetX klasterio konfigūracija	14
4 pav. Sistemos darbo patikimumą turinti užtikrinanti jungimo schema	15
5 pav. Diegimo aplinkos schema	29
6 pav. Sistemos veiklos kontekstas.....	32
7 pav. Panaudos atvejų diagrama.....	35
8 pav. Bendra sistemos architektūra – komponentų išdėstymas.....	39
9 pav. Operatoriaus konsolę sudarantys paketai.....	41
10 pav. Virtualią telefoninę stotelę sudarantys komponentai	43
11 pav. Skambučio persiuntimo su konsultacija būsenų diagrama – virtualios telefoninės stotelės vaizdas	45
12 pav. Skambučio persiuntimas su konsultacija – skambučio iniciatoriaus operatoriaus konsolės	46
13 pav. Skambučio persiuntimo su konsultacija sekų diagrama – virtualios telefoninės stotelės vaizdas	48
14 pav. Skambučio persiuntimo su konsultacija sekų diagrama – operatoriaus konsolės vaizdas	49
15 pav. Skambučio persiuntimo su konsultacija sekų diagrama– operatoriaus konsolės vaizdas (tęsinys).....	50
16 pav. Duomenų bazės modelis	51
17 Pav. Testavimui naudotas skambučio scenarijus	55
18 pav. Testuojamų platformų darbo sutrikimai.....	55
19 pav. Skambučiai per sekundę.....	57
20 pav. Atminties naudojimas	58
21 pav. Skambučio komutavimo laikas	59
22 pav. TSS kūrimo aplinka	60
23 pav. Kodo eilučių skaičiaus pasiskirstymas pagal komponentą	62

LENTELIŲ SĄRAŠAS

1 lentelė. Probleminiai sistemos realizavimo uždaviniai	13
2 lentelė. Sistemos pasiekiamumas.....	14
3 Lentelė. Veiklos įvykių sąrašas	32
4 lentelė. Sistemos panaudojamumo grupės	33
5 lentelė. Duomenų bazės modelio esybės	51
6 lentelė. Pagrindinės kūrimo aplinkų savybės.....	61
7 Lentelė. Platformų palaikomi protokolai	62
8 Lentelė. Platformų savybės.....	63

IVADAS

Šiame dokumente nagrinėjama tyrimo sritis susijusi su mobiliojo ryšio operatoriams skirtomis virtualių telefoninių stotelių sistemomis. Analitinėje darbo dalyje atsakomi pagrindiniai šio projekto klausimai, pristatomos bendros idėjos ir vėliau apžvelgiama panašių sistemų specifika. Projektinėje dalyje į sistemą pažvelgiama iš architektūros pusės. Sukonkretinami priimti sprendimai ir pateikiamas jų realizacijos aprašymas. Tiriamoje ir eksperimentinėje darbo dalyje atliekamas dviejų telekomunikacijoms skirtų platformų tyrimas – jNetX OCFS ir Mobicents JAIN SLEE. Įvertinama sistemų kokybė ir pateikiamas apibendrintas platformų vaizdas kartu su išvadomis ar vertą esamą sistemą migruoti į naują platformą ir kokios kliūtys gali kilti tai atliekant.

Pastaraisiais metais sparčiai augo mobiliųjų telefonų populiarumas, o dabar jie tapo neatsiejama šiuolaikinio žmogaus gyvenimo dalimi [1]. Šiuo projektu buvo siekiama sukurti programinę įrangą, kuri yra skirta praplėsti mobiliojo ryšio operatorių teikiamų paslaugų spektrą tokiomis paslaugomis, kurios anksčiau buvo prieinamos tik laidiniams telefonams, prijungtiems prie telefoninės stotelės. Kuriama sistema sudaryta iš kelių komponentų. Svarbiausi iš jų yra virtuali telefoninė stotelė ir operatoriaus konsolė.

Virtuali telefoninė stotelė, tai programinė įranga veikianti operatoriaus serveriuose ir atliekanti analogiškas funkcijas aparatinei telefoninei stotelei: skambučių blokavimas, peradresavimas, persiuntimas, priparkavimas ir kt. Operatoriaus konsolė yra skirta stebėti ir valdyti skambučiams, bei keisti telefonų konfigūraciją.

Programinė įranga skirta mobiliojo ryšio operatoriams, o jos paslaugomis naudosis smulkaus ir vidutinio verslo atstovai. Šios programinės įrangos dėka operatoriai galės pateikti papildomų paslaugų, ir dėl to gaus didesnes pajamas bei prisivilios naujų klientų iš tų operatorių, kurie neteikia analogiškų paslaugų. Mobiliojo ryšio operatorių abonentai galės suteikti kokybiškesnes paslaugas savo klientams, bei sumažinti kaštus.

1. ANALITINĖ DALIS

Šiame skyriuje analizuojama kuriama sistema bei aptariamoms pasirinktos technologinės priemonės. Apibrėžiama tyrimo sritis, analizuojamas objektas bei įvardijamos problemos, kurias turėtų išspręsti kuriamas produktas. Taip pat atsakomi pagrindiniai šio projekto klausimai, pristatomos bendros idėjos ir vėliau apžvelgiamos kelios panašios sistemos bei jų specifika. Suformuluojamas uždavinys ir projektavimo metu iškilusios projektavimo ar programavimo inžinerijos problemos. Pateikiama galimų sprendimų apžvalga, priimto ir realizuoto sprendimo pagrindimas bei įvertinimas.

1.1 Literatūros apžvalga

1.1.1 JAIN SLEE architektūra

Šiuo metu sparčiai populiarėja aplikacijų serveriai, tokie kaip JBOSS, GlassFish, WebSphere. Jie jau naudojami daugelyje sričių, ne išimtis ir telekomunikacijos. Poskyryje bus apžvelgiama Java kalbai telekomunikacijoms skirtų serverių architektūra JAIN SLEE.

JAIN SLEE trumpinys iššifruojamas taip: Java sąsaja intelektualiems tinklams, servisų logikos vykdymo aplinka.

Servisų logikos vykdymo aplinka yra gerai žinoma koncepcija telekomunikacijų industrijoje. SLEE yra didelio pralaidumo, mažo atsako laiko įvykius apdorojanti aplikacijų vykdymo aplinka. JAIN SLEE [1] yra Java kalbos standartas skirtas SLEE.

JAIN SLEE yra sukurtas taip, jog leistų jį realizuojančioms platformoms atitikti visus telekomunikacijų aplikacijų keliamus reikalavimus, tokius kaip tinklo signalizacijos užtikrinimas. JAIN SLEE siekia, jog jį realizuojančios platformos užtikrintų geresnes plėtos galimybes ir didesnę sistemos pasiekiamumą.

JAIN SLEE yra vienintelis industrijos standartas sukurtas turint siekį užtikrinti komunikacijos aplikacijų pernešamumą – vieną kartą parašytos komunikacijoms skirtos aplikacijos turi galimybę veikti daugelyje skirtingų JAIN SLEE serverių realizacijų. Aplikacijų pernešamumą užtikrina giežtai nurodyti programavimo API, interpretacijos laisvės nepaliekanti specifikacija. Kaip pavyzdys gali būti pateikiama realizacija bei didžiulis kiekis testų, kuriuos gamintojas turi tenkinti, prieš tai, kol jis yra paskelbiamas kaip suderinamas su JAIN SLEE specifikacija.

JAIN SLEE apjungia įvairius tinklo resursus ir protokolus vienoje platformoje. Aplikacijos gali pasiekti daug skirtingų išorinių tinklo resursų iš JAIN SLEE aplinkos.

Ši specifikacija leidžia kūrėjams kurti nedidele apimtimi, bet didelę greitaveiką pasižyminčius komponentus, o jau sukurti komponentai gali būti apjungti siekiant išspręsti sudėtingesnius uždavinius.

JAIN SLEE specifikacija apibrėžia tokius pagrindinius komponentus:

- **Servisas (Service).** Servisas JAIN SLEE aplinkoje yra apibrėžiamas kaip valdomas aplinkos vienetas, kuris bet kada gali būti pakeistas. Sistemos administratorius kontroliuoja serviso gyvavimo ciklą (įdiegimą, pašalinimą, bei jo atnaujinimą nestabdant sistemos). Gyvavimo ciklo valdymas yra pasiekiamas naudojantis JAIN SLEE standarte apibrėžtomis sąsajomis. Servisą sudaro informaciją apie jį patį (serviso vardas, gamintojas, versija) ir programos kodas. Programos kodas gali būti sudarytas iš Java klasių, profilių, bei serviso blokų.

- **Profilis (Profile).** JAIN SLEE profilį sudaro konfigūracijos informacija. Konfigūracijos informacijai apibrėžti yra naudojama speciali schema, kuri yra sudaryta iš atributų ir jų reikšmių. Telefono numeris ir elektroninio pašto adresas gali būti laikomi kaip du skirtingi profilio atributai. JAIN SLEE taip pat apibrėžia ir profilio specifikacijos koncepciją. Profilio specifikacija sudaryta iš valdymo sąsajų skirtų profilio modifikavimui ar atnaujinimui. Servisų blokai veikiantys JAIN SLEE platformoje gali pasiekti šiuos profilius ir naudoti juos savo logikai vykdyti.

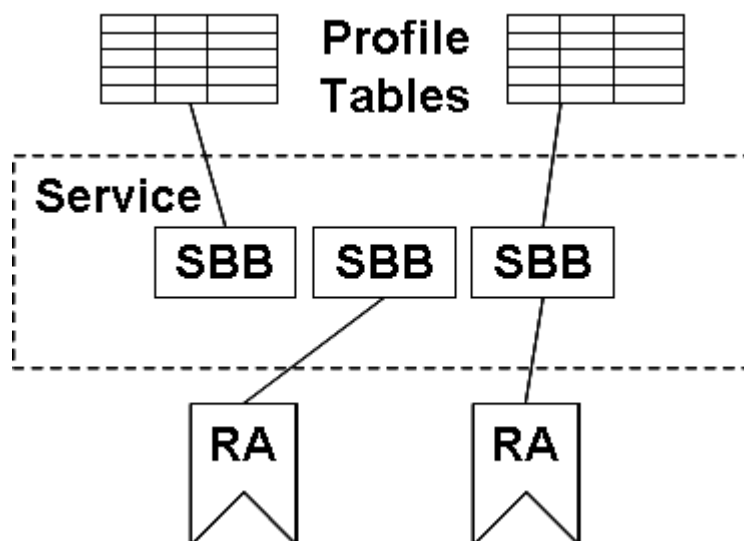
- **Serviso blokas (SBB).** Tai pakartotinio panaudojimo elementai, kuriuos apibrėžia JAIN SLEE specifikacija. Serviso blokas yra programinės įrangos komponentas, kuris siunčia ir gauna įvykius, bei atlieka atitinkamus veiksmus remiantis gautais įvykiais ir savo dabartine būseną. Serviso blokai yra komponentai saugantys būseną, kurios pagalba gali pasiekti prieš tai atliktų veiksmų rezultatus, bei atitinkamai elgtis gavus naujus įvykius. Serviso blokai gali būti sudaryti iš kelių kitų serviso blokų, taip galima gauti pakankamai sudėtingas operacijas atliekančias aplikacijas. Servisas yra sudarytas iš informacijos apie jį (pavadinimas, gamintojas, versija), įvykių kuriuos gauna servisas ir programos kodo (Java klasių) kuris yra vykdomas serviso veikimo metu.

- **Įvykis (Event).** Įvykis reprezentuoja nutikimą, kuris gali reikalauti jo apdorojimo. Jis saugo informaciją, kuri gali būti nutikimo šaltinis ar pan. Įvykis gali būti sukurtas daugelio skirtingų šaltinių – išorinio resurso (komunikacijos protokolų steko), pačios SLEE platformos, iš kito serviso SLEE platformoje. Įvykiai yra abstrakcijos, skirtos reprezentuoti nutikimus, kurie nėra susiję su jokia vykdoma programos atšaka - servisu. Kaip pavyzdį galima pateikti skambutį tarp Jono ir Petro. Petras gali paskambinti Jonui bet kuriuo metu. Tai nutikimas kai Jonas

skambina Petruui gali būti pateiktas kaip vienas įvykis, o nutikimas kai Petras atsiliepia kaip kitas įvykis.

- **Resursai ir resursų adapteriai.** Resursai yra išoriniai dariniai, kurie sąveikauja su kitomis sistemomis esančiomis už SLEE platformos ribų (MSC, HLR, duomenų bazės). Resursų adapteris yra skirtas pritaikyti šiuos resursus vidinėms JAIN SLEE sąsajoms, bei užtikrinti jų pasiekiamumą iš servisų.

Žemiau pateiktame paveiksle (žr. 1 pav.), galima matyti JAIN SLEE sudarančių komponentų išsidėstymo schemą. Brūkšniai jungiantys servisus (SBB) ir resursų adapterius (RA) reprezentuoja koks servisas iš kokio resursų adapterio gauna ir siunčia vykius. Brūkšniai jungiantys profilius (Profile Tables), nurodo koks servisas kokiais profiliais naudojami.



1 pav. JAIN SLEE komponentų išsidėstymo schema

1.1.2 Kliento serverio architektūra

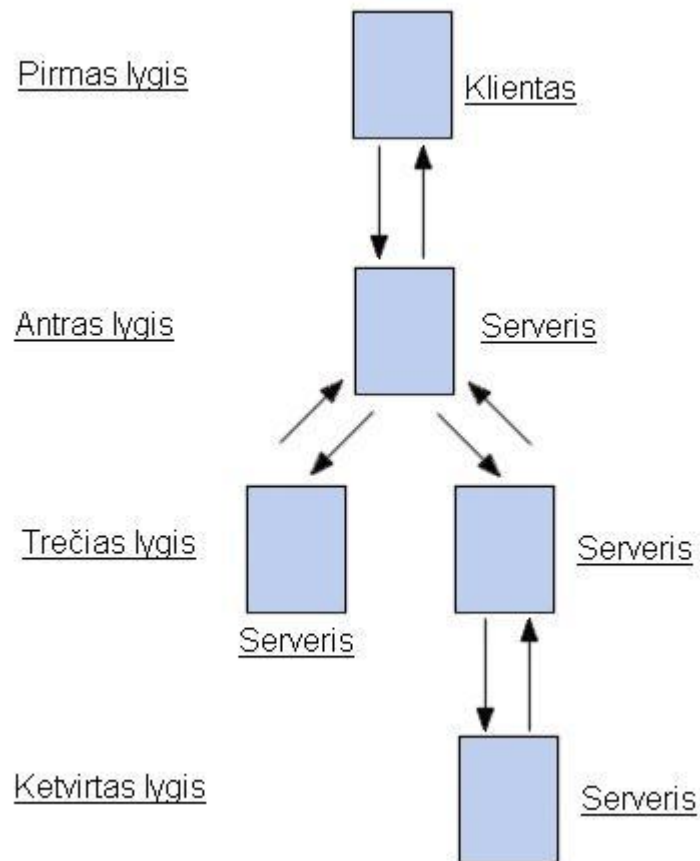
Kliento serverio architektūros modelis yra vienas iš pagrindinių naudojamų programinėje įrangoje dirbančioje kompiuterių tinkluose. Dauguma verslo sistemų sukurtų šiomis dienomis remiasi būtent kliento serverio architektūros modeliu.

Kliento serverio modelyje viena programa (serveris) laukia, kol prie jo prisijungs kita programa (klientas). Taip yra užtikrinamas darbo pasidalinimas tarp paslaugų tiekėjų (serverių) ir paslaugų vartotojų (klientų).

Kiekvienas klientas gali siųsti užklausas serveriui, kuris jas priima, apdoroja ir gražina užklaustą informaciją. Nors ši koncepcija gali būti pritaikyta daugeliui skirtingų problemų spręsti, tačiau bendra architektūra išlieka beveik nepakitusi.

Pats paprasčiausias kliento serverio architektūros variantas apibrėžia tik du skirtingus programų tipus – *klientą* ir *serverį*. Ji gali būti vadinama ir dviejų *lygių kliento serverio architektūra*, kuri leidžia programoms keistis įvairiais resursais.

Sudėtingesnės kliento serverio architektūros atšakos gali apimti ir daugiau lygių (žr. 2 pav). Tuomet viena programa gali būti ir serveris ir klientas vienu metu. Tokia architektūra paprastai naudojama norint tolydžiau ir tarp daugiau kompiuterių paskirstyti sistemos darbus.



2 pav. Kelių lygių kliento serverio architektūra

Kelių lygių kliento serverio architektūros privalumai:

- Leidžia paskirstyti darbus tarp kelių skirtingų kompiuterių.
- Visa informacija saugoma serveriuose, kas paprastai reiškia didesnę duomenų saugumą.
- Kadangi visa informacija saugoma centralizuotai, jos atnaujinimas yra gerokai paprastesnis.

Kelių lygių kliento serverio architektūros trūkumai:

- Didėjant lygiagrečiai veikiančių klientų skaičiui, užklausų į serverį skaičius taip pat didėja, todėl serveris gali būti perkrautas.

- Plono kliento-serverio architektūra didžiąją dalį darbo perkelia serveriui.

1.2 Virtualios telefoninės stotelės sistemos analizė

Taikymo sritis – virtuali telefoninė stotelė ir operatoriaus konsolė mobiliojo ryšio operatoriams. Šiuo produktu siekiama sukurti programinę įrangą atliekančią virtualios telefoninės stotelės funkcijas, grafinę sąsają konfigūruoti jos teikiamoms paslaugoms ir grafinę sąsają skambučių vaizdavimui bei valdymui. Ši programinė įranga turėtų pateikti stotelės sprendimą mobiliojo ryšio paslaugų operatoriams, kurie jos dėka galės teikti šios stotelės paslaugas savo abonentams – smulkiojo ir vidutinio dydžio verslo atstovams.

Pateikiami pradiniai *funkciniai* ir *nefunkciniai* užsakovo *reikalavimai* ir *pageidavimai* kuriamai sistemai. Šie reikalavimai daro įtaką kuriamos sistemos architektūros tipo pasirinkimui, funkcionalumui, vartotojo sąsajos pasirinkimui bei kt.

Pageidavimai kuriamai sistemai (nefunkciniai reikalavimai):

- jNetX OCFS platformos panaudojimas;
- Panaudoti jau turimą Iq NetSolutions VistaPoint Console operatoriaus konsolę;
- Didelio skambučių kiekio palaikymas;
- Vartotojo duomenų saugumas;
- Didelis sistemos patikimumas

Analizuojamo projekto užsakovas – įmonė UAB „Elitnet“, kurios tikslas buvo sukurti kažką naujo šioje srityje. Kaip matyti iš aukščiau pateiktų reikalavimų, o ypač iš pirmojo, tikimasi panaudoti pakankamai naują platformą, kurios gamintojas jNetX yra UAB „Elitnet“ technologinis partneris. Kuriamas produktas turi praplėsti šioje platformoje jau egzistuojančių produktų ratą.

Užsakovas pateikė sekančius funkcinis reikalavimus sistemai:

- Reikia virtualios telefoninės stotelės, teikiančios tokias paslaugas:
 - skambinimo grupių paslauga (angl. hunt groups);
 - sek mane/rask mane paslauga (angl. find me/follow me);
 - skambučio persiuntimo paslauga (angl. transfer call);
 - skambučio persiuntimo su konsultacija paslauga (angl. supervised transfer call);
 - skambučio priparkavimo/išparkavimo paslauga (angl. call park);
 - skambučio nukreipimo paslauga (angl. forward);

- netrukdyti paslaugą (angl. DND);
- skambučių blokavimo paslauga (angl. call screening);
- trumpųjų numerių palaikymas – virtualus privatus tinklas (angl. VPN).
- Reikia grafinės vartotojo sąsajos skirtos stebėti ir valdyti vykstantiems skambučiams.
- Reikia žiniatinkliu paremtos vartotojo sąsajos skirtos stebėti ir valdyti vykstantiems skambučiams.
- Reikia žiniatinkliu paremtos vartotojo sąsajos skirtos konfigūruoti virtualios telefoninės stotelės teikiamoms paslaugoms.
- Reikia galimybės integruotis su trečių šalių valdymo konsolėmis.
- Reikia galimybės nesudėtingai integruotis su skirtinga operatorių įranga.

Detaliau šie reikalavimai ir su jais susiję probleminiai uždaviniai aptariami sekančiame poskyryje.

1.2.1 Probleminiai uždaviniai ir jų sprendimai

Iš ankstesniame poskyryje aprašytų pradinių reikalavimų, žemiau esančioje lentelėje (žr.1 lentelė **Error! Reference source not found.**) pateikti pagrindiniai sistemos realizavimo probleminiai uždaviniai. Siekiant sėkmingai išspręsti šiuos uždavinius, analizės ir projektavimo metu reikės į visus lentelėje pateiktus klausimus.

1 lentelė. Probleminiai sistemos realizavimo uždaviniai

Probleminiai uždaviniai	Pagrindiniai klausimai
<i>Sistemos patikimumas</i>	<ul style="list-style-type: none"> • Kaip užtikrinti nepertraukiamą sistemos darbą
<i>Vartotojo duomenų saugumas</i>	<ul style="list-style-type: none"> • Kuriuos vartotojo duomenis saugoti • Kaip apsaugoti tinklu siunčiamus vartotojo duomenis
<i>Didelio skambučių kiekio palaikymas</i>	<ul style="list-style-type: none"> • Kaip užtikrinti didelį skambučių kiekį virtualioje telefoninėje stotelėje • Kaip užtikrinti didelį operatorių skaičių neapkraunant sistemos
<i>Integracija su trečių šalių valdymo konsolėmis</i>	<ul style="list-style-type: none"> • Kokie protokolai/duomenų formatai turi būti naudojami
<i>Integracija su operatorių įranga</i>	<ul style="list-style-type: none"> • Kokia galima įrangos konfigūracija • Kokių būdu realizuoti lanksčia integraciją su skirtinga operatorių įranga

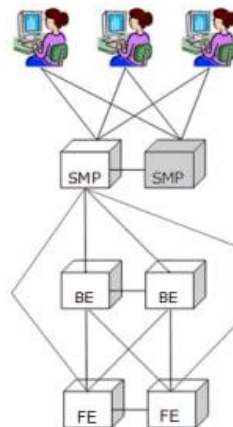
Probleminių uždavinių spėdimai:

Sistemos darbo patikimumas. Sistemos darbo patikimumas yra viena iš svarbiausių savybių telekomunikacijų sistemai, nes nedirbanti sistema neša tiesioginį nuostolį klientams ir nepatogumus bei galimą nuostolį abonentams. Sistemos darbo patikimumas matuojamas procentais laiko kurį ji dirba. Telekomunikacijų programinės įrangos pasaulyje terminas penki devynetai. reiškia, jog sistema turi dirbti 99.999% laiko. Apačioje pateiktoje lentelėje (žr 2 lentelė.), galime matyti kiek laiko prie kokio patikimumo procento sistema neveikia.

1 lentelė. Sistemos pasiekiamumas

Devynetai	Pasiekiamumas	Laikas kurį sistema neveikia	
		valandos/metams	minutės/mėnesiui
2	99 %	87.6	438
3	99.9 %	8.76	44
4	99.99 %	0.88	4.4
5	99.999 %	0.09	0.44
6	99.9999 %	0.01	0.04

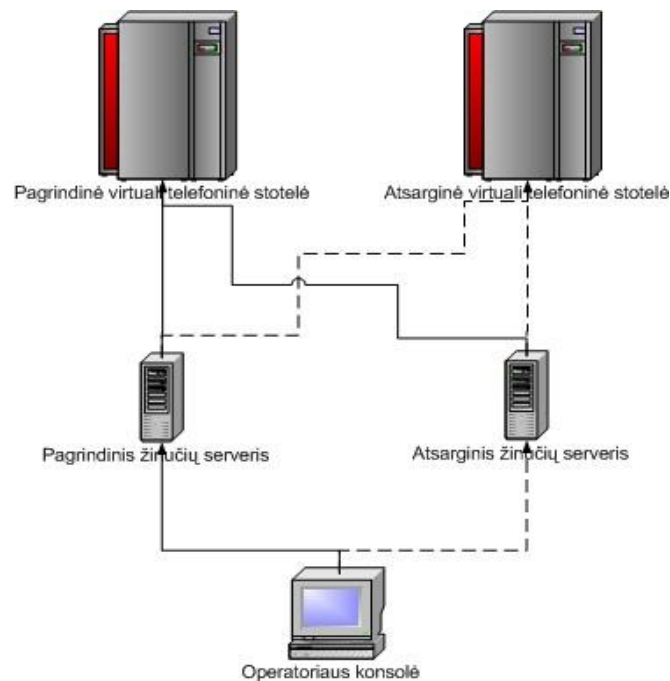
Nepertraukiamą virtualios telefoninės stotelės sistemos darbą padės užtikrinti jNetX OCFS serverio galimybė sujungti kelis serverius lygiagrečiai. Kaip matome žemiau pateiktame paveiksle (žr.2.3 pav.), pagrindiniai serveriai (FE) sujungti su atsarginiais (BE). Bet kurio iš pagrindinių serverių darbui sutrikus, valdymo serveris (SMP) perskirstys darbą atsarginiams serveriams. jNetX OCFS platforma leidžia jungti serverius pakankamai lankčiais, o 3 paveiksle pateikta paprasčiausia konfigūracija, kur yra skirta užtikrinti nepertraukiamam sistemos darbui išlaikant serverius sujungtus į klasterį.



3 pav. Galima jNetX klasterio konfigūracija

Siekiant užtikrinti nepertraukiamą operatoriaus konsolės darbą, buvo nuspręsta naudoti panašią jungimo schemą, matomą 4 paveiksle. Kaip ir jNetX serveryje – bus galima kelis žinučių serverius sujungti su visais virtualios telefoninės stotelės serveriais. Tuo atveju jei dingtų kuris nors iš telefoninės stotelės serverių, žinučių serveris automatiškai persijungtų prie kito serverio, kuriame veikia virtuali telefoninė stotelė.

Operatoriaus konsolėi buvo nuspręsta panaudoti paprastą sprendimą – joje bus galima nurodyti 2 žinučių serverius – pagrindinį ir atsarginį. Dingus ryšiui su pagrindiniu žinučių serveriu, ji bandys prisijungti prie kito nurodyto atsarginio žinučių serverio.



4 pav. Sistemos darbo patikimumą turinti užtikrinanti jungimo schema

Vartotojo duomenų saugumas: Vartotojo duomenų saugumui užtikrinti turi būti skirtas atskiras dėmesys. Nors sistema ir nesaugo itin svarbių vartotojo duomenų, tačiau vartotoją apsaugoti vis tiek būtina. Tam nuspręsta panaudoti *šifravimą*, bei *autentifikaciją*. Autentifikacija apsaugos vartotojų privačius duomenis, nuo kitų vartotojų, kurie galbūt turi piktavališkų tikslų.

- Slaptažodžiai bus saugomi užšifruoti.
- Slaptažodžiai tinklu bus perduodami šifruotame pavidale.
- Įvedami slaptažodžiai nebus rodomi.
- Slaptažodžiai turės nustatytą minimalų ilgį.
- Slaptažodžių šifravimui bus naudojami saugūs algoritmai.
- Perduodami vartotojų prisijungimo vardai taip pat turi būti šifruojami.

- Prisijungimai ir bandymai prisijungti bus saugomi specialiuose failuose, esant neaiškumams bus galimybė sužinoti kas ir kada buvo prisijungęs ar bandė prisijungti ir kokių IP adresu.
- Vartotojai bus automatiškai atjungiami nuo sistemos jei nebus aktyvumo per nustatytą laiką.

Siekiant apsaugoti vartotojus nuo konfidencialios informacijos perėmimo naudojant tinklo srautą stebinčią programinę įrangą, bus naudojamas duomenų šifravimas. Šifravimo algoritmas turėtų pasižymėti keliomis svarbiausiomis savybėmis [5]:

- Šifravimo algoritmas turi užtikrinti ateinančių paketų eilės tvarką.
- Šifravimo algoritmas turi patikimai šifruoti duomenis ir padaryti atkodavimo operaciją naudojant „grubios jėgos“ algoritmą kaip įmanoma sudėtingesnį užimamo laiko prasme.

Išanalizavus esamus šifravimo algoritmus, buvo nuspręsta pasirinkti AES. Jis yra gana greitas, palaiko visus mūsų programinės įrangos keliamus reikalavimus, be to praktiškai laikomas nenulaužiamu [6].

Integracija su trečiųjų šalių valdymo konsolėmis: Kaip žinia, nemažai įmonių dabar naudojami CRM programine įranga. Šios programinės įrangos dėka, jos gali geriau aptarnauti savo klientus. Dalis šių CRM turi integruotą telefonijos posistemę, kuri dažniausiai gali būti suderinta su trečiųjų šalių programine įranga. Taip pat yra nemažai programinės įrangos, kuri yra skirta trečiųjų šalių skambučiams valdyti. Ji yra naudojama įvairiose kompanijose.

Norint, jog prie telefoninės stotelės galėtų prisijungti ne tik mūsų kuriama operatoriaus konsolė, bet ir kitų gamintojų programinė įranga, buvo nuspręsta atsižvelgti į komunikacijos protokolą tarp virtualios telefoninės stotelės ir operatoriaus konsolės – reikia rinktis atviro standarto populiarių protokolą.

Peržvelgus esamus protokolus pasirinktas CSTA III XML. Tai CSTA [7] standarto vienas iš pašakių. Jis užtikrina tokias paslaugas kaip:

- 26 skambučio valdymo funkcijos: skambinti, atsiliepti ir daug kitų.
- 6 su skambučiu susijusias savybes: vartotojo duomenų siuntimas ir kt.
- 19 loginių įrenginių savybių: netrukdyti, skambučio peradresavimas, t.t.
- 23 fizinių įrenginių savybės: rašymas į įrenginio ekraną, kt.
- 5 savybių apsikeitimo funkcijos: savybių nustatymas, kt.
- 4 funkcijos esamai būsenai gauti: egzistuojančių skambučių įrenginyje gavimas, kt.
- 3 monitoravimo savybės: užsiregistruoti įvykių gavimui, kt.

- 17 balso paslaugų: klausančiajam, DTMF tonai, perspėjimai ir žinutės, kt.

CSTA standarto pagalba dar galima realizuoti ir tokias paslaugas kaip maršrutizavimas, informacijos prikabinimas, palaikymas, duomenų surinkimas, paskyros ir autorizacija. Toks platus palaikomų paslaugų spektras, leidžia daryti prielaidą, jog ateityje plečiant virtualią telefoninę stotelę nereikės pereiti prie naujo protokolo. Dar viena iš gerų šio standarto savybių yra ta, jog programa laikoma pilnai suderinama su CSTA standartu, net jei nepalaiko visų jo savybių [8].

Kaip alternatyva dar svarstytas ir Microsoft TAPI, tačiau jo buvo atsisakyta dėl to, jog jis skirtas Microsoft Windows platformai.

Integracija su operatorių įranga: pagrindinė integracijos problema yra priderinti virtualią telefoninę stotelę prie operatoriaus MSC. Kadangi skirtingų gamintojų ar net to paties gamintojo skirtingi MSC gali dirbti su skirtingais protokolais, reikia užtikrinti, jog kuriama virtuali telefoninė stotelė galėtų dirbti su kaip įmanoma daugiau protokolų vienu metu.

Iš šiuo metu naudojamų protokolų skambučių komutavimui populiariausi yra CAMEL Ph2 ir INAP CS1+. Šių protokolų palaikymą užtikrinti padės jNetX OCFS platformos sukurta aukšto lygio tinklo abstrakcija MPCC, kuri paslepia tikrąjį protokolą su kuriuo dirba jNetX OCFS serveris. Kuriant virtualią telefoninę stotelę reiks naudoti šį abstrakcijos lygį, o ne dirbti su kiekvienu protokolu atskirai, todėl užteks atitinkamai sukonfigūruoti platformą ir virtualiai telefoninei stotelei bus nesvarbu, prie kurio gamintojo ir kokį protokolą palaikančios MSC ji yra prijungta. CAMEL Ph1 ir INAP CS1 nebus palaikomas, dėl savo ribotų galimybių (šių protokolų pagalba negalima realizuoti viso siekiamo virtualios telefoninės stotelės funkcionalumo).

Didelio skambučių kiekio palaikymas: kadangi sistema dirbs pas mobiliojo ryšio operatorius turinčius daug abonentų, tai reikia manyti, jog ir virtuali telefoninė stotelė turės aptarnauti nemažai skambučių. Iš kilo klausimas kiek skambučių per sekundę reikės virtualiai telefoninei stotelei, kiek jų sugebės aptarnauti jNetX OCFS platforma ir kokios įrangos reikės. Pasitarus su vietiniais operatoriais buvo nuspręsta, jog virtualiai telefoninei stotelei turėtų pilnai pakakti iki 100 CPS. Čia yra atsižvelgiama į tai, jog ne visi sistemos įvykiai bus raportuojami į jNetX OCFS platformą, nes galima MSC stotį sukonfigūruoti taip, jog ši raportuotų įvykius tų telefonų skambučiams, kuriuos turi komutuoti mūsų kuriama programinė įranga.

Analizės metu paaiškėjo, jog jNetX platformą galima jungti į klasterį – keli aplikacijų serveriai veikia lygiagrečiai, dėl ko sistemos našumą galima padidinti pakankamai nesunkiai. 2.3.

paveiksle matome vieną iš galimų jNetX serverio jungimo į klasterį variantų. SMP – valdymo serveriai, FE – dirbantys aplikacijų serveriai, BE – atsarginiai aplikacijų serveriai.

Taip pat buvo analizuojama grafinės vartotojo sąsajos sparta. Galima teigti, kad šiuo metu ji tikrai nepajėgia apdoroti reikiamo skambučių kiekio. Išanalizavus jos esamą kodą specialiomis priemonėmis, buvo nustatyta kuriose vietose, ji užtrunka daugiausia laiko ir ką reikia optimizuoti.

1.2.2 Realizavimo technologijos parinkimas

Virtualios telefoninės stotelės realizavimo technologija yra nulemta vieno iš nefunkcinių reikalavimų – reikia naudoti jNetX OCFS platformą. Šioje platformoje programavimas vyksta Java kalba. Programuoti teks remiantis JAIN SLEE specifikacija, kas reiškia, jog teks sukurti servisų blokus (angl.: Service Building Block, SBB) ir resursų adapterius (angl.: Resource Adapter).

Žinučių serveriui sukurti pasirinkta JAVA programavimo kalba, kuri remiasi „valdomo kodo“ koncepcija. Ši koncepcija išsprendžia nemažai problemų susijusių su atminties valdymu, masyvų ribų valdymu ir t.t. Java gali veikti beveik visose egzistuojančiose operacinėse sistemose. Programuoti JAVA kalba yra gana nesudėtinga, nes egzistuoja daugybė puikių įrankių – kūrimo, testavimo, kodo analizės ir kt.

Grafiinei vartotojo sąsajai buvo pasirinkta C++ programavimo kalba. Tai lėmė sprendimas naudoti egzistuojančią grafinę sąsają IQ NetSolutions VistaPoint Console, kuri iki šiol buvo naudota aparatinių telefoninių stotelių skambučių valdymui.

Duomenų bazės serveriui pasirinkta SQL kalba. Tai gali būti MySql, Microsoft SQL Express 2005 ar panaši duomenų bazių programinė įranga. Taip pat tikėtina, jog tai gali būti viena iš darbinėje atmintyje saugomų duomenų bazių, tokių kaip Oracle TimesTen. Tokia duomenų bazė gali būti naudojama tada, jei netenkins įprastų duomenų bazių programinės įrangos sparta.

Komunikacijos protokolas tarp virtualios telefoninės stotelės, žinučių serverių ir operatoriaus konsolės bus CSTA III XML.

Virtualios telefoninės stotelės konfigūravimo grafiinei sąsajai pasirinktos **2 technologijos** aprašytos žemiau. Pačios konfigūravimo grafinės sąsajos architektūros darbe detaliau nenagrinėsime.

Jetty – tai visiškai Java realizuotas, atviro kodo, standartais paremtas, daug galimybių turintis žiniatinklio serveris. Svarbios šio serverio savybės yra tai, jog jis gali veikti kaip visiškai atskiras arba kaip dinaminio turinio serveris už išskirto serverio (pvz. Apache) arba kaip

komponentas integruotas į kitą Java programą. Ši savybė gali būti panaudota ateityje, jei prireiks realizuoti daugiau biznio logikos konfigūravimo sąsajoje ir tam bus nuspręsta naudoti kurį nors iš J2EE serverių.

Tapestry – tai žinatinklio kūrimo karkasas, skirtas Java programavimo kalbai. Šio produkto dėka internetinių puslapių kūrimas tampa paremtu komponentais, o tai padeda žymiai sutrumpinti kūrimo laiką. Tiesa, kaip ir dirbant bet kuriuo karkasu, galima susidurti su problemomis kai prireikia to ko nemoka karkasas, o kartais netgi draudžia.

1.2.3 Egzistuojančios sistemos ir jų palyginimas

Šiuolaikinėje rinkoje jau yra nemažai virtualių telefoninių stotelių. Tarpusavyje jos gan stipriai skiriasi savo kaina, teikiamomis paslaugomis, konfigūravimo galimybėmis, taip pat ir skirtumų lyginant su aparatinėmis telefoninėmis stotelėmis:

- Panašių galimybių virtuali stotelė yra žymiai pigesnė.
- Virtualios stotelės aptarnavimo kaina yra mažesnė.
- Minutė pokalbių su virtualia stotele kainuoja daugiau.
- Virtualios stotelės neturi bendrų įeinančių linijų.
- Virtualios stotelės turi galimybę paradresuoti skambučius į bet kokį telefoną.
- Virtualios stotelės turi galimybę dirbti su plačiai „išmėtytais“ telefonais, nebūtinai esančiais vienoje vietoje.
- Virtualios stotelės turi didesnes praplėtimo galimybes.
- Virtualios stotelės paprastai gali apdoroti daugiau skambučių.
- Virtualios stotelės nebrangiai, lyginant su aparatinėmis, įgyvendina ACD eiles.
- Virtualios stotelės turi galimybę būti administruojamos per nuotolį, aparatinės su šia galimybe yra gana brangios.

Bendrai paėmus nėra daug vertų dėmesio produktų su geru aprašymu, dėl to jog jie orientuoti į mažą pirkėjų skaičių – telekomunikacijos paslaugų tiekėjus, kurie dažniausiai glaudžiai bendradarbiauja su tokio pobūdžio programinės įrangos kūrėjais norėdami programinę įrangą prisitaikyti savo poreikiams. Egzistuojančių sprendimų palyginimui buvo atrinkti didžiausią pasaulinės rinkos dalį besidalinantys produktai:

- **BroadSoft BroadWorks** – labai išplėtotas ir didelių galimybių produktas, aprėpiantis kur kas didesnę paslaugų sferą nei mūsų kuriamas produktas.

- **Sylantro Synergy Application Feature Server** – daug paslaugų teikiantis produktas telekomunikacijų paslaugų operatoriams.

Verta paminėti, jog nė vienas iš šių produktų nėra tikslus atitikmuo kuriamam produktui. Abu produktai yra atskiros platformos, ir aparatinė įranga yra derinama prie šių produktų.

BroadSoft „BroadWorks“ sprendimas. Palyginti neseniai BroadWorks pristatė programinės įrangos paketą aprėpiantį labai plačią paslaugų sferą. Šis produktas nėra visiškai analogas mūsų kuriamam, kadangi jis panašesnis į atskirą didžiulę platformą, skirtą ne tiek integruotis su operatorių turima įranga, kiek pats tapti operatorių įrangos pagrindu, tačiau pagal siūlomas telekomunikacijų paslaugas galima labai puikiai įsivaizduoti į ką turėtų lygiuotis mūsų produktas. Be to ir produkto apibūdinimas teigia, jog tai integruota sistema skirta balso aplikacijoms bei kitoms telekomunikacijų paslaugoms.

Pagrindinės savybės[3], [4]:

- Grafinės vartotojo sąsajos teikiamos paslaugos. CommPilot web sąsaja:
 - CommPilot Group internetinio portalo dėka įmonės gali pačios sukonfigūruoti ir valdyti savo paslaugas teikiamas į grupes suskirstytam personalui, taip sumažindamos paslaugų tiekėjo darbą ir sąskaitas susijusias su konfigūracijos pakeitimais.
 - CommPilot Personal interneto portalo dėka vartotojai gali patys sukonfigūruoti sistemą pagal savo poreikius.
 - Šalia CommPilot yra pateikiamas ir Call Manager, kurio dėka vartotojai gali valdyti išeinančius ir įeinančius skambučius, su tokiomis savybėmis kaip telefonų sąrašai arba integracija su Microsoft Outlook.

Sistemos teikiamos paslaugos:

- Skambinti (Dial);
- Atsiliepti (Answer);
- Baigti skambutį (Release);
- Užlaikyti skambutį (Hold);
- Aklas skambučio persiuntimas (Blind Transfer);
- Skambučio persiuntimas (Supervised Transfer);
- Konferencija (Conference);
- Telefonų sąrašai;
- Microsoft Outlook integracija;
- Skambučio peradresavimas: visada, užimta, neatsako;

- Įspėjimas apie skambutį;
- Skambučio parkavimas ir išparkavimas;
- Paskutinio numerio perrinkimas;
- Skambučio trasavimas;
- Skambučio laukimas, laukimo užbaigimas;
- Selektyvus ir anoniminio skambučio atmetimas;
- Greitasis skambinimas (Speed Dial);
- Netrukdyti funkcija (DND);
- Nustatomi pasveikinimai;
- Balso žinutė – atskambinti perspėjimas;
- Balso žinutė – laukiama perspėjimas;
- Balso žinutė – perspėjimas;
- Paskyrų ir autorizacijos kodai;
- Skambučių perėmimas;
- Skambinimo planai;
- Konfigūruojami paslaugų kodai;
- Skambinimo grupė;
- Momentinės n-dalyvių konferencijos;
- Pranešimas per garsiakalbį;
- Muzika užlaikius skambutį;
- Balso virtualus privatus tinklas – Voice VPN;
- Mažiausios kainos sujungimas;
- Tinklo URL adreso skambinimas;
- Įspėjimas apie avarinį skambutį (911, 112, 01 ir kt.);
- Abonento vietos identifikavimas.

Kitos savybės:

- Sun Solaris 2.8 ar naujesnė operacinė sistema;
- Kelių serverių apjungimas į vieną;
- Lengvas praplėtimas, paskirstyta architektūra;
- Darbas dideliais apkrovimais – 1.7 milijono vartotojų ir 7 milijonai skambučių piko metu;

- Išsamūs skambučių aprašai;
- Transakcijų įrašai;
- Komandinės eilutės sąsaja;
- Konfigūruojamos laiko juostos;
- CORBA sąsaja;
- SLL.

Sylantro Synergy Application Feature Server sprendimas. Sylanro siūlo virtualią telefoninę pastotę, kuri yra puiki alternatyva aparatine įranga pagrįstiems sprendimams. Kaip ir BroadSoft sprendimas, tai yra didžiulė atskira platforma, savo siūlomomis galimybėmis šiuo tuo panaši į BroadSoft sprendimą. *Sylantro sprendimo pagrindiniai aspektai:*

- **ComPortal II** – žiniatinkliu paremta vartotojo sąsaja, pasižyminti savybe laisvai ją keisti ir konfigūruoti pagal vartotojo poreikius. Ši sąsaja suteikia visą reikiamą informaciją galutiniams vartotojams, administratoriams, paslaugų perpardavinėtojams, bei jų operatoriams.
- Sąsaja galutiniams vartotojams pasižymi tokiais galimybėmis kaip priėjimu prie direktorijos, informacija apie skambučių istorijas, galimybe konfigūruoti tam tikras operatoriaus paslaugas, tokias kaip sek mane/rask mane (angl. Find-me/Follow-me), bei galimybe konfigūruoti telefono savybes iš bet kurios vietos, kur yra pasiekiamas internetas.
- Vietinio administratoriaus grafinė sąsaja leidžia greitai ir paprastai įtraukti naujus vartotojus, šalinti esamus, keisti duomenis apie juos, konfigūruoti telefonus bei duomenis apie juos. Administratorius taip pat gali konfigūruoti paslaugas gaunamas iš paslaugų tiekėjo, taip taupydamas lėšas ir laiką. Yra galimybė keisti tam tikrus vartotojo grafinės sąsajos parametrus. Administratoriaus grafinė sąsaja yra visiškai atskirta nuo vartotojo grafinės sąsajos.
- Paslaugų perpardavinėjo grafinė sąsaja leidžia konfigūruoti su tiekėju susijusius duomenis, atributiką, su reklama susijusią informaciją. Taip pat yra galimybė konfigūruoti teikiamas paslaugas, nepriklausomai nuo pagrindinio paslaugų tiekėjo.
- Tinklo operatoriaus grafinė sąsaja leidžia konfigūruoti viską kas susiję su teikiamomis paslaugomis programinės įrangos lygyje. Ši sąsaja suteikia galimybę konfigūruoti papildomus programinės įrangos modulius, kurie gali būti pateikti ir trečių šalių, taip sudaroma galimybė pridėti net tokias visiškai originalias paslaugas, kurių Sylanro

sprendimas neturi. Yra galimybė keisti su programine įranga susijusią atributiką, bei viską kas susiję su reklama.

- **ComConference** – tai paslauga leidžianti lengvai sukurti daug vartotojų sudarančius konferencinius skambučius, pasinaudojant per žiniatinklį prieinama vartotojo sąsaja.
- **ComCarriedge** – Šios paslaugos dėka vartotojai gali visada likti prieinami kitiems, nesvarbu kas benutiktų. Tai rask mane/sek mane paslauga, kurią galima susikonfigūruoti taip, jog keli telefonai priklausantys vartotojui skambėtų paeiliui pagal tam tikrą sąrašą arba tiesiog visi iškart. Šios paslaugos tikrai sunku atsisakyti tiems vartotojams kurie pajaučia jos teikiamą naudą.
- **ComSchedule** – tai įrankis dar labiau praplečiantis ComCierge teikiamą naudą, leisdamas nurodyti laikus, kuriuo metu kuriuo telefonu juos gali pasiekti kiti vartotojai. Jis leidžia konfigūruoti kas tuo metu gali juos pasiekti.
- **Voice Portal** – tai balsu konfigūruojamos paslaugos, jos leidžia konfigūruoti paslaugas net kai nėra galimybės prieiti prie interneto. Tai tokio tipo paslauga, kai jūs paskambinate ir balsu pranešama, jog paspaudus ta ar kita klavišą ant telefono, bus įvykdytas vienoks ar kitoks veiksmas. Viską kas susiję su šia paslauga sukonfigūruoją tinklo operatorius, o vėliau vartotojas ją gali naudoti atlikti nustatytiems veiksams.
- **ComOffice** – tai integracija su Microsoft Outlook. Jos dėka galima sinchronizuoti kontaktus tarp Outlook ir operatoriaus bazės, šios integracijos dėka jūs galite atlikti tokius veiksmus kaip skambinti tiesiais iš Outlook. Ši paslauga yra gan aktuali verslo vartotojams, nes leidžia viską pasiekti iš vienos programos.
- **ComMerchant** – leidžia konfigūruoti skambučių valdymą ir jų kiekio padidėjimą, kas leidžia tinklo operatoriams geriau valdyti situaciją, padidinti produktyvumą. Operatorius gali konfigūruoti sistemos resursų paskirstymą taip, jog didelių apkrovimų metu jo paslaugos būtų prieinamos, o ne tiesiog skambinančiuosius nukreipti į balso pašto dėžutes ar groti užimtos linijos signalą. Šios naujoviškos paslaugos dėka operatorius gali pateikti kokybiškesnes paslaugas, padidinti sėkmingų skambučių skaičių ir išvengti papildomos įrangos diegimo vartotojo pusėje siekiant susidoroti su šiomis neįprastomis situacijomis.
- **ComConsole** – tai papildoma programa skirta Microsoft Windows operacinės sistemos vartotojams. Ji suteikia lengvai naudojamą ir sparčią prieigą prie operatoriaus teikiamų paslaugų bei galimybę turėti tarsi savo atskirą skambučių centrą, ir su juo dirbančius operatorius. Šios konsolės dėka net ir didelės įmonės gali visiškai atsisakyti įprastų telefoninių stotelių ir internetine telefonija pagrįstų telefoninių pastočių.

- **ComRIO** – ši paslauga leidžia įgyvendinti „nutolusio ofiso idėją“ iš bet kurios vietos. Tam tereikia priejimo prie interneto bei laidinio ar mobilaus telefono – ComRIO paslauga gali nukreipti įeinančius ir išeinančius skambučius per jūsų ofiso telefoną, taip nuslėpdama tikrąją jūsų buvimo vietą ir padarydama jus prieinamą per jūsų ofiso telefoną.
- **ComTraveler** – suteikia galimybę prieiiti prie Sylanro paslaugų per konfigūruojamą WAP vartotojo sąsają. Mobilūs vartotojai gali valdyti tokias funkcijas kaip skambučių istorija, rask mana/sek mane.

Programų sistemų savybių palyginimas. Apžvelgtas programos galima palyginti tik remiantis viešai prieinama informacija, kuri yra labiau reklaminio pobūdžio. Pabandyti paminėtą programinę įrangą nebuvo galimybės, kadangi demonstracinės versijos nėra nepateikiamos, o norint sistemas išbandyti, reiktų kreipti į pačius gamintojus turint tikslą jas įsigyti.

Panašu, jog geriau išdėstyta grafinę sąsają turi BroadSoft produktas, kadangi ji yra suskirstyta į dalis priklausomai nuo jos vartotojo – vartotojai, administratoriai, paslaugų perpardavinėtojai, bei paslaugų tiekėjai turi atskiras sąsajas. Iš jų kiekviena aprėpia visas programos funkcijas prieinamas to tipo vartotojams. Tuo tarpu Sylanro sprendimas sąsajas skirsto pagal paslaugas, o tai nėra labai patogu vartotojui, nes norint pasiekti kažkokią funkciją, reikia pasileisti kitą vartotojo sąsają.

Apžvelgiant programos teikiamas paslaugas galima teigti, jog BroadSoft sprendimas turi daugiau galimybių, nes jis teikia daugumą dabar egzistuojančių telekomunikacijos paslaugų. Tačiau reikia nepamiršti, kad Sylanro sprendimas nepateikia laisvai prieinamos techninės specifikacijos, o tik abstrakčiai apibrėžia produkto galimybes.

Sylanro sprendimas turi vieną patrauklią galimybę – tai WAP vartotojo sąsaja, kas patogu jei esate kitoje šalyje, kur negalite pasiekti pilnaverčio interneto, o tik vietinių mobiliųjų paslaugų operatorių teikiamą internetą, kuris dažniausia apsiriboja WAP.

Remiantis turimais duomenimis galima teigti, jog *BroadSoft* produktas turi daugiau *išskirtinių savybių*, tokių kaip: komandinės eilutės sąsaja, CORBA sąsaja, kelių serverių apjungimas į vieną. Šios savybės yra naudingos paslaugų operatoriams, kurie ir renkasi produktą, todėl nenuostabu, jog BroadWorks yra vienas iš populiariausių savo tarpe.

Produkto kainos nėra pateikiamos, bet jos turi didelę įtaką renkantis, tai ypač aktualu mažesniems operatoriams turintiems mažiau lėšų, kadangi tokio tipo produktai yra labai brangūs ir kainos paprastai nebūna mažesnės nei šešiaženklės. Lieka neaiškus ir apmokestinimo principas – neretai yra apmokestinama pagal abonentų skaičių, kas esant dideliame jų skaičiui gali labai pakelti kainą.

1.3 Analizės išvados

1. Apžvelgti esminiai akcentai susiję su sistemos struktūros bei architektūros kūrimu: JAIN SLEE specifikacija, kliento serverio architektūra.

2. Apibrėžta sistemos taikymo sritis – virtuali telefoninė stotelė ir operatoriaus konsolė mobiliojo ryšio operatoriams, kuris savo teikiamomis papildomomis paslaugomis padidins mobiliojo ryšio operatorių pelną ir suteiks jų abonentams naujų paslaugų

3. Išanalizavus užsakovo iškeltus reikalavimus sistemai, nustatyti penkti probleminiai sistemos realizavimo uždaviniai: sistemos darbo patikimumas, vartotojo duomenų saugumas, didelio skambučių kiekio palaikymas, integracija su trečių šalių valdymo konsolėmis, integracija su operatorių įranga. Šiems probleminiams uždaviniams išspręsti buvo pateikti jų galimi sprendimai.

4. Atsižvelgus į sistemai keliamus vartotojo reikalavimus ir numatomus poreikius jų realizavimui, pasirinktos sistemos realizavimo technologijos – jNetX OCFS, Java, C++, Jetty, Tapestry, SQL, CSTA III XML.

5. Apžvelgti egzistuojantys sprendimai, palygintos jų savybės. Įsitikinta, jog rinkoje nėra daug analogiškų produktų, o virtuali telefoninė stotelė ras savo nišą.

2 PROJEKVINĖ DALIS

Šiame skyriuje trumpai aptariami projekto tikslai, sprendžiami uždaviniai bei priimti techniniai sprendimai. Vėliau apžvelgiamas bendras sistemos architektūros vaizdas, tuo pačiu pateikiamos esminės sistemos architektūrą paaiškinančios diagramos, pateikiamas sistemos duomenų modelis ir aptariama projekto išeiga.

Skyrius yra akcentuotas į tas sistemos dalis prie, kurių buvo dirbta: grafinę vartotojo sąsają operatoriams, bei virtualios telefoninės stotelės programinę įrangą veikiančią jNetx OCFS SLEE serveryje. Dokumente nedaug užsiminama apie žinučių serverį ir CSTA III XML resursų adapterį, bei visiškai nedetalizuojamos grafinės vartotojo sąsajos paremtos žiniatinkliu, viena iš kurių skirta operatoriams, o kita serverio konfigūravimui.

2.1 Projekto tikslai

Šiuo projektu siekiama sukurti programinę įrangą skirtą praplėsti mobiliojo ryšio operatorių teikiamų paslaugų spektrą tokiomis paslaugomis, kurios anksčiau buvo prieinamos tik laidiniams telefonams prijungtiems prie aparatinės telefoninės stotelės. Šias paslaugas apima:

- Trumpieji numeriai
- Skambučio parkavimas (angl.: Park call) ir išparkavimas (angl.: Unpark call);
- Svetimo skambučio pakėlimas (angl.: Pickup call);
- Skambinimo grupės (angl.: Hunt groups);
- Skambučio peradresavimas (angl.: Forward call);
- Skambučio persiuntimas (angl.: Blind Transfer);
- Skambučio persiuntimas su konsultacija (angl.: Supervised transfer);
- Sek mane/rask mane paslauga (angl.: Find me/Follow me);
- Muzika užlaikius skambutį (angl.: Music on hold), bei galimybė virtualios telefoninės stotelės administratoriui ją parinkti;
- Virtualus privatus tinklas (angl.: Voice VPN).

Sistemos tikslas yra emuliuoti aparatinės telefoninės stotelės darbą, suteikti galybę konfigūruoti šią stotelę, stebėti bei valdyti joje vykstančių pokalbius. *Esminiai kuriamos sistemos komponentai:*

- Virtuali telefoninė stotelė – programa veikianti telekomunikacijų paslaugų tiekėjo serveriuose, emuliuojanti įprastos aparatinės telefoninės stotelės darbą, bei visas jos

teikiamas paslaugas, tokias kaip: trumpieji numeriai, skambinimo grupės, konferencinis skambutis, ir kt.

- Operatoriaus konsolė – grafinė sąsaja (programa Windows terpei, paremta žiniatinkliu) skirta stebėti, bei valdyti vykstančius skambučius.
- Grafinė sąsaja skirta konfigūruoti virtualią telefoninę stotelę. Šio sąsajos dėka vartotojas gali susikonfigūruoti kokių norimas virtualios telefoninės stotelės paslaugas.

2.2 Sprendžiami uždaviniai

Šiame skyriuje apžvelgiami projekto tikslai iš programų inžinerijos pusės. Toliau pateikiamas sistemos sprendžiamų uždavinių sąrašas su trumpais paaiškinimais:

- Reikia realizuoti grafines vartotojo sąsajas skambučiams valdyti ir virtualiai telefoninei stotelei konfigūruoti.
- Vartotojo sąsajos turi būti paprastos ir lengvos naudotis. Vartotojui svarbiausia, kad sistema, kuria jis naudojasi, būtų nesudėtinga. Taigi, svarbu sistemos funkcionalumą kuo paprasčiau pateikti sistemos naudotojui.
- Sistema turi atlikti veiksmus analogiškus aparatinėms telefoninėms stotelėms:
 - Trumpieji numeriai;
 - Skambučio parkavimas (angl.: Park call) ir išparkavimas (angl.: Unpark call);
 - Svetimo skambučio pakėlimas (angl.: Pickup call);
 - Skambinimo grupės (angl.: Hunt groups);
 - Skambučio peradresavimas (angl.: Forward call);
 - Skambučio persiuntimas (angl.: Blind Transfer);
 - Skambučio persiuntimas su konsultacija (angl.: Supervised transfer);
 - Sek mane/rask mane paslauga (angl.: Find me/Follow me);
 - Muzika užlaikius skambutį (angl.: Music on hold), bei galimybė virtualios telefoninės stotelės administratoriui ją parinkti;
 - Virtualus privatus tinklas (angl.: Voice VPN).
- Virtualios telefoninės stotelės sistema turi užtikrinti pakankamą greitaveiką. Kadangi tai telekomunikacijoms skirta sistema ja naudosis daug vartotojų, todėl kuriant sistemą reikia į tai atsižvelgti.

- Reikia užtikrinti didelį sistemos pasiekiamumą. Tiek virtuali telefoninė stotelė, tiek ir jos grafinės sąsajos vartotojui turi būti pasiekiamas kaip įmanoma didesnis laiko tarpas.

Aukščiau pateikti tik bendri sistemos sprendžiami uždaviniai, kurie realizacijos metu gali būti toliau smulkinami ir koreguojami. Pagal šiuos pateiktus uždavinius taip pat galima susidaryti bendrą sistemos panaudos atvejų schemą.

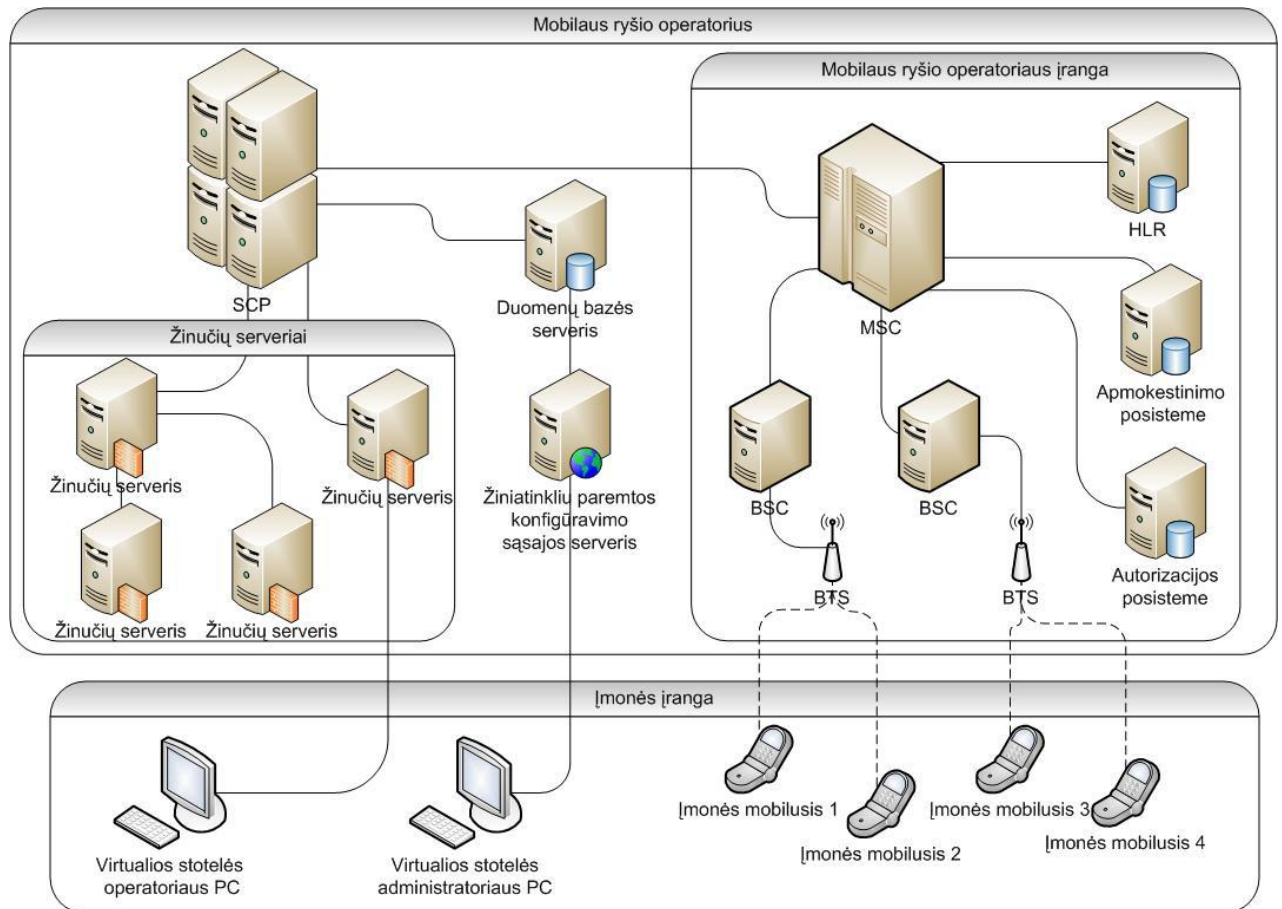
Dėl šių uždavinių išskylantys papildomi funkciniai ir nefunkciniai sistemos reikalavimai bei kitos problemos bus aptariamoms šioje ir kitoje darbo dalyse.

2.3 Priimti techniniai sprendimai

Įvertinus visus projektui iškeltus reikalavimus ir apribojimus, sukūrus ir išbandžius prototipą, nuspręsta panaudoti 3 lygių kliento serverio architektūros analogą apkrovos paskirstymui tarp virtualios telefoninės stotelės ir operatoriaus konsolių, tam panaudojant žinučių serverius. Virtuali telefoninė stotelė bus kuriama jNetX OCFS platformai. Operatoriaus konsolė bus perdaryta iš jau turimos IQ NetSolutions konsolės, pritaikius ją darbui su CSTA III XML protokolu, bei pertvarkius grafinę sąsają taip, jog ši atitiktų virtualios telefoninės stotelės teikiamas paslaugas.

2.4 Diegimo aplinka

Diegimo aplinkos schema pateikta 5 paveiksle.



5 pav. Diegimo aplinkos schema

- SCP serveriuose bus įdiegtas jNetX SCP serveris, CSTA III XML resursų adapteris, bei jVPBX virtualios telefoninės stotelės programinė įranga taip pat atskiruose serveriuose jNetX SMP serveris. Visi šie serveriai veiks Linux operacinėje sistemoje. Serveriuose, kuriuose bus įdiegta jNetX SCP, privalės būti įdiegta ir Ulticom aparatinė ir programinė įranga užtikrinanti E1 ryšio kanalą ir priėjimą prie MSC per jį. SCP serverių gali būti vienas ir daugiau, nes jie yra identiškai sukonfigūruojami ir jungiami pagal tokią schemą, kuri užtikriną apkrovos paskirstymą (load balancing), jNetX OCFS tai palaiko. Šiam serveriui reikalinga profesionali aparatinė serverio įranga, kuri turėtų pasižymėti galimybėmis pakeisti diskus bei praplėtimo plokšte veikiančiai sistemai (Hot-Swap), galimybe įdiegti bent du procesorius, galimybė įdiegti bent du vienas kitą dubliuojančius maitinimo šalinius. Visos šios savybės reikalingos norint užtikrinti nepertraukiamą paslaugų teikimą. Tikslūs techniniai

reikalavimai bus formuojami priklausomai nuo operatoriaus turimo klientų skaičiaus ir kitų faktorių.

Preliminarūs reikalavimai: bent du serveriai, turintys po du Intel Pentium Xeon 2Ghz (ar atitinkamos spartos AMD analogą) keturių branduolių procesorius, 8 GB darbinės atminties, du 80 Gb diskus, bent 1 Gb/s tinklo plokštė, įdiegta Ulticom E1 plokštė. Šiuose serveriuose turėtų veikti Linux operacinė sistema, su 2.6x versijos branduoliu, šioje operacinėje sistemoje turi būti įdiegta 1.6 versijos Java virtuali mašina, bei Ulticom SS7 steką palaikanti programinė įranga.

- Žinučių serveryje bus įdiegta žinučių serverio programinė įranga taip pat serveryje veiks Linux operacinė sistema. Prie CSTA III XML resursų adapterio besijungiančiam serveriui keliami nemažesnis reikalavimai nei SCP serveriui, tikslūs reikalavimai priklausys nuo konkretaus operatoriaus.

Preliminarūs reikalavimai: serveris turintis du Intel Pentium Xeon 2Ghz (ar atitinkamos spartos AMD analogą) keturių branduolių procesorius, 8 GB darbinės atminties, du 80 Gb diskus, bent dvi tinklo plokštės po 1 Gb/s. Rekomenduojama SCP ir išorinius žinučių serverius (besijungiančius prie žinučių serverio kuris jungiasi prie CSTA III XML resursų adapterio) laikyti atskiruose potinkliuose. Išoriniams žinučių serveriams, priklausomai nuo jų apkrovos gali pakakti pakankamai silpnų asmeninių kompiuterių, reiktų tiktai pasirūpinti, jog jie turėtų bent 1 GB darbinės atminties, bei bent 100 Mb/s tinklo plokštę. Bet kurio tipo žinučių serveryje gali veikti tiek Linux tiek Windows operacinės sistemos, svarbu jog būtų įdiegta 1.6 versijos Java virtuali mašina.

- Duomenų bazės serveris – šiame serveryje veiks Linux arba Microsoft Windows operacinė sistema ir duomenų bazės programinė įranga. Tai gali būti MySQL, Microsoft SQL Express 2005 ar panaši duomenų bazių programinė įranga. Taip pat tai gali būti viena iš darbinėje atmintyje saugomų duomenų bazių, tokių kaip Oracle TimesTen in Memory Database. Tokia duomenų bazė gali būti naudojama tuo atveju, jei netenkina įprastų duomenų bazių programinės įrangos sparta. Šiam serveriui taip pat keliami aukšti reikalavimai, ypač patikimumo.

Preliminarūs reikalavimai: serveris turintis du Intel Pentium Xeon 2Ghz (ar atitinkamos spartos AMD analogą) keturių branduolių procesorius, bent 8 GB darbinės atminties (kuo daugiau atminties, tuo didesnė bazės dalis bus saugoma joje ir tuo greičiau veiks bazė), bent du 80 Gb greitaeigiai, serverio klasės, besisukantys 10-15k RPM, palaikantys SCSI arba SATA2 diskai sujunti į RAID 0 (arba 3, 4, 5, 6) konfigūracijos masyvą, tinklo plokštė 1 Gb/s.

- Žiniatinkliu paremtos konfigūravimo sąsajos serveris. Šiame serveryje veiks Microsoft Windows operacinė sistema, Jetty – Java paremtas žiniatinkliu serveris, bei žiniatinkliu paremta konfigūravimo sąsajos programinė įranga. Aparatinė įranga priklauso nuo konkretaus operatoriaus, tačiau bendru atveju turėtų užtikrinti didelį patikimumą, bei greitą prieigą prie interneto.

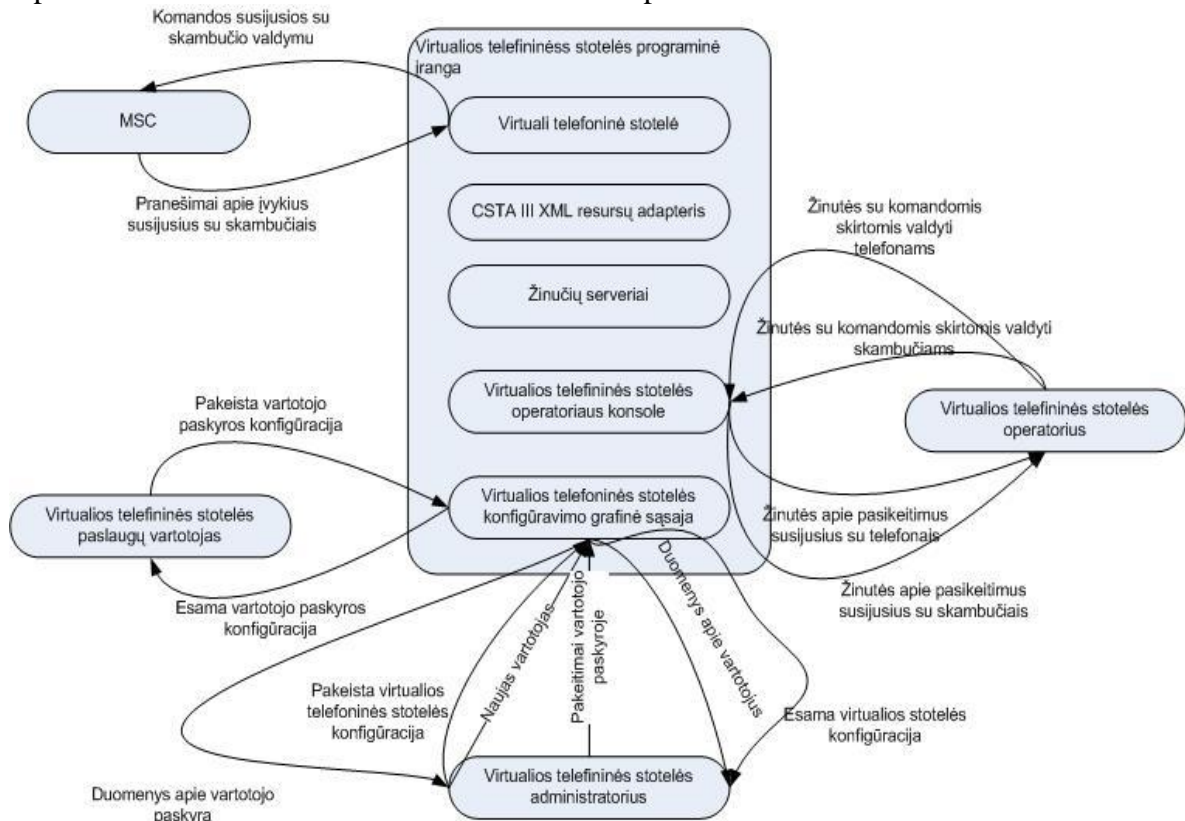
Preliminarūs reikalavimai: serveris turintis du Intel Pentium Xeon 2Ghz (ar atitinkamos spartos AMD analogą) keturių branduolių procesorius, 8 GB darbinės atminties, du 80 Gb diskus, tinklo plokštę 1 Gb/s su prieiga prie interneto.

- Virtualios stotelės operatoriaus PC – šiame kompiuteryje veiks Microsoft Windows operacinė sistema, bei skambučių centro programinė įranga. Minimalūs sistemos reikalavimai: 1 Ghz Intel ar AMD procesorius, 256 MB darbinės atminties, 10 Mb/s tinklo plokštė, prieiga prie interneto.

- Virtualios stotelės administratoriaus PC – šiame kompiuteryje gali veikti bet kokia operacinė sistema, svarbiausia, jog šis kompiuteris turėtų naršyklė palaikančia HTTPS protokolą (SSL šifravimas). 1 Ghz Intel ar AMD procesorius, 256 MB darbinės atminties, prieiga prie interneto.

2.5 Veiklos kontekstas

6 paveiksle **Error! Reference source not found.** pateikiamas sistemos veiklos kontekstas.



6 pav. Sistemos veiklos kontekstas

Paveiksle matome (žr. 6 pav.) sistemos veikimo metu vykstančius duomenų mainus. Pagal šią schemą galime susidaryti bendrą supratimą apie sistemos veikimą bei jos funkcinius reikalavimus.

3 lentelėje pateikiamas veiklos įvykių sąrašas, pagal kuriuos yra padalinama veikla.

3 Lentelė. Veiklos įvykių sąrašas

Eil. Nr.	Įvykio pavadinimas	Įeinantys išeinantys informacijos srautai
1.	Vartotojo skambučių aptarnauja virtuali telefoninė stotelė	Pranešimai apie įvykius susijusius su skambučiais (in) Komandos susijusios su skambučio valdymu (out)
2.	Virtualios telefoninės stotelės administratorius sukuria naują vartotoją	Duomenys apie vartotojus (out) Nauja vartotojo paskyra (in)
3.	Virtualios telefoninės stotelės administratorius pakoreguoja esamą vartotoją	Duomenys apie vartotojus (out) Duomenys apie vartotojo paskyrą (out) Pakeitimai vartotojo paskyroje (in)
4.	Virtualios telefoninės stotelės	Esama virtualios telefoninės stotelės konfigūracija

	administratorius pakeičia virtualios stotelės konfigūraciją	(out) Pakeista virtualios telefoninės stotelės konfigūracija (in)
5.	Virtualios telefoninės stotelės operatorius stebi skambučių, telefonų būseną	Pranešimai apie įvykius susijusius su skambučiais (in) Žinutės apie pasikeitimus susijusius su skambučiais (out) Žinutės apie pasikeitimus susijusius su telefonais (out)
6.	Virtualios telefoninės stotelės operatorius atlieka veiksmus susijusius su skambučiais (skambina, persiunčia, sustabdo, nutraukia, priparkuoja, išparkuoja ir kt.)	Pranešimai apie įvykius susijusius su skambučiais (in) Žinutės apie pasikeitimus susijusius su skambučiais (out) Žinutės su komandomis skirtomis valdyti skambučiams (in) Komandos susijusios su skambučio valdymu (out)
7.	Virtualios telefoninės stotelės operatorius atlieka veiksmus susijusius su telefono būsenomis (netrukdyti, peradresavimas ir kt.)	Žinutės apie pasikeitimus susijusius su telefonais (out) Žinutės su komandomis skirtomis valdyti telefonams (in)
8.	Virtualios telefoninės stotelės paslaugų vartotojas konfigūruoja savo paskyrą	Duomenys apie vartotojo paskyrą (out) Pakeitimai vartotojo paskyroje (in)

Pasitelkdami informacija iš šios lentelės, galime susidaryti bendrą sistemos struktūros vaizdą. Matome atskirus sistemos komponentus, bendrus duomenis ir ryšius tarp šių komponentų ir jų duomenų.

2.6 Panaudojimo atvejai

Nagrinėjant sistemos veiklos kontekstą bei surinktus sistemos reikalavimus, buvo sudaryta žemiau pateikta sistemos panaudojimo atvejų diagrama (žr. 7 pav. **Error! Reference source not found.**). Šiame darbe pateikiami tik aukščiausio lygio panaudos atvejai.

Pagal šią diagramą galime vartotojus suskirstyti pagal jų atliekamas funkcijas. Išnagrinėjus diagramą, galima išskirti pagrindines sistemos panaudojamumo grupes (žr. **Error! Reference source not found.**).

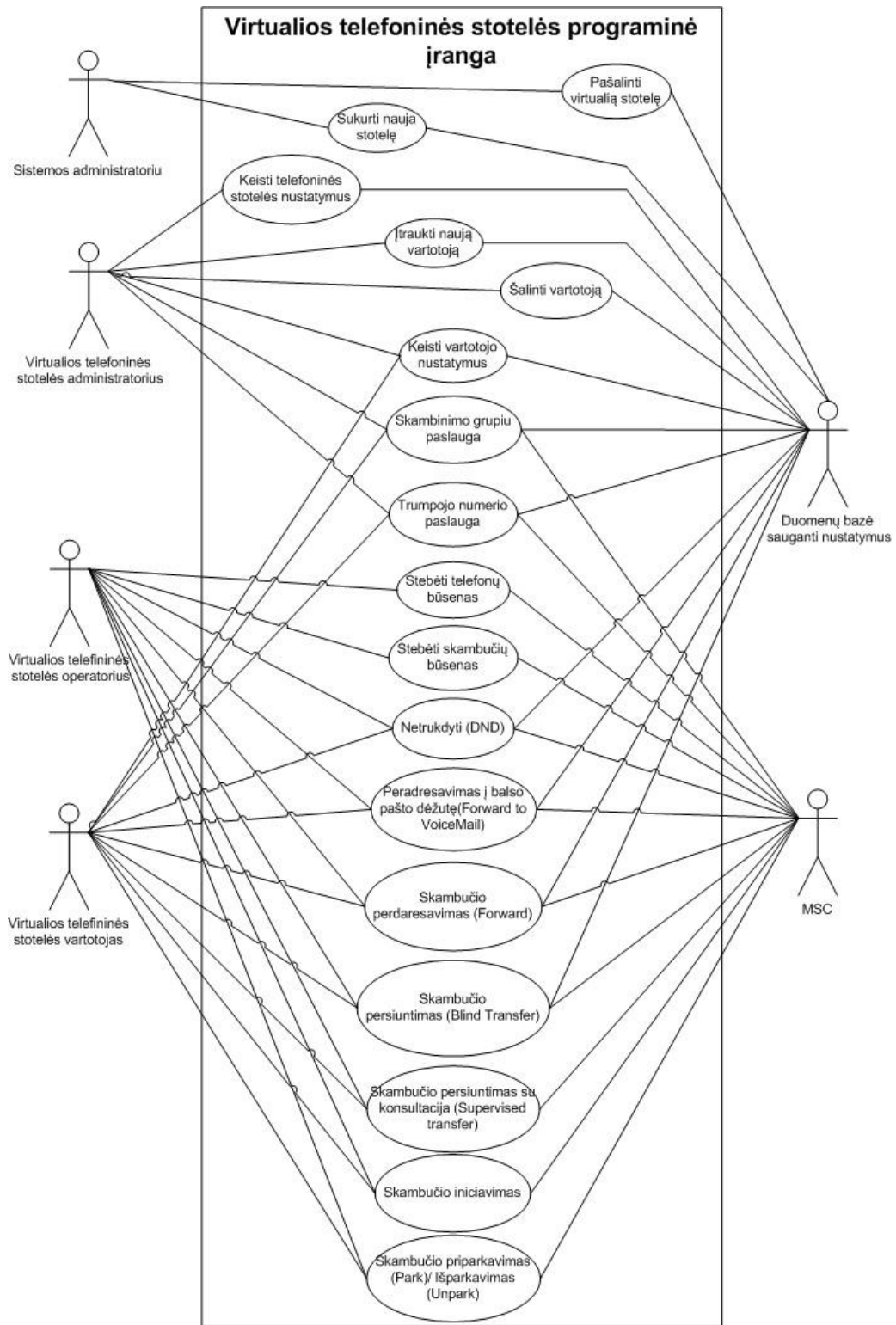
4 lentelė. Sistemos panaudojamumo grupės

Grupė	Grupę sudarantys panaudojimo atvejai
<i>Sistemos administravimas</i>	Virtualios stotelės sukūrimas Virtualios telefoninės stotelės pašalinimas

<p><i>Virtualios telefoninės stotelės administravimas</i></p>	<p>Vartotojo įtraukimas Vartotojo pašalinimas Vartotojo nustatymų keitimas Virtualios telefoninės stotelės nustatymų keitimas</p>
<p><i>Virtualios telefoninės stotelės ir operatoriaus konsolės teikiamų paslaugų teikiamų paslaugų naudojimas</i></p>	<p>Peradresavimas Peradresavimas į balso pašto dėžutę Netrukdyti Skambinimo grupių paslauga Trumpojo numerio paslauga Telefonų būsenos valdymas Skambučių būsenos valdymas Skambučio priparkavimas Skambučio išparkavimas Skambučio persiuntimas Skambučio persiuntimas su konsultacija</p>

Aktoriai:

1. Sistemos administratorius – tai vartotojas kuriantis naujas telefonines stoteles, priskiriantis joms abonentus.
2. Virtualios telefoninės stotelės administratorius – tai vartotojas kuris konfigūruoja virtualia telefoninę stotelę. Šis vartotojas gali konfigūruoti tam tikrų paslaugų teikimą savo telefoninės stotelės abonentams, įtraukti ar išmesti vartotojus.
3. Virtualios telefoninės stotelės operatorius – tai vartotojas, kuris stebi telefoninėje stotelėje vykstančius pokalbius, bei turi galimybę juos valdyti
4. Virtualios telefoninės stotelės vartotojas – tai vartotojas kuris naudojasi virtualios telefoninės stotelės teikiamomis paslaugomis, tokiomis kaip: skambučio peradresavimas, netrukdyti, sek mane/rask mane ir kitomis.
5. Duomenų bazė sauganti nustatymus – tai duomenų bazė kurioje yra saugomi visos sistemos nustatymai.
6. MSC – tai mobiliojo ryšio operatoriaus telefoninė stotis, kuri atlieka realius veiksmus su skambučiais, ir siunčia pranešimus naujus įvykius į virtualią telefoninę stotelę.



2.7 Funkciniai reikalavimai

Sistemos reikalavimai buvo renkami naudojantis *Volere* šablonu. *Pateikiami tokie svarbiausi sistemos funkciniai reikalavimai:*

1. Sistema turi leisti konfigūruoti virtualią stotelę individualiai – kiekviena įmonė turi turėti galimybę virtualia telefoninę stotelę susikonfigūruoti pagal savo poreikius.
2. Sistema turi suteikti galimybę kiekvienam vartotojui keisti savo paskyros nustatymus.
3. Sistema turi leisti atlikti skirtingus veiksmus konfigūruojant virtualia telefoninę stotelę, kurie priklauso nuo vartotojo tipo: sistemos administratorius, virtualios telefoninės stotelės administratorius, virtualios telefoninės stotelės vartotojas.
4. Sistema turi leisti priskirti naują telefoną virtualiai telefoninei stotelei.
5. Sistema turi leisti konfigūruoti balsinius pranešimus.
6. Sistema turi palaikyti skambinimo grupių paslaugą.
7. Sistema turi palaikyti sek mane/rask mane paslaugą (Find me/follow me).
8. Sistema turi leisti konfigūruoti trumpuosius numerius prie stotelės prijungtiems telefonams.
9. Sistema turi leisti stebėti prie virtualios telefoninės stotelės prijungtų telefonų būsenas.
10. Sistema turi leisti stebėti prie virtualios telefoninės stotelės prijungtų telefonų skambučius.
11. Sistema turi leisti telefonui nustatyti Netrukdyti (DND) režimą.
12. Sistema turi leisti peradresuoti įeinančius skambučius į kitą numerį (Forward).
13. Sistema turi leisti peradresuoti skambučius į vartotojo balso pašto dėžutę (Forward to VoiceMail).
14. Sistema turi leisti persiųsti vykstantį skambutį (Blind transfer).
15. Sistema turi leisti persiųsti vykstantį skambutį su papildoma konsultacija (Supervised transfer).
16. Sistema turi leisti priparkuoti skambutį (Park).
17. Sistema turi leisti išparkuoti priparkuotą skambutį (Unpark).

18. Sistema turi vesti atliktų veiksmų apskaitą, pagal kurią operatorius galėtų apskaičiuoti sąskaitą už suteiktas paslaugas.

2.8 Nefunkciniai reikalavimai

Šiame skyriuje glaustai pateikiamas nefunkcinių reikalavimų sąrašas sugrupuotas į kelias pagrindines kategorijas.

a) Reikalavimai sistemos išvaizdai

1. Sąsaja turi būti intuityvi.
2. Sąsajos visi veiksmai turėtų būti pasiekiami iki į pelės mygtukų paspaudimų.
3. Sąsaja turi būti lanksti, vartotojas turi turėti galimybę pats susikonfigūruoti matomus mygtukus – konfigūruojamos mygtukų juostos (angl. toolbars).
4. Sąsaja turi būti lanksti, vartotojas turi turėti galimybę pats susikonfigūruoti matomus langus, keisti jų vietą bei dydžius programoje Novatoriška ir šiuolaikinė išvaizda.
5. Vartotojo sąsajos spalvos turėtų derėti tarpusavyje, neturėtų būti varginančios akis.
6. Vartotojo sąsajos mygtukai ir meniu punktai turėtų turėti piktogramas, kurios paspartina navigaciją.
7. Vartotojo sąsaja turi turėti greitosios paleisties klavišus, kadangi jie gerokai pagreitina darbą įgudusiems vartotojams.
8. Kur įmanoma naudoti dialogus neblokuojančios pačios programos darbo.
9. Reikia naudoti garsinius pranešimus ir atoveikį į tam tikrus vartotojo veiksmus ar pranešimus apie klaidas. Garsiniai išpėjimai ar atoveikis neturėtų būti nemalonus ausiai ar šaižus.

b) Reikalavimai panaudojamumui

10. Vartotojo sąsaja turi būti daugiakalbė, kadangi planuojama, jog sistema bus parduoda ne tik Lietuvos operatoriams, be to netgi vienoje šalyje sistema gali naudotis kelių tautybių žmonės.
11. Turėtų būti sumažinta galimybė vartotojams padaryti klaidą, o net ir padarius klaidą, jos pasekmės neturėtų būti kritinės.

c) Reikalavimai vykdymo charakteristikoms

12. Sistema turi sugebėti apdoroti iki 100 CPS .
13. Grafinė vartotojo sąsaja turėtų sugebėti sklandžiai dirbti esant 5 CPS.

14. Grafinė vartotojo sąsaja neturėtų vėlinti vaizdo daugiau negu vieną sekundę
Reikalavimai veikimo sąlygoms
15. Sistema turi veikti visose populiariausiose interneto naršyklėse.

d) Reikalavimai veikimo sąlygoms

16. Nepriklausomai nuo klaidų atsirandančių dėl per didelės tinklo apkrovos, dėl klaidų duomenų bazės konfigūracijoje, sistema turi aptarnauti skambučius telefoninės pastotės abonentams jeigu tai įmanoma.

e) Reikalavimai sistemos priežiūrai

17. Grafinės vartotojo sąsajos turi būti lengvai administruojamos, kadangi jomis naudosis nepatyrę vartotojai.
18. Grafinė vartotojo sąsaja telefoninės stotelės konfigūravimui turi būti lengvai administruojama, šių stotelių administratorių apmokymo laikas turi būti kaip įmanoma trumpesnis.
19. Grafinė vartotojo sąsaja turėtų veikti Windows 2000, XP, Vista bei 7 sistemose.
20. Grafinė vartotojo sąsaja turi turėti galimybę lengvai įdiegti naują kalbą, kalbos failai turėtų būti saugomi atskiruose failuose, naujos kalbos įdiegimas turėtų vykti nekeičiant pačios programos kodo.
21. Grafinė vartotojo sąsaja ir virtualios telefoninės stotelės programinė įranga turi būti suprojektuota taip, jog būtų sąlyginai lengva įterpti naujas paslaugas, nesugriaunant visos sąsajos architektūros.

f) Reikalavimai saugumui

22. Turi būti šifruojami patys svarbiausi duomenys susiję su vartotojo paskyros informacija .
23. Vartotojo slaptažodžiai duomenų bazėje turi būti saugomi šifruotame pavidale.

g) Kultūriniai–politiniai reikalavimai

24. Sistemoje naudojama korektiška kalba, nežargoniniai terminai.
25. Sistema neturi ižveisti religinių ar etninių grupių.
26. Anglų kalboje negalima vartoti termino „Blind Transfer“, dėl jo ižeidžiančio pobūdžio.

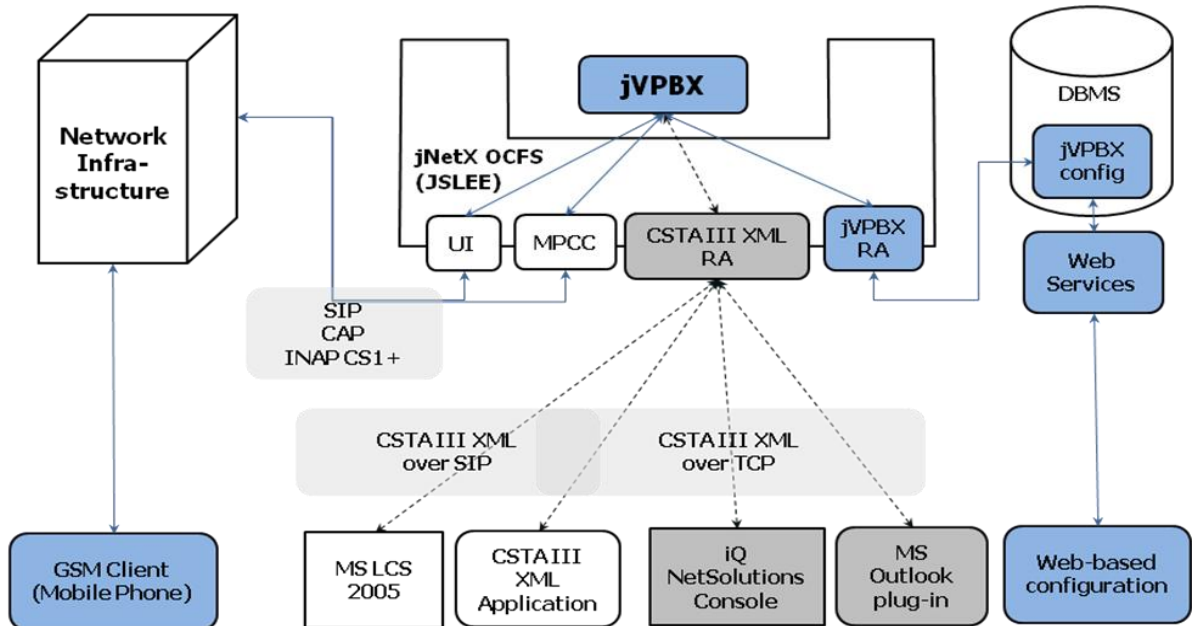
h) Teisiniai reikalavimai

27. Sistema turi veikti pagal šalies, kurioje ji įdiegta, įstatymus.

2.9 Architektūros specifikacija

2.9.1 Bendras architektūros vaizdas

Pateiktame paveiksle (žr. 8 pav.) pateikiama principinė programinių komponentų išdėstymo schema.



8 pav. Bendra sistemos architektūra – komponentų išdėstymas

Šios schemas dėka galime susidaryti vaizdą apie aukšto lygio sistemos architektūrą:

- jVPBX – tai virtuali telefoninė stotelė veikianti jNetX OCFS (JAIN SLEE specifikacija paremtas) aplikacijų serveryje.
- CSTA III XML resursų adapteris – šis resursų adapteris taip pat veikia jNetX OCFS serveryje. Jis užtikrina prisijungimą prie jVPBX telefoninės stotelės per CSTA III XML protokolą – suteikia galimybę gauti pranešimus apie įvykius, bei suteikia galimybę iškviešti įvairias komandas. Realizuota JAVA programavimo kalba.
- jVPBX RA – šis resursų adapteris, veikiantis jNetX OCFS serveryje, suteikia priėjimą prie įvairių nustatymų saugomų duomenų bazėje, bei užtikrina tarpinį lygį tarp duomenų bazės ir jVPBX virtualios stotelės, suteikia galimybę išlaikyti vieną prisijungimą prie duomenų bazės. Realizuota JAVA programavimo kalba.
- jVPBX config – tai duomenų bazė sauganti virtualios telefoninės stotelės nustatymus. Realizuota panaudojant Oracle Times-Ten in Memory Database RDBS duomenų bazių valdymo sistema.

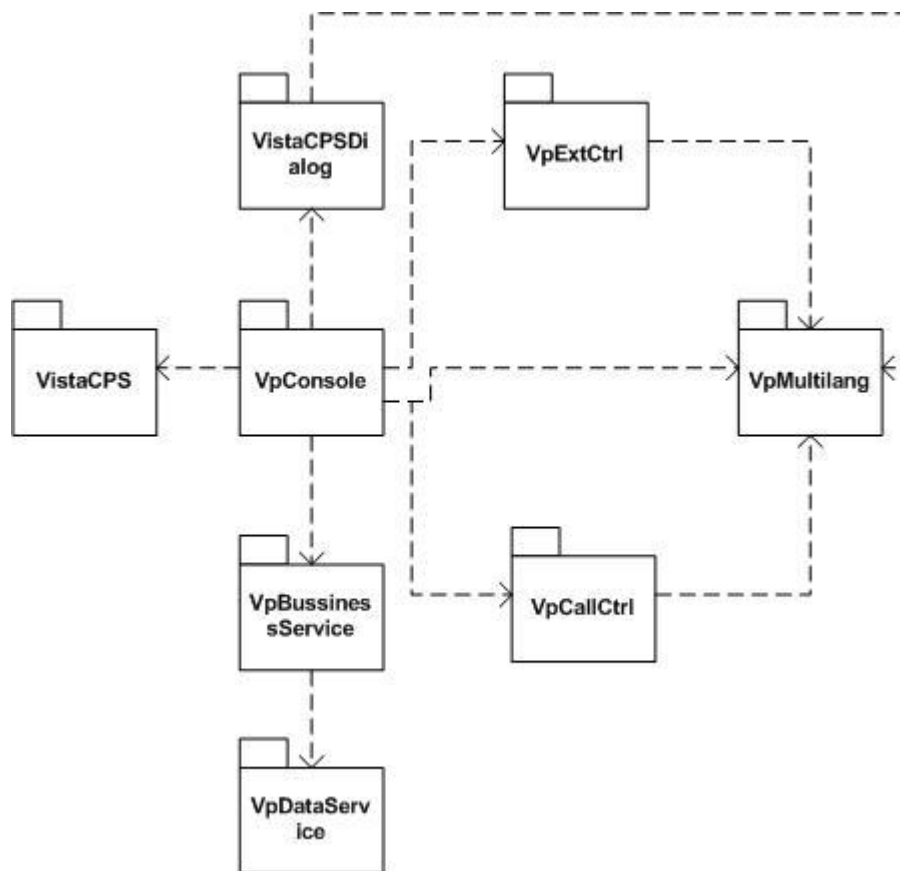
- Web-based configuration – tai žiniatinkliu paremta vartotojo sąsaja skirta, konfigūruoti virtualią telefoninę stotelę. Realizuota JAVA programavimo kalba, veikia Jetty serveryje.
- iQ NetSolutions Konsole – Skambučių valdymo centro programinę įrangą – operatoriaus konsolė. Ši įranga skirta stebėti ir valdyti įeinančius skambučius iš vieno ar keleto telefonų. Programinė įranga yra paimta ir kito proginės įrangos paketo – iQ NetSolutions VistaPoint, perdaryta ir pritaikyta veikti su CSTA III XML protokolu ir dideliu skambučių srautu. Programinę įrangą yra realizuota C++ kalba.
- MS Outlook plug-in – Microsoft Outlook programinė įrangos įskiepis, skirtas stebėti ir valdyti vieno telefono skambučius. Paremtas komponentais sudarančiais iQ NetSolutions programinę įrangą. Šiame darbe į šį komponentą plačiau nesigilinsime.
- MS LCS 2005 ir CSTA III XML Application – trečių šalių skambučių valdymo centrų programinę įrangą suderinama su CSTA III XML standartu.

Tarp CSTA III XML programos klientės ir CSTA III XML resursų adapterio įsiterpia žinučių serveriai. Detaliau galima pažiūrėti aparatinės įrangos išdėstymo schemeje (žr. 6 pav.).

Sistemos architektūra. Dėl sistemos dydžio trumpai apžvelgsiu tik komponentus prie kurių buvo dirbta: virtualia telefoninę stotelę (jVPBX) ir operatoriaus konsolę. Operatoriaus konsolė (žr. 9 pav.) yra sudaryta iš kelių pagrindinių modulių:

- VpConsole – tai pagrindinės programos komponentas, jame vaizduojami visi menui, į jį susideda VpExtCtrl ir VpCallCtrl komponentų langai, per jį siunčiamos komandos iš ir į VpCallCtrl ir VpExtCtrl komponentus.
- VistaCPS – šis komponentas yra atsakingas už skambučius apjojančių scenarijų (angl.: script) vykdymą.
- VistaCPSDialog – šis komponentas, tai langas kuriame galime redaguoti, trinti senus, kurti naujus skambučius apdorojančius scenarijus.
- VpExtCtrl – tai COM komponentas, vaizduojamas kaip langas ir yra skirtas vaizduoti ir valdyti viską kas susiję su telefonais.
- VpCallCtrl – tai COM komponentas, vaizduojamas kaip langas ir yra skirtas vaizduoti ir valdyti viską kas susiję su skambučiais.
- VpMultilang – šis komponentas atsakingas už daugiakalbystę.
- VpBussinessService – šis komponentas užtikrina visą biznio logiką susijusią su skambučių valdymu, suformuoja atitinkamus pranešimus tinklui ir išskviečia atitinkamus VpConsole metodus atėjus įvykiams iš tinklo.

- VpDataService – užtikrina sąsają su tinklu, formuoja ir siunčia atitinkamus pranešimus, o naujus gautus stato į VpBussinessService eilę.



9 pav. Operatoriaus konsolę sudarantys paketai

jVPBX virtualią stotelę sudarantys komponentai matomi 10 paveiksle. Šiuos komponentus nėra labai teisinga vadinti paketais. MOApplication ir MTApplication yra SBB (serviso) tipo komponentai, o visi kiti yra ABB (servisą sudarančios aplikacijos) tipo komponentai. Terminai plačiau paaiškinti terminų ir santrumpų žodyne. Klasių diagrama nei vienam iš šių komponentų nebus pateikiama, kadangi programos kodas yra generuojamas iš nubraižytos schemos, kurioje yra įterpiami standartiniai arba mūsų sukurti ABB komponentai ir komponentai, kuriuose užrašomas trūkstamas Java kodas.

Galima pateikti trumpą virtualią telefoninę stotelę sudarančių komponentų aprašymą:

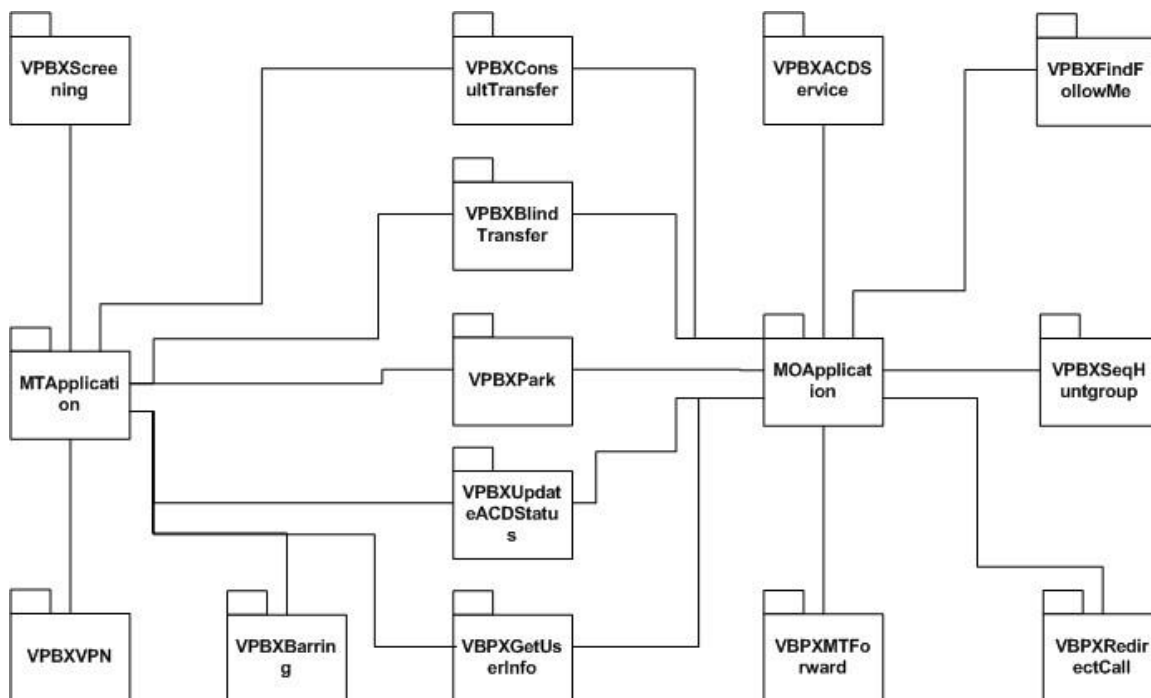
SBB tipo komponentai:

- MOApplication – tai programa, kuri sužadinama sukuriant naują skambutį ir aptarnauja ta telefoną, kuris sukūrė skambutį. Šis komponentas aptarnauja ir trumpuosius numerius.

- MTApplication – tai programa, kuri sužadinama įeinančiam skambučiui ir aptarnaują tą telefoną, kuriam skambinama.

ABB tipo komponentai:

- VPBXScreening – šis komponentas realizuoja įeinančių skambučių blokavimą.
- VPBXVPN – šis komponentas realizuoja su trumpaisiais numeriais susijusią logiką.
- VPBXBarring – šis komponentas realizuoja išeinančių skambučių blokavimą.
- VPBXGetUserInfo – šis komponentas skirtas informacijai gauti ir siųsti į duomenų bazę.
- VPBXUpdateACDStatus – šis komponentas skirtas eilės statuso atnaujinimui.
- VPBXPark – šis komponentas atlieka skambučio priparkavimo ir išparkavimo logiką.
- VPBXBlindTransfer – šis komponentas atlieka skambučio persiuntimo logiką.
- VPBXConsultTransfer – šis komponentas atlieka skambučio persiuntimo su konsultacija logiką.
- VPBXACDService – šis komponentas realizuoja skambinimo eilių logiką.
- VPBXForward – šis komponentas realizuoja skambučio nukreipimo į kitą numerį ar balso pašto dėžutę logiką.
- VPBXFindFollowMe – šis komponentas realizuoja sek mane/rask mane paslaugos logiką.
- VPBXSeqHuntgroup – šis komponentas realizuoja skambinimo grupių paslaugos logiką.
- VPBXRedirectCall – šis komponentas realizuoja skambučio nukreipimo logiką.



10 pav. Virtualią telefoninę stotelę sudarantys komponentai

2.9.2 Sistemos dinaminis vaizdas

Šiame poskyryje pateikiamos esminės sistemos būsenų, veiklos ir sąveikos UML diagramos. Dėl diagramų dydžio ir sąlyginio panašumo, pateikiame diagramas tik vienam panaudos atvejui – skambučio persiuntimas su konsultacija.

„Skambučio persiuntimas su konsultacija“ yra vienas sudėtingiausių panaudos atvejų – esamas skambutis yra persiunčiamas sukūrus konsultacinį skambutį. Yra laikoma, jog visi trys telefonai priklauso virtualiai telefoninei stotelei. Diagramose naudojami terminai plačiau aprašyti dokumento gale esančiame terminų ir santrumpų žodyne.

Paaiškinimai:

- Originating app, terminating app ir CSTA III XML RA – virtualios telefoninės stotelės komponentai.
- VPDataService, VPDataService, MainFrame, VPExtCtrl ir VPCallCtrl – grafinės vartotojo sąsajos komponentai.
- CSTA3Client, CSTA3Server ir CSTA3Devices – žinučių serverio komponentai.

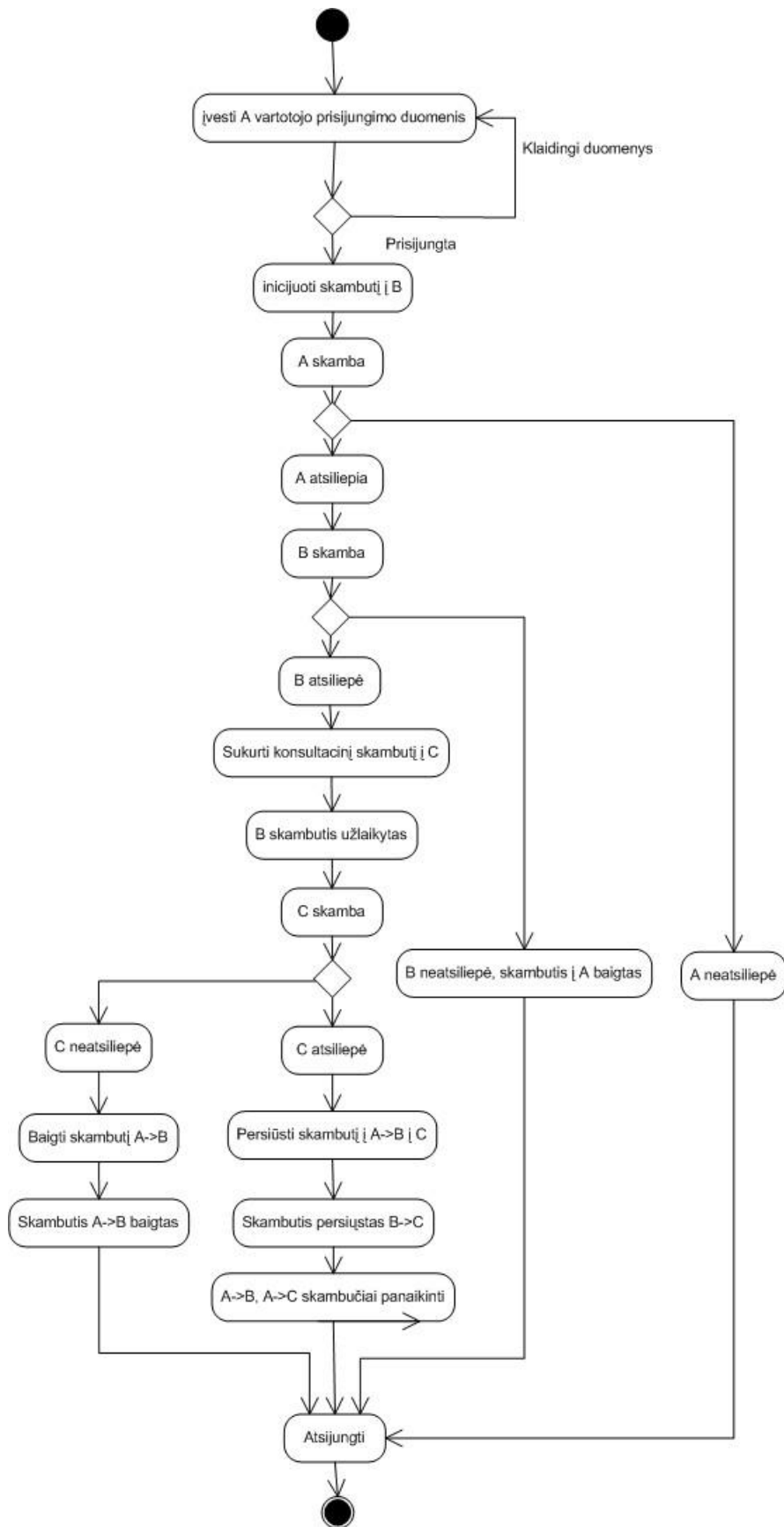
Būsenų diagramos: Žemiau pateiktos dvi būsenų diagramos (žr. 11 pav. ir 12 pav), kuriuose matome kokias būsenas pereina virtuali telefoninės stotelė ir operatoriaus konsolė vykstant skambučio persiuntimui su konsultacija – pradinio ir konsultacinio skambučio užmezgimas, bei galiausiai skambučio persiuntimas.

Pirmoje diagramoje „kita pusė“ yra telefonas su kuriuos užmezgamas pradinis skambutis, o „trečia pusė“ – telefonas su kuriuo užmezgamas konsultacinis skambutis. Antroje diagramoje minima „A pusė“ yra operatoriaus telefonas, „B pusė“ – telefonas su kuriuo užmezgamas pradinis skambutis, o „C pusė“ – telefonas su kuriuo užmezgamas konsultacinis skambutis.

Kaip matome iš diagramų, kuriant skambutį iš operatoriaus konsolės, visų pirma yra paskambinama į operatoriaus telefoną ir tik šiam atsiliepus, yra kuriamas konsultacinis skambutis. Tokia veiksmų seka yra įtakota mobiliojo ryšio telefonų specifikos – nėra specialios komandos, kurios pagalba galime sujungti skambutį, be vartotojo įsikišimo.

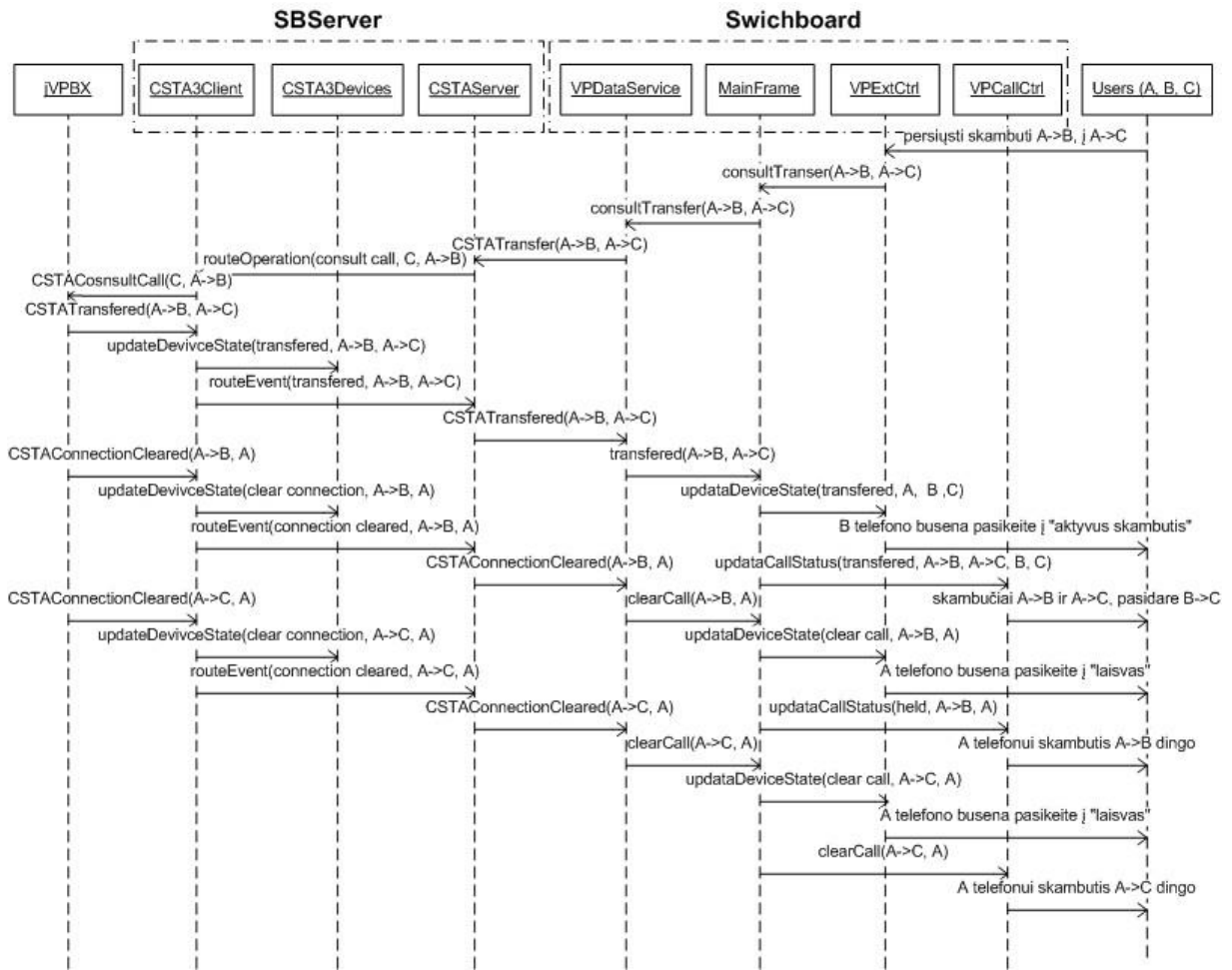


11 pav. Skambučio persiuntimo su konsultacija būsenų diagrama – virtualios telefoninės stotelės vaizdas



12 pav. Skambučio persiuntimas su konsultacija – skambučio iniciatoriaus operatoriaus konsolės

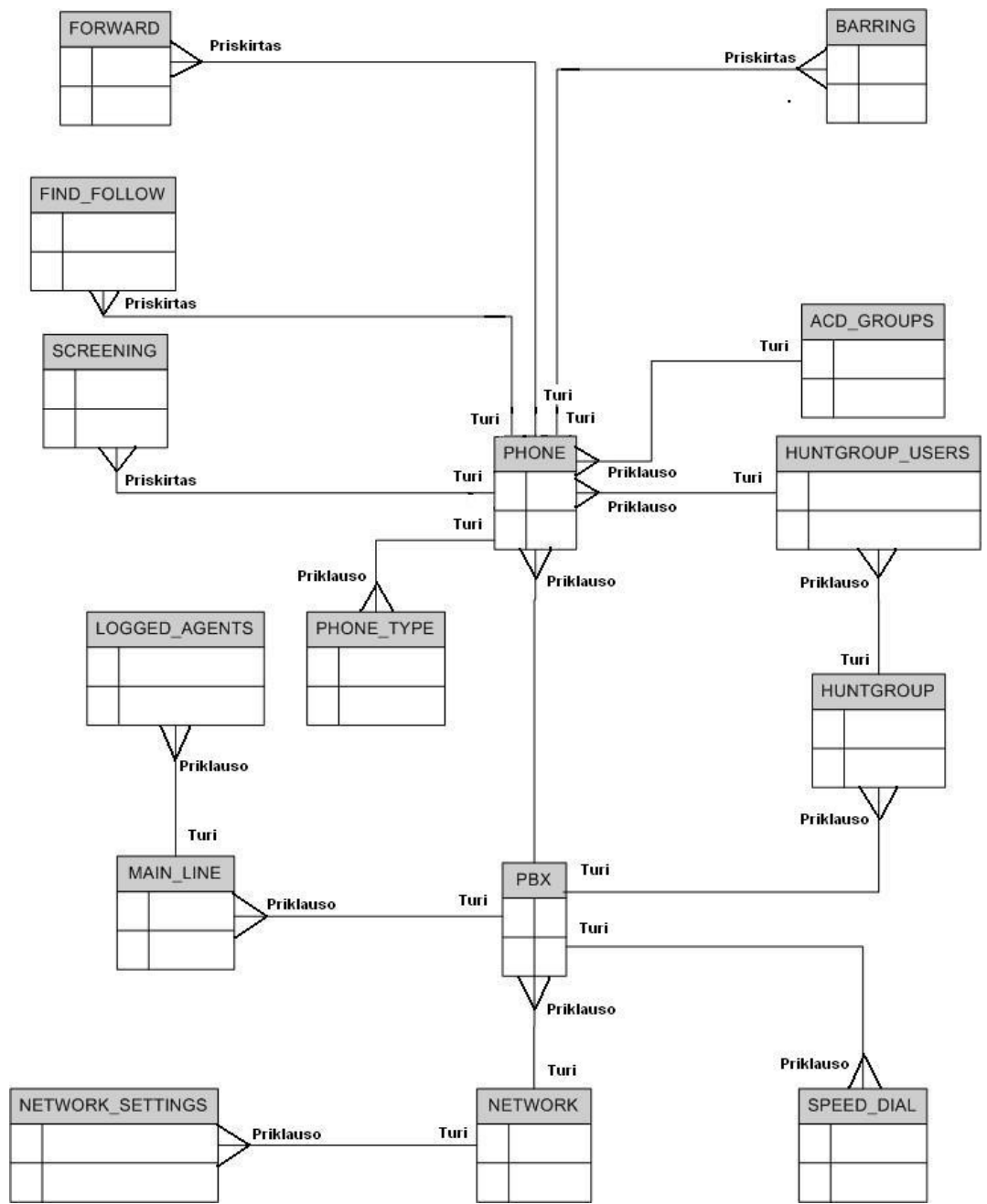
Sąveikos diagramos. Žemiau esančioje diagramoje (žr. 13pav.) matome kas vyksta virtualioje telefoninėje stotelėje. Sekančiose diagramose (žr.14 pav. ir **Error! Reference source not found.**) yra pavaizduota kas vyksta grafinėje vartotojo sąsajoje. Pirmoje diagramoje vaizduojamas konsultacinio skambučio sukūrimas ir užlaikymas, antroje skambučio persiuntimas. Pateiktose diagramose matosi kokie komponentai bendrauja tarpusavyje ir koks yra veiksmų eiliškumas.



15 pav. Skambučio persiuntimo su konsultacija sekų diagrama– operatoriaus konsolės vaizdas (tęsinys)

2.9.3 Duomenų vaizdas

16 paveiksle pateikiamas virtualios telefoninės stotelės sistemos duomenų bazės modelio vaizdas:



16 pav. Duomenų bazės modelis

Duomenų bazės modelyje esančių esybių aprašymai pateikti 5lentelėje.

5 lentelė. Duomenų bazės modelio esybės

Lentelė	Aprašymas
FORWARD	Saugo informaciją susijusią su skambučio nukeipimais
BARRING	Saugo informaciją susijusią su išeinančių skambučių blokavimu
FIND_FOLLOW	Saugo informaciją susijusią su sek mane/rask mane paslauga
SCREENING	Saugo informaciją apie įeinančių skambučių blokavimą

Lentelė	Aprašymas
ACD_GROUPS	Saugo informaciją apie skambinimo grupes
HUNTGROU	Saugo informaciją apie skambinimo eiles
HUNTGROU_USERS	Saugo informaciją apie skambinimo eilės vartotojus
PHONE	Saugo informaciją apie virtualiai telefoninei stotelei priklausančius telefonus
PHONE_TYPE	Saugo informaciją apie telefonų tipus
LOGGED_AGENTS	Saugo informaciją apie saugo informacija apie prie sistemos prisijungusius operatorius
MAIN_LINE	Saugo informaciją apie virtualiai telefoninei stotelei priskirtą liniją
PBX	Saugo informaciją apie sukonfigūruotas virtualias telefonines stoteles
NETWORK	Saugo informaciją apie tinklą
SPEED_DIAL	Saugo informaciją apie trumpuosius numerius
NETWORK_SETTINGS	Saugo informaciją apie tinklo nustatymus

2.10 Praktinis darbo rezultatas

Užbaigus darbą gautas programinės įrangos produktas. Šiuo metu vyksta derybos su keliais potencialiai užsakovais dėl viso produkto pardavimo. Dalis produkto jau parduoda atskirai. Produktas toliau sėkmingai tobulinamas, šiuo metu planuojamas atlikti produkto perkėlimą į kitą platformą.

Taip pat, darbo vykdymo metu buvo sukurta programinės įrangos techninė dokumentacija. Visą dokumentų sąrašą sudaro:

- projekto paraiškos dokumentas;
- projektavimo metodologijos ir technologijos analizė;
- detalus projekto planas;
- reikalavimų specifikacija;
- architektūros specifikacija;
- detalios architektūros specifikacija;
- programinės įrangos licencija;
- testavimo planas;

- kokybės analizės dokumentas;
- vartotojo dokumentacija ir kiti pagalbiniai dokumentai.

Dalis šių dokumentų medžiagos panaudota ir šiame darbe. Taip pat darbų pradžioje buvo sukurta projekto informacinė sistema, kurioje yra saugoma visa su projektu susijusi dokumentacija.

3 EKSPERIMENTINĖ IR TIRIAMOJI DALIS

Šiame skyriuje bus palyginamos ir analizuojamos dvi telekomunikacijų platformos – jNetX ir Mobicents. Tyrimo tikslas yra nustatyti ar Mobicents platforma yra realus kandidatas pakeisti jNetX OCFS.

Kol buvo kuriama virtuali telefoninė stotelė, jNetX OCFS platforma taip pat vystėsi, o jai besivystant žymiai pakilo ir jos kaina. Kadangi sukurta virtuali telefoninė stotelė veikia būtent jNetX OCFS platformoje, todėl kyla keblumų ją parduodant – dėl smarkiai išaugusios jos kainos, dauguma potencialių klientų tiesiog atsisako pirkti virtualią telefoninę stotelę, todėl kad turi kartu įsigyti ir jNetX OCFS platformą, kuri yra brangesnė už pačią virtualią telefoninę stotelę. Dėl to yra ieškoma alternatyvų šiai platformai. Kaip vienas iš galimų variantų svarstoma atviro kodo Mobicents JAIN SLEE platforma. Šios sistemos bus palyginamos keliais svarbiausiais aspektais. Taip pat bus suformuluotos tyrimo išvados.

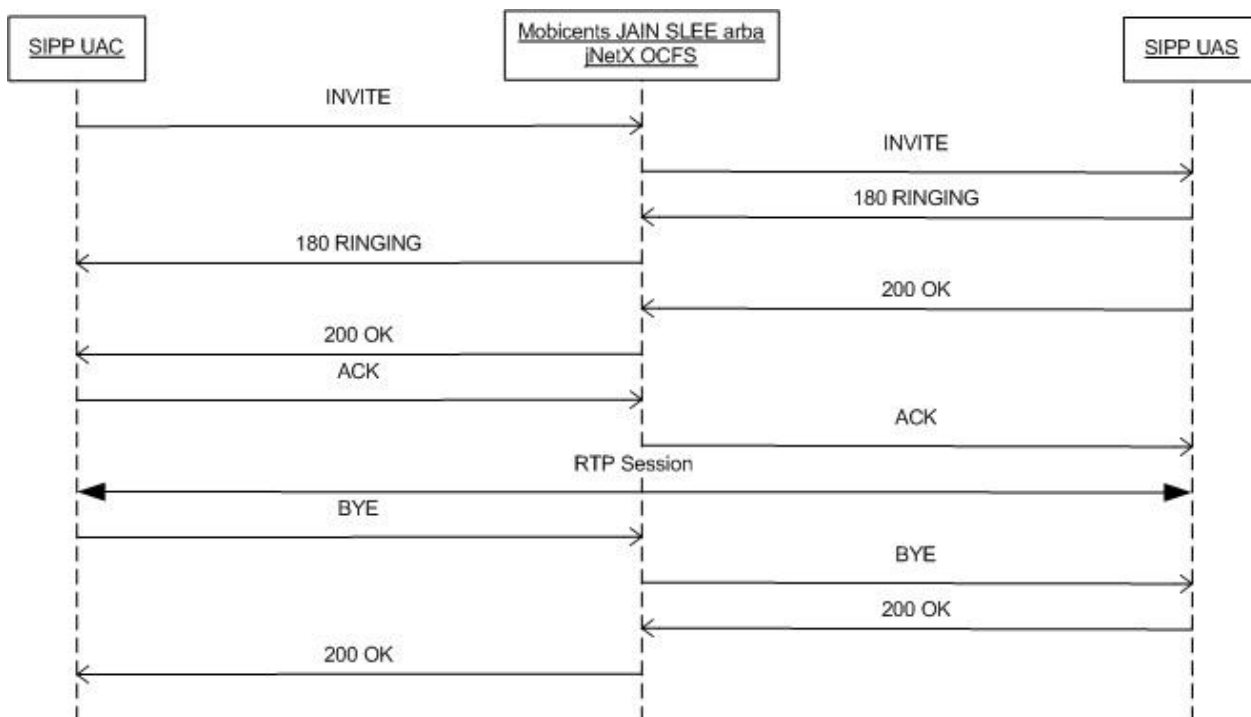
3.1 Platformų darbo patikimumas

Ko gero viena iš svarbiausių savybių telekomunikacijų sistemai yra jos darbo stabilumas ir pasiekiamumas, nes laikas kai sistema neveikia atneša tiesioginį nuostolį operatoriui bei sukelia nepatogumus ir galimus nuostolius abonentams.

Patikimumui nustatyti buvo atliktas apkrovos testas. Testavimui buvo sukurta po nesudėtingą komponentą kiekvienai platformai. Komponento tikslas yra sukомуtuoti skambutį. Testuojama buvo VOIP tipo skambučiai paremti SIP protokolu. Testavimui buvo panaudota tam skirta programinė įranga SIPP. Šios įrangos dėka galima nesunkiai sukurti mums reikiamą skambučio scenarijų ir imituoti tiek skambinantįjį tiek atsiliepiantįjį galą. Testo metu bus imituojamas įprastinis skambutis su tokia įvykių seka:

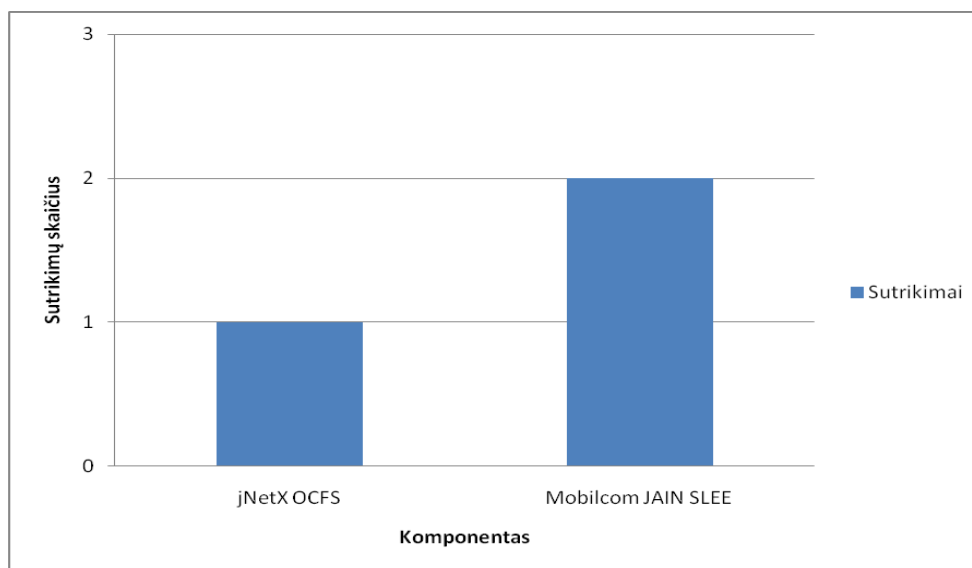
- 1) A pusė kviečia B
- 2) B pusė skamba
- 3) B pusė atsiliepia
- 4) Skambutis tarp A ir B užmegztas

Testavimo metu naudoto scenarijaus sekų diagrama ir programinės įrangos išdėstymo schema pateikta paveiksluke žemiau (žr. 17 pav.) Kaip matome tiriamoji platformą atlieka nesudėtingo B2BUA vaidmenį ir tiesiog sukомуtuoja skambutį. SIPP UAC atlieka skambinančiojo telefono vaidmenį, SIPP UAS atlieka atsiliepiančiojo telefono vaidmenį.



17 Pav. Testavimui naudotas skambučio scenarijus

Testavimui buvo nustatytas 500 skambučių per sekundę dažnumas – 500 CPS. Jis yra didesnis nei bet kuri iš šių platformų pajėgia apdoroti testuojamoje aparatinėje įrangoje. Testavimui buvo paskirta po 1 savaitę laiko, platformos buvo testuojamos paeiliui. Grafike (žr. 18 pav.) matome abiejų platformų darbo sutrikimų pasiskirstymą.



18 pav. Testuojamų platformų darbo sutrikimai

Kadangi buvo testuojama Windows sistemoje, tai iš jos log failų buvo nustatyta ir lūžių priežastis – pasibaigė darbinė atmintis, o kadangi buvo išjungtas SWAP failo palaikymas

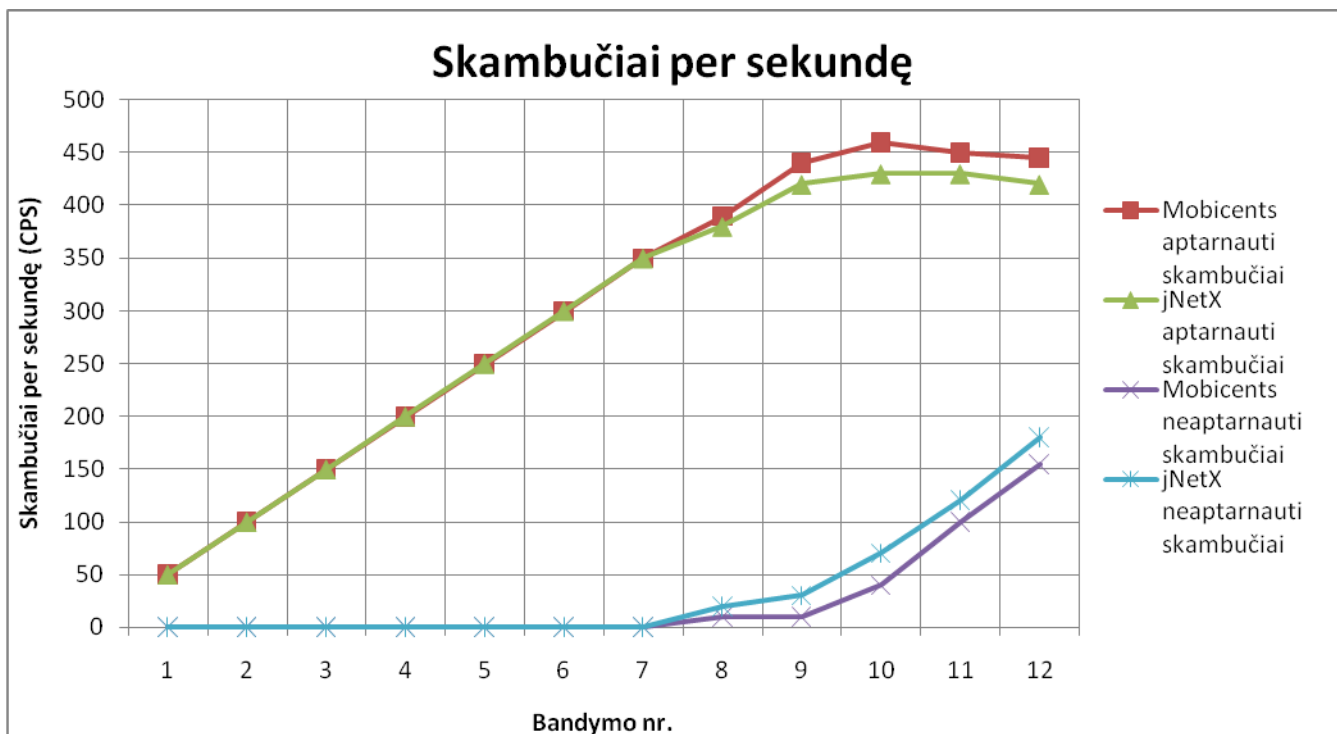
operacinė sistema nusprendė nutraukti daugiausia atminties naudojančių programų darbą, kurios šiuo atveju ir buvo testuojamos platformos. Taigi teisingai sukonfigūravus operacinę sistemą, šių lūžių turėtų pavykti išvengti. Tačiau atliktas testas yra sintetinis ir nevisai atitinka realaus pasaulio scenarijų, kurie yra gerokai įvairesnis.

Apibendrinant galima teigti, jog tikrasis sistemų stabilumas paaiškės tik jas naudojant realiame gyvenime. Testavimo metu platformos dirbo pakankamai stabiliai net prie maksimalių apkrovimų. Įvykusius sutrikimus labiausiai nulėmė operacinės sistemos konfigūracijos, o nei testuojamos platformos.

3.2 Platformų darbo našumas

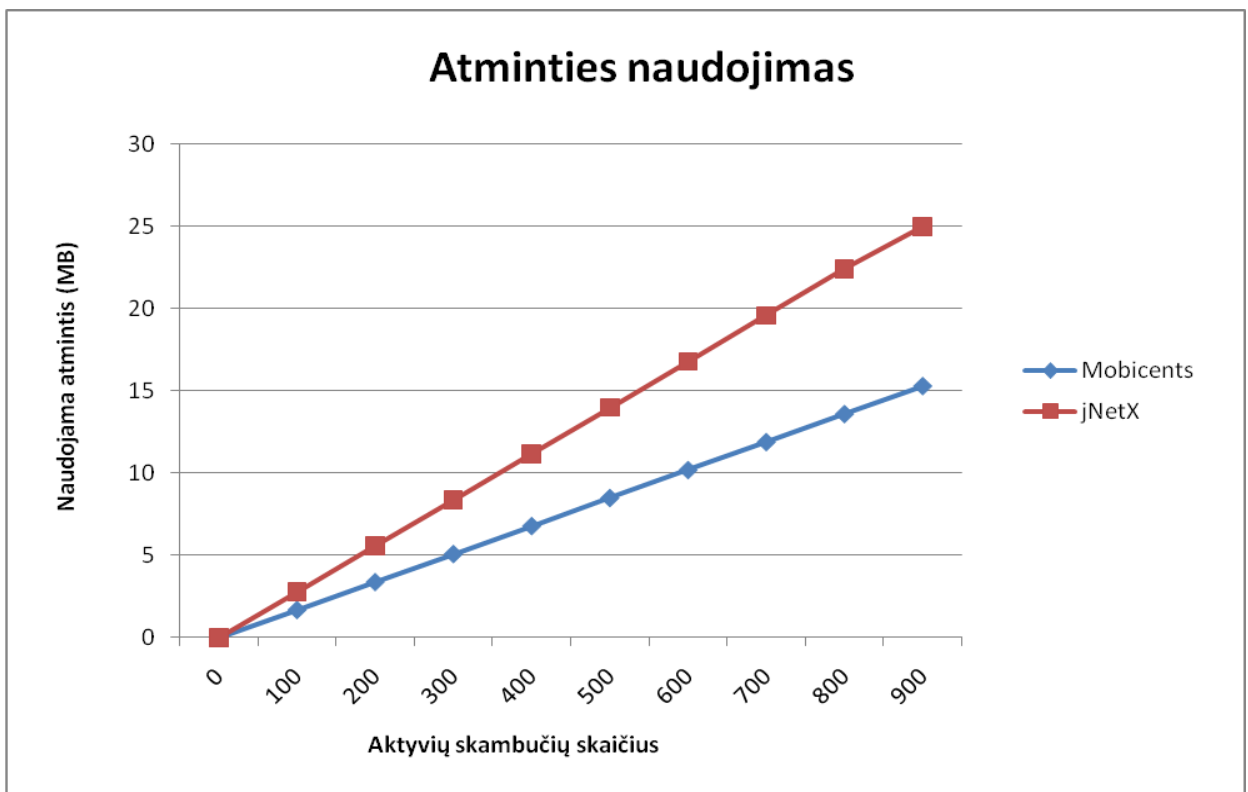
Tiriant platformų darbo našumą, buvo tikrinama koks maksimalus apdorojamų skambučių kiekis (CPS) abiejose platformose, kiek atminties yra naudojama priklausomai nuo aktyvių skambučių skaičiaus, kaip kinta skambučio sukmutavimo laikas priklausomai nuo aktyvių skambučių skaičiaus. Buvo naudojamosi tais pačiais metodais ir priemonėmis, kaip ir testuojant platformų darbo patikimumą.

Grafike (žr. 19 pav.) matome, kaip sistemos elgiasi prie skirtingo skaičiaus aktyvių skambučių (CPS). Testuojant buvo nuosekliai didinamas skambučių skaičius per sekundę ir žiūrima, kada sistema pradeda nebeaptarnauti skambučių. Galima pastebėti, jog jNetX platforma savo greitaveika šiek tiek nusileidžia Mobicents platformai, tačiau žymaus skirtumo nėra, galbūt tai susiję su tuo, jog platforma yra senesnė ir pagal apimtį gerokai didesnė. Abi platformos be didesnių problemų sugeba apdoroti 300 skambučių per sekundę, vėliau pasirodo jau ir neaptarnautų skambučių. Savo piką abi platformos pasiekia ties 450 skambučių per sekundę, tačiau tuomet jau yra ir didelė dalis nebeaptarnautų skambučių. Reikia pabrėžti, jog buvo numatyta 100 skambučių per sekundę virtualios telefoninės stotelės programinei įrangai. Taip pat čia matome sistemos greitaveiką kai skambutis yra tiesiog sukmutuojamas, o realiame gyvenime dar reikia atlikti ir papildomų veiksmų: užsiklausti duomenų iš bazės, išsiųsti žinutes grafinei vartotojo sąsajai ir kt.



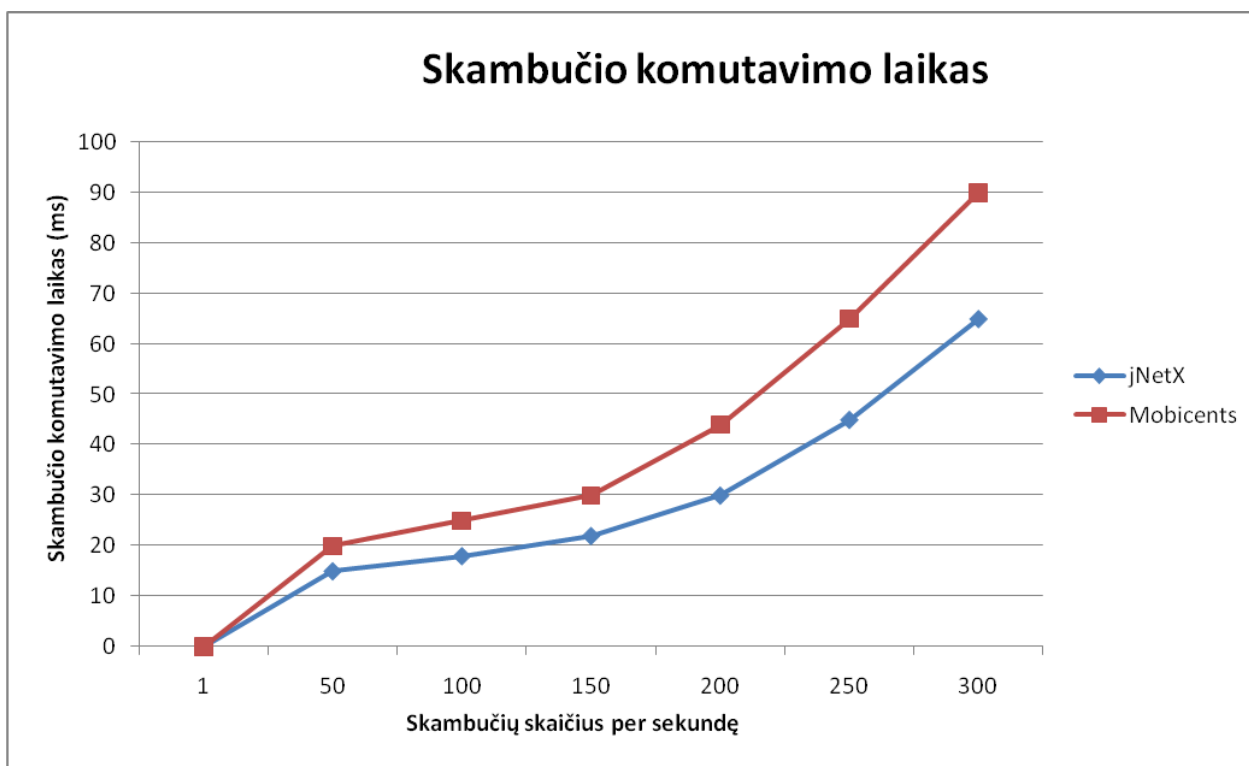
19 pav. Skambučiai per sekundę

Žemiau esančiame grafike (žr. 20 pav.) matome, kiek atminties naudojama esant skirtingam aktyvių skambučių skaičiui. Galima pastebėti, jog atminties sunaudojimas auga tiesiškai. jNetX platformos atveju, gauname jog vienam skambučiui reikia apie 17 KB atminties, o Mobicents platformoje jos reikia apie 26 KB vienam skambučiui. Realioje sistemoje komponentai bus gerokai didesni, todėl yra tikimasi ir didesnio atminties sunaudojimo. Tačiau toks koks jis buvo nustatytas testų metu yra pakankamai mažas, sistemoje palaikyti 1000 aktyvių skambučių užtenka viso labo ~20MB. Čia reikia pabrėžti ir JAIN SLEE architektūros specifiką – kiekvienam skambučiui yra sukuriama viena aplikacija – servisas, kuris yra laikomas atmintyje iki tol kol skambutis nebaigiamas. Beje pastebime, jog Mobicents serveris užima gerokai daugiau atminties net kai sistemoje visai nėra aktyvių skambučių. Bendrai žvelgiant, galima daryti išvadą, jog dėl naudojamos sistemos atminties problemų nekyla nei vienai platformai, kadangi vienam skambučiui jos sunaudojama sąlyginai mažai, o ir pati atmintis dabar pigi.



20 pav. Atminties naudojimas

Žemiau pateiktame grafike (žr. 21 pav.) matoma, kiek laiko užtunka skambučio sukmutavimas prie skirtingų sistemos apkrovimų. Galima pastebėti, jog didėjant skambučių skaičiui, skambučio komutavimo laikas didėja eksponentiškai, tačiau išlieka sąlyginai nedidelis net prie maksimalaus platformų palaikomo skambučių skaičiaus per sekundę. Reikia pastebėti ir tai, jog Mobicents platformoje jis yra didesnis, be to ir kyla sparčiau. Tačiau vis tiek išlieka priimtinas. Testavimo metu skambinantįjį ir galą į kurį skambinama emuliuojanti programinė įranga veikė tame pačiame kompiuteryje, todėl skambučio komutavimo laikuose neatsispinti vėlinimas tinkle ir kiti aspektai, tačiau iš to galime susidaryti puikų vaizdą, kiek vėlina pačios platformos. Netgi maksimalus vėlinimas ~100 ms yra tikrai nedidelis, nes jeigu panagrinėtume mobiliojo ryšio skambutį, pastebėtume jog kol pasigirsta signalas iš kito galo neretai pračina kelios sekundės.



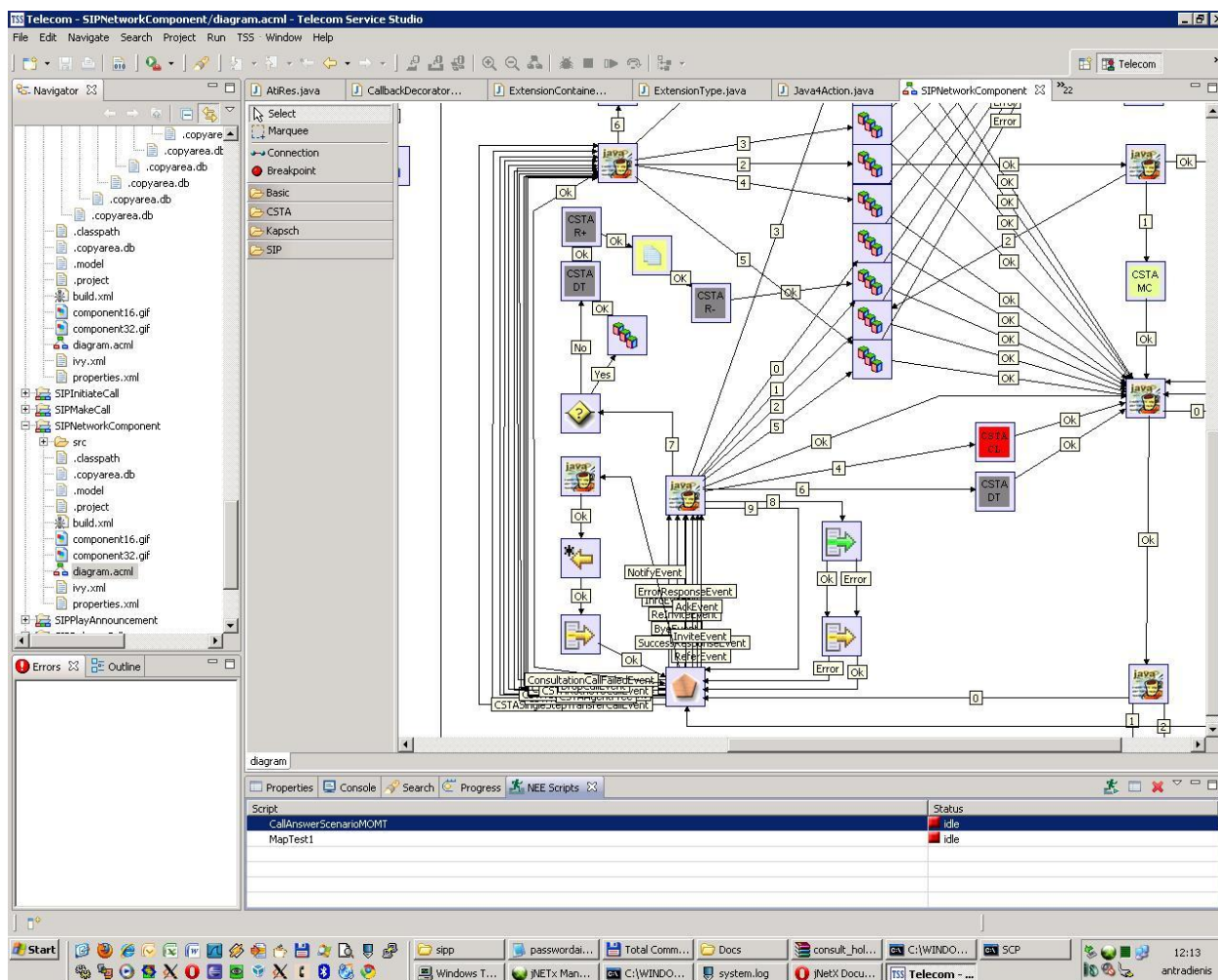
21 pav. Skambučio komutavimo laikas

3.3 Kūrimo aplinkos

Abi platformos turi joms skirtas kūrimo aplinkas, palengvinančias įvairių komponentų kūrimą, jų diegimą ir serverius. Kitaip nei ir patys serveriai, kūrimo aplinkos savo galimybėmis ženkliai skiriasi viena nuo kitos. Abi yra sukurtos kaip pagrindą paėmus atviro kodo kūrimo aplinką Eclipse.

jNetX OCFS platformai skirta kūrimo aplinka vadinasi Telecom Service Studio (TSS), tai pakankamai išbaigtas įrankis. Pagrindinė TSS savybė yra ta, jog servais kuriami naudojantis grafiniu redaktoriumi, tiesiog nubraižant reikiamą „būsenų mašiną“ ir įterpant kodą specialiai tam skirtuose vietose. Aplinka labai efektyviai realizuoja ir pakartotinio panaudojimo galimybę, tam sukūrusi specialią priemonę – aplikacijos bloką (angl. Application Building Block arba ABB). 22 paveiksle matome kaip atrodo šios diagramos. Ko gero rimčiausias šios kūrimo aplinkos trūkumas yra kaina (30 000 eurų).

Mobicents JAIN SLEE platformai skirtas kūrimo įrankis yra vadinamas EclipsLEE. Savo galimybėmis tai kur kas kuklesnis įrankis, leidžiantis šiek tiek automatizuoti tik būtiniausių komponentų kūrimą. Jis neturi grafinio redaktoriaus, todėl kuriant komponentus su juo darbas vyksta kur kas lėčiau. Tačiau nereikia pamiršti, jog tai atviro kodo ir nemokamas įrankis.



22 pav. TSS kūrimo aplinka

Žemiau pateiktoje lentelėje (žr. 6 lentelė.) galime matyti kokiomis galimybėmis pasižymi kiekviena iš kūrimo aplinkų. Kaip matome TSS kai kuriuos darbus atlieka automatiškai, kaip įvykių arba serviso blokų XML diegimo aprašų sukūrimas. TSS taip pat turi ir resursų adapterio projektą, nors jis ir nėra toks patogus, kaip grafinis serviso redaktorius, tačiau vis tiek sugeneruoja nemažą kodo dalį.

TSS dar turi tinklo ir skambučių emuliatorius projektą, kurie yra naudojami testuojant programinę įrangą, šių savybių dėka testuojant iš dalies galima apsieiti ir be paties serverio. Abu kūrimo aplinkos gali automatiškai įdiegti sukurtą komponentą į serverį.

6 lentelė. Pagrindinės kūrimo aplinkų savybės

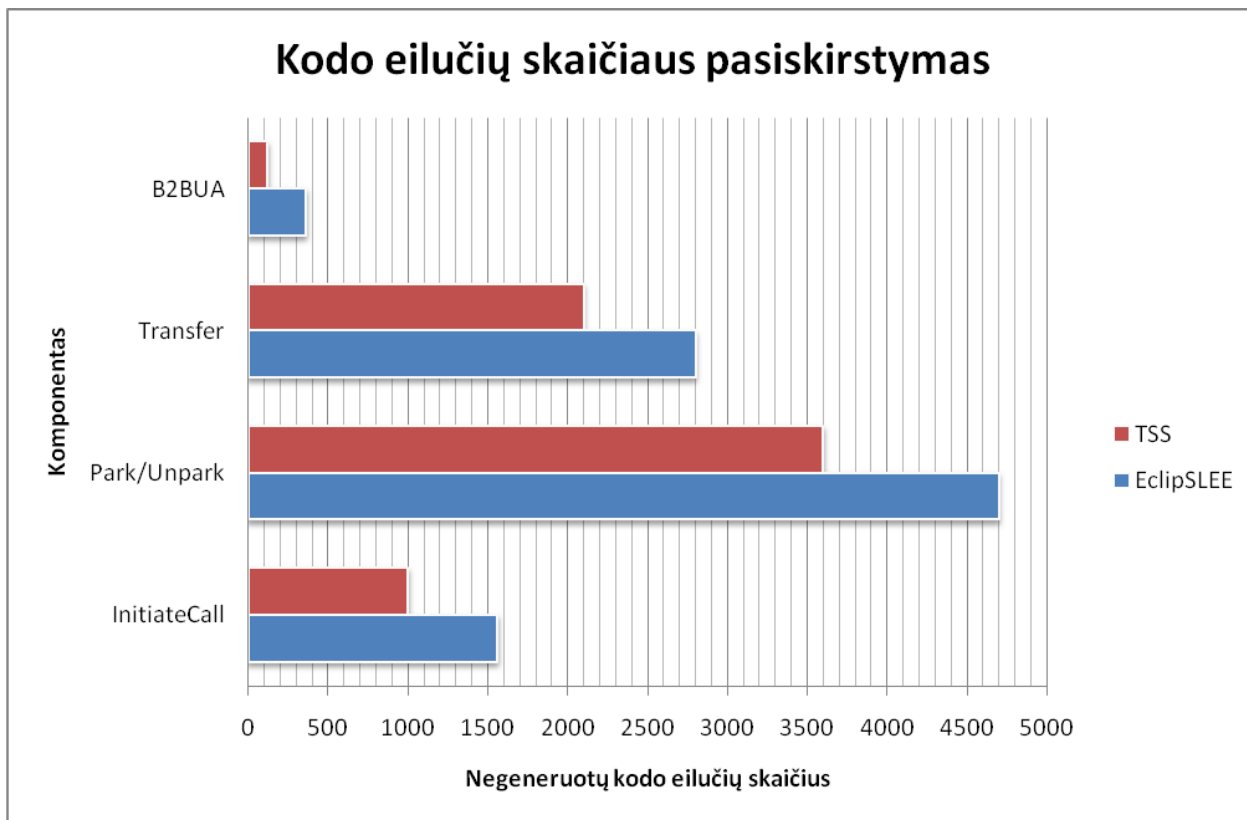
Savybė	Telecom Service Studio (TSS)	EclipSLEE
Serviso (SBB) projektas	+	+
Aplikacijos bloko (ABB) projektas	+	-
Skambučių generatorius projektas	+	-
Tinklo emuliatorius projektas	+	-
Resursų adapterio projektas	+	-
Automatinis komponento diegimas į serverį	+	+
Įvykiu (Events) projektas	+ (Resursų adapterio projektui sugeneruojamas automatiškai)	+
Serviso XML konfigūracijos projektas	+ (Serviso projektui sugeneruojamas automatiškai)	+

Kadangi TSS turi grafinį redaktorių, kuris generuoja kodą iš sudarytos diagramos, programuotojui tenka rašyti mažiau kodo, dėl to taupomas laikas, o diagramos braižosi sąlyginai greitai. Pabandėme paanalizuoti kiek kodo reikia parašyti analogiškiems komponentams abiejose kūrimo aplinkose. Rezultatai matomi lentelėje pateiktoje žemiau (žr. 23 pav.).

Skirtumas tarp kodo eilučių skaičiaus neatrodo labai žymus. Kuo komponentas mažesnis, tuo ženklesnis skirtumas, kadangi didelę dalį pradinio kodo susieto su SBB aprašymu sugeneruoja TSS. Skirtumą taip pat įtakoja, kiek ir kuriuo momentu laukiame įvykių iš tinklo – diagramoje užtenka nubrėžti atitinkamą perėjimą, o viskas kitkas yra sugeneruojama, kitokią situaciją matome EclipSLEE kūrimo aplinkoje – čia jau turime patys sukurti reikiamą kodą, kad užsisakytume šiuos veiksmus. Grafiniai elementai gali būti nesunkiai pakeisti nepasikartojančiu kodu, tam užtenka logiškai apirašyti dažniausia pasitaikančius veiksmus ir iškelti juos į atskirą biblioteką, taip gauname tiesioginį analogą TSS aplikacijų blokams.

Vistik padirbėjus su abejomis kūrimo aplinkomis, galima teigti, jog grafinis redaktorius yra tikrai patogesnis, nes komponento atliekamą logiką perprasti užtrunka kur kas mažiau laiko žvelgiant į diagramą, nei ieškant ryšių programos kode. Nors kodo apimties skirtumas ir neatrodo

didelis, tačiau laiko atžvilgiu analogišką komponentą sukurti EclipSLEE užtrunka gerokai daugiau laiko, nes didėjant komponentui darosi vis sunkiau tvarkyti kodą.



23 pav. Kodo eilučių skaičiaus pasiskirstymas pagal komponentą

Atlikus abiejų kūrimo aplinkų tyrimą, priėjome prie išvados, jog TSS yra neabejotinas lyderis tiek savo funkcionalumu tiek ir patogumu, tačiau jis ir kainuoja atitinkamai.

3.4 Egzistuojantys komponentai ir platformų savybės

Trumpai aptarsiu kokius protokolus ir kokias savybes palaiko platformos.

7 lentelėje galime pamatyti kokius protokolus palaiko kiekviena platforma.

7 Lentelė. Platformų palaikomi protokolai

Protokolas	jNetX OCFS	Mobicents JAIN SLEE
VXML	+	-
DNS	+	-
INAP	+	-
CAP	+	-
MM7	+	-
PAP	+	-
POP3	+	-
SIP	+	-

MAP	+	-
LDAP	+	-
Radius	+	-
USSD (MAP)	+	-
XMPP	-	+
SIP	+	+
Diameter	+	+
XCAP	-	+
HTTP	+	+
MGCP	-	+
SMPP	+	+
Viso palaiko protokolų:	16	7

Kaip matome šiuo atžvilgiu jNetX žymiai lenkia Mobicents platformą. Galime pastebėti, jog Mobicents nepalaiko SS7 šeimos protokolų, o tai yra žymus trūkumas, kadangi virtuali telefoninės stotelė naudoja būtent šios šeimos protokolus: CAP arba INAP. Dėl to reiktų sukurti savo resursų adapterius, kurie užtikrintų šių protokolų palaikymą. Be to ateityje planuojama, jog virtualiai telefoninei stotelei gali prireikti MAP ir SMPP protokolų palaikymo, kurių pagalba galima užtikrinti SMS žinučių siuntimą ir tokias paslaugas kaip USSD. Abi platformos palaiko SIP protokolą, o virtualios telefoninės stotelės vystymo plane yra numatyta, jog ji turi palaikyti ir šį protokolą. Šiuo metu sparčiai vystosi IMS sistemos ir šių sistemų pagalba dalis operatorių jau teikia ir VOIP paslaugas paremtas SIP protokolu.

8 lentelėje galima pamatyti, kokias svarbiausias savybes, kuri platforma palaiko. Aktualiausia savybė yra klasterio palaikymas, kas leidžia lengvai plėsti sistemos spartą, bei užtikrina nepertraukiamą jos darbą, net jei vieno iš serverių darbas sutrinka.

Galime pastebėti, jog tinklo emaliavimo režimą turi tik jNetX platforma. Šio emuliatorius pagalba galima labai nesunkiai suimituoti praktiškai bet kokių įvykių scenarijų, o tai aktualu ieškant klaidų sukurtoje sistemoje.

jNetX OCFS platforma turi ir jos konfigūravimo konsolę. Nors konfigūravimo darbas atliekamas retai, tačiau dėl konfigūracinių failų dydžio ši konsolė sutaupo tikrai nemažai laiko.

Abi platformos turi daugumą reikalingiausių savybių ir aiškaus lyderio išskirti negalima. Galbūt jNetX yra šiek tiek labiau subrendusi platforma, su ja paprasčiau dirbti.

8 Lentelė. Platformų savybės

Savybė	jNetX OCFS	Mobicents JAIN SLEE
Tinklo emaliavimas	+	-
JAIN SLEE 1.1	+	+
Klasterio palaikymas (angl. Cluster)	+	+

Valdymas per SNMP	+	+
Valdymas per JMX	+	+
Valdymas per atskirą konsolę	SMP Console	Jopr Console
Platformos komponentų konfigūravimo konsolė	+	-
Komponento įdiegimas ar atnaujinimas netaikant serverio	+	+
Palaikomos platformos	Microsoft Windows 2000, XP, Vista, 7, 2003/2007 Server; Linux, Sun Solaris	Microsoft Windows 2000, XP, Vista, 7, 2003/2007 Server; Linux, Sun Solaris

3.5 Tyrimo išvados

Atlikus abiejų platformų tyrimą buvo prieita prie tokių išvadų:

- Savo svarbiausiomis savybėmis abi platformos yra panašios
- Abiejų platformų greitaveika yra panaši
- jNetX platforma turimų komponentų gausa lenkia Mobicents platformą
- Mobicents nepalaiko reikiamų protokolų – CAP ir INAP
- jNetX TSS yra patogesnis ir daugiau galimybių teikiantis įrankis už EclipSLEE

Remiantis turimomis išvadomis buvo priimtas sprendimas, jog migracija į Mobicents platformą yra tikslinga. Norint sėkmingai į ja migruoti reikės susikurti papildomų resursų adapterių, dirbančių su SS7 šeimos protokolais. Yra tikimasi, jog perkėlimo darbai pareikalaus papildomo darbo dėl sąlyginai nepatogios kūrimo aplinkos. Tačiau šiuos nepatogumus ir papildomas išlaidas atpirks faktas, jog platforma yra nemokama ir atviro kodo, o tai leis žymiai sumažinti virtualios telefoninės stotelės galutinę kainą.

IŠVADOS

1. Analitinėje darbo dalyje buvo aptarta JAIN SLEE architektūra bei kliento serverio sistemos architektūra. Aptartos galimos projektavimo problemos ir jų sprendimo būdai.

2. Išanalizavus užsakovo iškeltus reikalavimus sistemai, nustatyti penkti probleminiai sistemos realizavimo uždaviniai: sistemos darbo patikimumas, vartotojo duomenų saugumas, didelio skambučių kiekio palaikymas, integracija su trečių šalių valdymo konsolėmis, integracija su operatorių įranga. Šiems probleminiams uždaviniams išspręsti buvo pateikti jų galimi sprendimai.

3. Atsižvelgus į sistemai keliamus vartotojo reikalavimus ir numatomus poreikius buvo pasirinktos sistemos realizavimo technologijos – jNetX OCFS, Java, C++, Jetty, Tapestry, SQL, CSTA III XML.

4. Apžvelgti egzistuojantys sprendimai, palygintos jų savybės. Taip pat, įsitikinta, jog rinkoje nėra daug analogiškų produktų, o virtuali telefoninė stotelė ras savo nišą.

5. Remiantis suformuluotais projektiniais, techniniais bei probleminių uždavinių sprendimais, realizuota virtualios telefoninės stotelės ir operatoriaus konsolės programinė įranga. Sistemos elementai išdėstyti kliento serverio architektūra.

6. Tiriamojoje ir eksperimentinėje dalyje buvo palygintos dvi JAIN SLEE platformos – jNetX OCFS ir Mobicents JAIN SLEE. Tyrimo tikslas buvo nustatyti ar Mobicents platforma yra tinkamas kandidatas virtualiai telefoninei stotelei. Tyrimo metu nustatyta, jog platformos yra panašios savo veikimo charakteristikomis. Mobicents platforma šiek tiek atsilieka turimais komponentais, bei turi nusileidžiančia savo kūrimo aplinką. Tačiau atsižvelgiant į platformų kainų skirtumą, buvo nuspręsta jog virtualią telefoninę stotelę apsimoka perkelti į Mobicents platformą.

LITERATŪRA

- [1] JAIN SLEE architektūros aprašas. JAIN SLEE Fundamentals [Žiūrēta 2010 05 12], prieiga internete <http://www.jainslee.org/slee/fundamentals.html>
- [2] Cerrone, Emma (2002), Mobile Phone Users: A Small-Scale Observational Study [Žiūrēta 2010 05 12], prieiga internete <http://www.aber.ac.uk/media/sections/go.php?url=http://www.aber.ac.uk/media/Students/elc0201.doc&f=1>
- [3] Fowles, D. *Top 3 Personal Finance Software* [Žiūrēta 2010 05 12], prieiga internete: <http://financialplan.about.com/od/software/tp/MoneySoftware.htm> BroadSoftware
- [4] BroadWorks produkto aprašas. BroadWorks Overview. [Žiūrēta 2010 05 12], prieiga internete: www.fujitsu.com/downloads/AU/BroadsoftDataSheetOverview.pdf
- [5] BroadWorks, BEA WebLogic Editon. [Žiūrēta 2010 05 12]. Prieiga internete: http://www.bea.com/content/news_events/white_papers/BBWE_3.0_ds.pdf
- [6] Dr. Richards E. Smith, Principal Systems Engineering "Internet Cryptography". 1998 m. [Žiūrēta 2010 05 12] Prieiga internete: <http://www.visi.com/crypto/>
- [7] Joan Daemen, Vincent Rijmen „The Design of Rijndael,,. 2002 m. ISBN 3-540-42580-2
- [8] Tom Miller (Siemens, September 2007). „ECMA International. CSTA overview.“ [Žiūrēta 2010 05 12], prieiga internete: <http://www.ecma-international.org/activities/Communications/TG11/CSTAoverview.pdf>
- [9] Jim McGee. "Legacy Systems: Why History Matters", Enterprise Systems Journal, 2005-11-10. [Žiūrēta 2010 05 12], prieiga internete: <http://esj.com/news/article.aspx?EditorialsID=1529>

TERMINŲ IR SANTRUMPŲ ŽODYNAS

Santrumpa, terminas	Paiškinimas
PBX	telefoninė biuro stotelė. Angl.: Private Branch eXchange
VPBX, HPBX	virtuali telefoninė biuro stotelė. Paprastai tai programinės ir aparatinės įrangos derinys veikiantis telekomunikacijų paslaugų tiekėjo pusėje. Angl. Virtual Private Branch, Hosted Private Branch eXchange
Callback	perskambinimo paslauga, kuri automatiškai perskambina vartotojui, kuris buvo užimtas skambinimo metu, baigus pokalbį
DND	netrukdyti. Angl.: Do Not Disturb
Call Park	skambučio priparkavimas, kai skambutis perduodamas į laisvą parkavimo vietą, ir bet kas žinantis kur jis buvo priparkuotas gali jį išparkuoti
FindMe-FollowMe	tai tokia paslauga kai paeiliui yra skambinama į kelis nurodytus telefonus, bandant surasti vartotoją. Telefonų sąrašas gali kisti priklausomai nuo paros laiko
Blind Transfer	skambučio persiuntimas, kai skambutis tiesiog persiunčiamas kitam vartotojui
Supervised Transfer	skambučio persiuntimas, kai prieš peradresuojant skambutį į paklausiamą vartotojo ar jis nori priimti peradresuojamą skambutį
Pick up groups	grupės telefonų, į kurią įeinantį skambutį gali pakelti bet kuris iš jų
Conference call	konferencinis skambutis, kai vienu metu galima kalbėtis su keliais vartotojais
ACD groups	telefonų grupės tarp kurių įeinantys skambučiai yra paskirstomi automatiškai. Angl.: Call Distribution
FVM	skambučio persiuntimas į balso pašto dėžutę. Angl.: Forward to Voice Mail
CSTA	Protokolas skirtas kompiuterio palaikomoms telekomunikacijų programoms. Angl. Computer Supported Telecommunications Applications
CSTA XML	Protokolas skirtas kompiuterio palaikomoms telekomunikacijų programoms, paremtas XML žinutėmis, yra CSTA 3 standarto dalis. Angl. Computer Supported Telecommunications Applications Extensible Markup Language
CTI	kompiuterinės telefonijos integracija, užtikrina integraciją tarp kompiuterių ir telefonų. Angl.: Computer Telephony Integration
QoS	tinklo duomenų srauto kontrolės sistema. Angl.: Quality of Service
MSC	Pagrindinė mobiliojo ryšio paslaugų operatoriaus aparatinė ir programinė įranga aptarnaujanti mobiliuosius telefonus.

	Angl.: Mobile Switching Center
jNetX OCFS	jNetX sukurta platforma skirtas suteikti atvirą priėjimą prie skirtingos aparatinės ir programinės telekomunikacijose naudojamos įrangos . Angl.: jNetX Open Convergence Feature Server
E1	standartizuota 32 balsinių kanalų telekomunikacijose naudojama duomenų perdavimo magistralė
JAIN SLEE	Plačiai paplitęs terminas telekomunikacijose – Java sąsaja intelektualiems tinklams, servisų logikos vykdymo aplinka. Tai didelio pralaidumo, mažo reakcijos laiko įvykius apdorojanti programų aplinka. (Angl.: Java APIs for Integrated Networks, Service Logic Execution Environment)
CPS	skambučiai per sekundę, telekomunikacijose palčiai paplitęs terminas. (Angl.: Calls Per Second)
jVPBX	virtualios telefoninės stotelės programine įranga veikianti jNetX OCFS serveryje
SBServer	žinučių serveris (trumpinys nuo „switchboard server“)
Originating app	virtualios telefoninės sąsajos komponentas, skambučių inicijuojančios pusės aplikacija
Terminating app	virtualios telefoninės sąsajos komponentas, skambučių priimančios pusės aplikacija
CSTAIIXML RA	CSTA III XML protokolo resursų adapteris
VPDataService	grafinės vartotojo sąsajos komponentas, užtikrina komunikaciją su tinklu – žinučių serveriu.
MainFrame	grafinės vartotojo sąsajos komponentas, šakinis komponentas, kuris apdoroja pagrindinę logiką, sudaro karkasinį langą į kurį integruojasi OCX komponentai.
VPExtCtrl	grafinės vartotojo sąsajos komponentas, OCX technologija paremtas komponentas skirtas telefonams ir jų būsenai vaizduoti
VPCallCtrl	grafinės vartotojo sąsajos komponentas, OCX technologija paremtas komponentas skirtas skambučiams vaizduoti
CSTA3Client	žinučių serverio komponentas, užtikrina sąsają su CSTAIIXML RA, prie jo jungiasi kaip klientas
CSTA3Server	žinučių serverio komponentas, užtikrina sąsają su grafinėmis vartotojo sąsajomis, kurios prie jo jungiasi kaip klientai
CSTA3Devices	saugo telefonų būsenas ir esamus skambučius, kadangi to nedaro jVPBX
SBB	JAIN SLEE platformos servisas (angl. Service Building Block)
ABB	jNetX OCFS platformos servisą sudaranti aplikacija (angl. Application Building Block)
CRM	Santykių su klientais valdymo sistemos (angl. Customer Relationship Management)

SIP	Sesijos kontrolės protokolas (angl. Session Initiation Protocol)
UA	Vartotojo įrenginys – telefoną (angl. User agent)
UAC	Skambinantis telefonas (angl. User Agent Client)
UAS	Telefonas į kurį skambinama (angl. User Agent Server)
B2BUA	Tarpinis įrenginys, maršrutizatorius (angl. Back to Back User Agent)
USSD	Protokolas užtikrina tekstiniu meniu paremtas paslaugas mobiliesiems telefonams (angl. Unstructured Supplementary Service Data)