

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**DINAMISKU RISINĀJUMU IEVIEŠANA
TELEKOMUNIKĀCIJU PAŠAPKALPOŠANĀS SISTĒMAS
UZLABOŠANĀ
MAGISTRA DARBS**

Autors: **Ronalds Brēmers**

Studenta apliecības nr.: rb09137

Darba vadītājs: Asoc.prof., Dr. dat. Jānis Zuters

RĪGA 2016

ANOTĀCIJA

Darbā skaidrots, kā tika pētīta un analizēta esošā telekomunikāciju uzņēmuma pašapkalpošanās sistēma un kā tā tika uzlabota. Tika identificētas vairākas problēmas, kas traucēja gan uzņēmuma darbiniekiem, gan klientiem, kas ir sistēmas lietotāji. Balstoties uz šiem trūkumiem, tika izveidotas prasības jaunajai sistēmai, kura tika sadalīta divās daļās, viena daļa optimizēta privātpersonām jeb klientiem ar salīdzinoši mazu lietotāju skaitu, otra daļa veidota pēc juridisku personu klientu vēlmēm un vajadzībām. Sistēmas servera daļas projektēšanas laikā tika izplānoti un beigās arī ievesti vairāki dinamiski risinājumi, kas atvieglo informācijas sistēmas uzturētāju darbu, ieviešot sistēmā izmaiņas, gan klientu iespēju tiešsaistē prognozēt savu nākamo rēķinu. Noslēgumā tika izpētīta un izanalizēta kāda ir jaunizveidotās sistēmas datu ieguves ātrdarbība un salīdzināta ar iepriekšējo risinājumu.

Atslēgvārdi:

Pašapkalpošanās sistēma, Oracle, servera puses projektēšana.

ABSTRACT

Improving telecommunications self service system using dynamic solutions.

The aim of this masters thesis, is to explain how the analysis of the current self-service system was done and how it was improved. Many problems were identified that interfered with the work of company employees and also with customers, who are the main users of this system. The business requirements of the new system were designed keeping the flaws of the existing system in mind. The system was split in two parts, one was optimized for natural persons, clients with relatively small user count, the other was designed primarily for the needs of juridical persons. There were several dynamic solutions planned and implemented, that made information systems support work easier when making changes to the system, and that allowed clients to predict their next bill in real-time. In conclusion, the performance of old and new systems was compared and analyzed.

Keywords:

Self-service system, Oracle, back-end designing.

AUTOREFERĀTS

Darbā, papildus sistēmas projektējumam, ir aprakstīts tāds diezgan mazizplatīts risinājums kā dinamiskie vaicājumi, kas publiskajā telpā nav pārāk plaši skatīti, iespējams tādēļ, ka tos pielietot šādā veidā ir nepieciešams tikai ļoti specifiskos gadījumos. Uzņēmuma uzturēšanas fāzē resursu patēriņš uz jaunieviešamām izmaiņām ir stipri samazinājies.

Tā kā nav atrodami zinātniski raksti, kur ir aprakstīti pašapkalpošanās portālu projektējumi, kā pamatu izmantojot datubāzi, literatūras izpēte reducējas uz atsevišķu mazu rakstu lasīšanas un, vajadzības gadījumā pielietošanas. Visi avoti ir no lietotāju bāzētu, foruma veidu interneta vietnēm, kas gan nemazina to nozīmību darba veidošanā.

Darbā aplūkotās esošās sistēmas problēmas ir aprakstītas virspusīgi, taču pietiekami saturīgi, lai lasītājs saprastu kāpēc projektējumā tika pieņemti tieši tādi lēmumi. Darba teksta radīšanai veltītas aptuveni 120-150 stundas.

Praktiskajā, jeb projektēšanas un izpētes darbā ir ieguldīts pāri 500 stundām, testēšanas un uzturēšanas darbos ieguldītas vēl aptuveni 200.

Iegūtos rezultātus ir ļoti grūti salīdzināt ar jebko citu, jo, kā jau minēts, pašapkalpošanās portāli parasti netiek izklāstīti zinātniskos rakstos. Vienīgais ar ko autors var salīdzināt izveidotā projektējuma veiksmīgumu ir tas, ka cits Latvijas telekomunikāciju uzņēmums ir uzbūvējis, vismaz dizainā un funkcionalitātē līdzīgu portālu, kur līdzīga funkcionalitāte nav tik veiksmīgi ieviesta, lai gan servera puses arhitektūras izskats nav pieejams, tāpēc precīzi salīdzināt nav iespējams. Iegūtajos rezultātos jāvadās pēc sistēmas lietotāju un kolēģu atsauksmēm, kas lielākajā vairumā gadījumu bijušas pozitīvas un ļoti retu reizi, kad ir konstatēta kļūda sākotnējā projektējumā. Sistēmas pirmā daļa tiek izmantota produkcijā jau kopš 2015. gada maija, savukārt otrā daļa tika iedarbināta produkcijā 2016. gada maijā.

Darbā nevajadzētu būt ievades un valodas kļūdām, tas tika pārbaudīts no diviem neatkarīgiem avotiem.

Darba struktūra un noformējums, lielākoties, atbilst Latvijas Universitātes Datorikas Fakultātes maģistra izstrādes un aizstāvēšanas metodisko norādījumu 3. nodaļai, darbā iztrūkst tikai nodaļa 'rezultāti un diskusija', kas, autora uzskatā, šajā darbā nebija nepieciešama.

Neviens attēls nav aizgūts no cita autora, taču visas idejas, formulējumi, kas aizgūti vai radušies diskusijā ar citiem cilvēkiem, ir minēti darbā, kā arī tie, kas ar savu atbalstu ir palīdzējuši darba tapšanā ir minēti pie pateicībām. Darba tekstā nav burtisku tulkojumu vai tuvu pārstāstu no jebkura literatūras avota, ja neskaita apzīmējumu sarakstu.

SATURS

APZĪMĒJUMU SARAKSTS	1
IEVADS	2
1. ESOŠĀ VAAC SISTĒMA.....	4
1.1. Vispārējā datubāzes uzbūve	4
1.2. VAAC sistēmas uzbūve	5
1.2.1. Datu shēmas struktūra	6
1.2.2. Datu iegūšana	8
1.3. Sistēmas trūkumi.....	9
1.3.1. Biznesa un lietotāju trūkumi.....	9
1.3.2. IS uzturētāju problēmas	10
2. JAUNĀS SISTĒMAS PRASĪBAS	12
2.1. Funkcionālās prasības	12
2.1.1. Provizoriskais rēķins	12
2.1.2. Klienta iesniegumu apstrādes automatizācija.....	12
2.1.3. Lietotāja anulēšana	13
2.1.4. Masveida pakalpojumu pieslēgšana	13
2.1.5. Ilgtermiņa lietotāja paradumu statistika	13
2.1.6. Top10 zvanītie numuri	14
2.1.7. Jauna lietotāja izveidošana	14
2.2. Nefunkcionālās prasības	14
2.2.1. Reāllaika datu iegūšana	14
2.2.2. Datu iegūšanas ātrums piesakoties	14
2.2.3. Iesniegumu apstrādes ātrums.....	15
3. IZAICINĀJUMI PRASĪBU IZPILDĒ.....	16
3.1. Esošā projektējuma uzlabošana	16
3.1.1. Uzlabot datu ieguves ātrumu	16
3.1.2. Saglabāt resursu patēriņu esošajā līmenī.....	16

3.1.3.	Uzlabot drošības līmeni.....	16
3.2.	Jaunievedumu ieviešanas izaicinājumi	17
3.2.1.	Paātrināt izmaiņu ieviešanu.....	17
3.2.2.	Rēķinu pieejamība uzreiz pēc izveidošanas	18
3.2.3.	Klientu dalījums pēc lietotāju skaita/atrādāmās informācijas.....	18
4.	PRIVĀTPERSONU SISTĒMAS PROJEKTĒJUMS	20
4.1.	Datu ieguves shēma	20
4.2.	Datu ieguve (procedūras).....	21
4.3.	Asinhronie procesi	22
4.3.1.	Iesniegumu apstrādes process	22
4.3.2.	Lietotāju paradumu statistikas aprēķina process	22
4.3.3.	Top10 zvanīto numuru process	23
4.4.	Datu strukturēšana	23
4.5.	Datu drošība	23
4.6.	Datu konfigurācija	24
5.	DINAMISKU RISINĀJUMU IEVIEŠANA	27
5.1.	Atgriežamo datu izmaiņu vadība	27
5.1.1.	Privātpersonu datu atlases shēmas trūkumi.....	27
5.1.2.	Dinamiskā datu vaicājumu izmaiņu vadība.....	27
5.2.	Provizoriskais lietotāja rēķins	29
5.2.1.	Risinājuma izvēle	29
5.2.2.	Dinamiska prognozējamā rēķina izveidošana	30
5.3.	Konfigurējamo plānu lietošanas integrācija	30
6.	JURIDISKU PERSONU SISTĒMAS PROJEKTĒJUMS.....	32
6.1.	Privātpersonu un juridisko personu dalījums.....	32
6.2.	Datu ieguve (procedūras).....	32
6.3.	Asinhronie procesi	33
6.3.1.	Lietotāju bilanču process.....	33

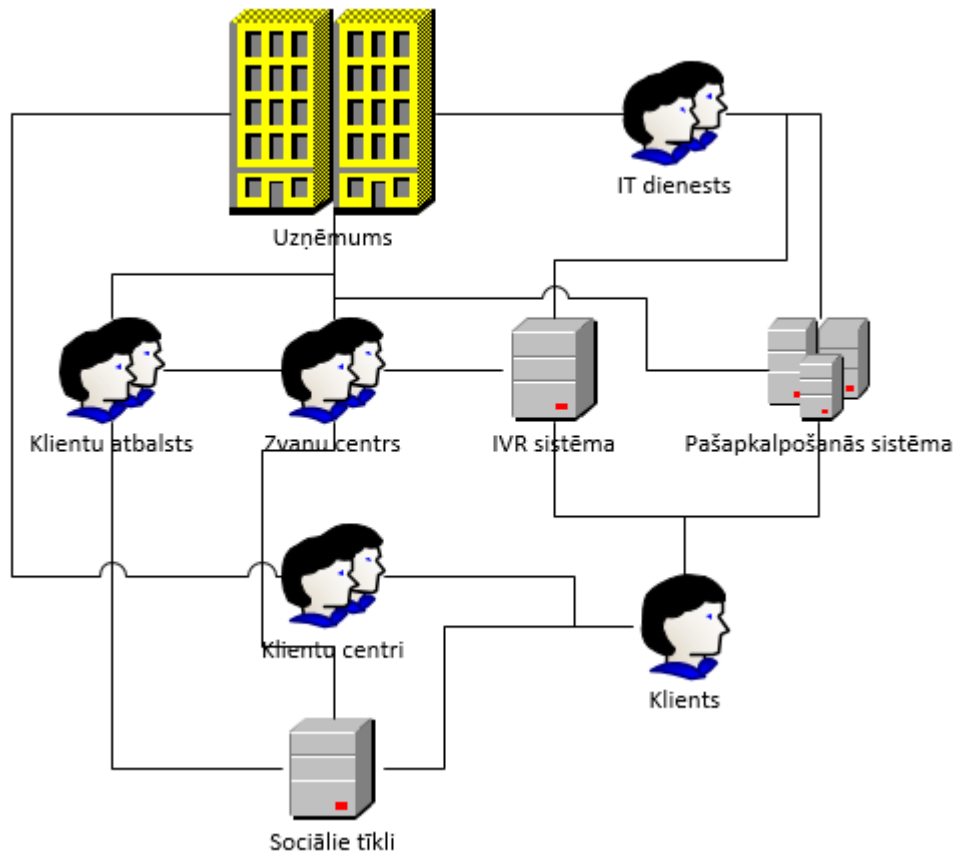
6.3.2.	M2M lietotāju datu process	33
6.3.3.	Iekārtu datu savākšanas process	34
6.3.4.	Pēdējā rēķina rindu process	35
6.4.	Papildus datu ieguve	35
6.4.1.	Lietotāju pamatdatu saraksts	35
6.4.2.	M2M lietotāju pamatdatu saraksts	36
6.5.	Datu konfigurācija	36
7.	PRODUKCIJAS SISTĒMAS ĀTRDARBĪBA	38
7.1.	Datu ieguves ātrdarbība piesakoties.....	38
7.1.1.	Privātpersonas	38
7.1.2.	Juridiskas personas	39
7.1.3.	Vecās sistēmas ātrdarbība.....	41
7.1.4.	Vecās un jaunās sistēmas ātrdarbības salīdzinājums.....	42
7.2.	Pārējo procedūru ātrdarbība.....	43
7.3.	Asinhrono procesu ātrdarbība	44
7.3.1.	Lietotāju bilanču process	44
7.3.2.	M2M lietotāju process	45
7.4.	Iesniegumu apstrādes ātrums	45
	SECINĀJUMI	48
	PATEICĪBAS.....	49
	IZMANTOTĀ LITERATŪRA UN AVOTI.....	50

APZĪMĒJUMU SARAKSTS

Schema(shēma)	Datubāžu kontekstā, shēmās tiek strukturēti sadalītas tabulas, procedūras un pakotnes, pārskatāmības nolūkos[1]
PDF	Pārnēsājamo dokumentu formāts, kas nav atkarīgs no lietotās operētājsistēmas, programmatūras vai datora fiziskajām komponentēm[2]
Unicode	Simbolu kodējuma veids, kas standartizēti tiek izmantots lielākajā daļā no pasaules digitālajām rakstāmmašīnām[3]
SQL	Strukturēto datu valoda, kas paredzēta datu manipulēšanai relāciju datu bāžu sistēmās[4]
PL/SQL	Oracle radīta procedurālā programmēšanas valoda [5]
PHP	Atklātā pirmkoda skriptu valoda, kas ir no platformām neatkarīga, kas sākotnēji bija paredzēta servera puses lietojumos dinamisku tīmekļa lapu ģenerēšanai[6]
XML	Strukturēta teksta iezīmēšanas valoda [7]
UTF-8	Astoņu bitu unikoda pārveidošanas formāts[8]
nvarchar	Unikodu atbalstošs Oracle datubāzes teksta datu tipa mainīgais[11]

IEVADS

Uzņēmuma pašapkalpošanās sistēma, jeb Virtuālais abonentu apkalpošanas centrs (VAAC) ir viens no galvenajiem veidiem kā uzņēmumam komunicēt ar klientu un kā klientam mijiedarboties ar sistēmu.



Attēls 0.1. Klienta komunikācijas kanāli ar uzņēmumu

Pašreizējās iespējas klientam komunicēt ar uzņēmuma darbiniekiem ir sekojošas

- Zvanu centrs
- Pašapkalpošanās sistēma
- Sociālie tīkli
- Klientu centri

Tieši VAAC sistēmas mērķis un jēga ir ietaupīt uzņēmuma administratīvos resursus un klienta laiku, vienkāršās darbības automatizējot un dodot iespēju klientam pašam pārvaldīt savus lietotājus un ļaujot to visu izdarīt no jebkuras vietas pasaulē. Tā ir galvenā saikne lietotājam ar saviem pieslēgumiem, kur vienmēr ir iespējams apskatīties vai ir pieslēgts viss vajadzīgais, lai dotos ārzemju ceļojumā, cik daudz interneta ir palicis mobilajai iekārtai, nomainīt pakalpojumu plānu savam numuram, kad tas ir nepieciešams, kā arī ir iespējama rēķinu apmaksa un saziņa ar uzņēmuma darbiniekiem. Situācijā, kad šī sistēma ir morāli

novcojusi, komunikācija ar klientu kļūst sarežģītāka, jo izmainoties pārējai datubāzei, atbalsts šai sistēmai kļūst sarežģītāks, tāpēc ik pa laikam virtuālo klientu centru nepieciešams uzlabot. Tāpat ir iespējams arī apskatīt informāciju par iztērēto mēneša ietvaros un tādā veidā prognozēt aptuvenās rēķina izmaksas.

Ne visiem gan sistēmā ir pieejamas vienādas iespējas. Parastā gadījumā veikt jebkādas izmaiņas klienta lietotājiem tiesības ir tikai galvenajam lietotājam, kas, lielākoties, ir rēķina maksātājs, pārējiem lietotājiem ir iespējams apskatīties informāciju tikai par sevi un nekādas izmaiņas veikt nav atļauts.

Iepriekšējā uzņēmuma VAAC sistēma sāka darbu 2006. gadā un šajā laikā ir gan tehniski gan morāli novecojusi un nespēja apmierināt ne mūsdienīgas dizaina, ne datu atlasē prasības. Tieši šo iemeslu dēļ tika pieņemts lēmums atjaunot šo sistēmu no pašiem pamatiem, jeb datu ieguves no datubāzes, kas arī ir autora, kā sistēmu analītiķa, uzdevums un ar ko šī uzlabošana vispār tika sākta. Projektēšanas ietvaros tika izmainīta pilnīgi visa datu atlasē, kas, galvenokārt tika pārveidota uz Oracle procedūrām, savukārt datu atgriešanā tiek izmantots teksta fails, kas ir strukturēts kā XML objekts. Procedūru sistēma, savukārt, atvieglo gan atgriežamo datu apstrādi, gan drošību.

Darbā galvenokārt tiek skarta VAAC sistēma un pārējās shēmas tiek pieminētas epizodiski.

Darba nodaļās ir aprakstīts kā tika izpētīta pašlaik lietotā sistēma un kas tika identificēti kā trūkumi, kādas papildus darījumprasības parādījās izpētes laikā, kādas nepieciešamās sistēmas izstrādes bija jāveic, lai tās apmierinātu, kā arī tika izpētīta produkcijas risinājuma ātrdarbība un salīdzināta ar veco produktu.

Maģistra darba mērķis ir skaidrot kā tika veikta sistēmas izpēte un kā veiksmīgāk realizēt prasības, kas identificētas analīzes laikā.

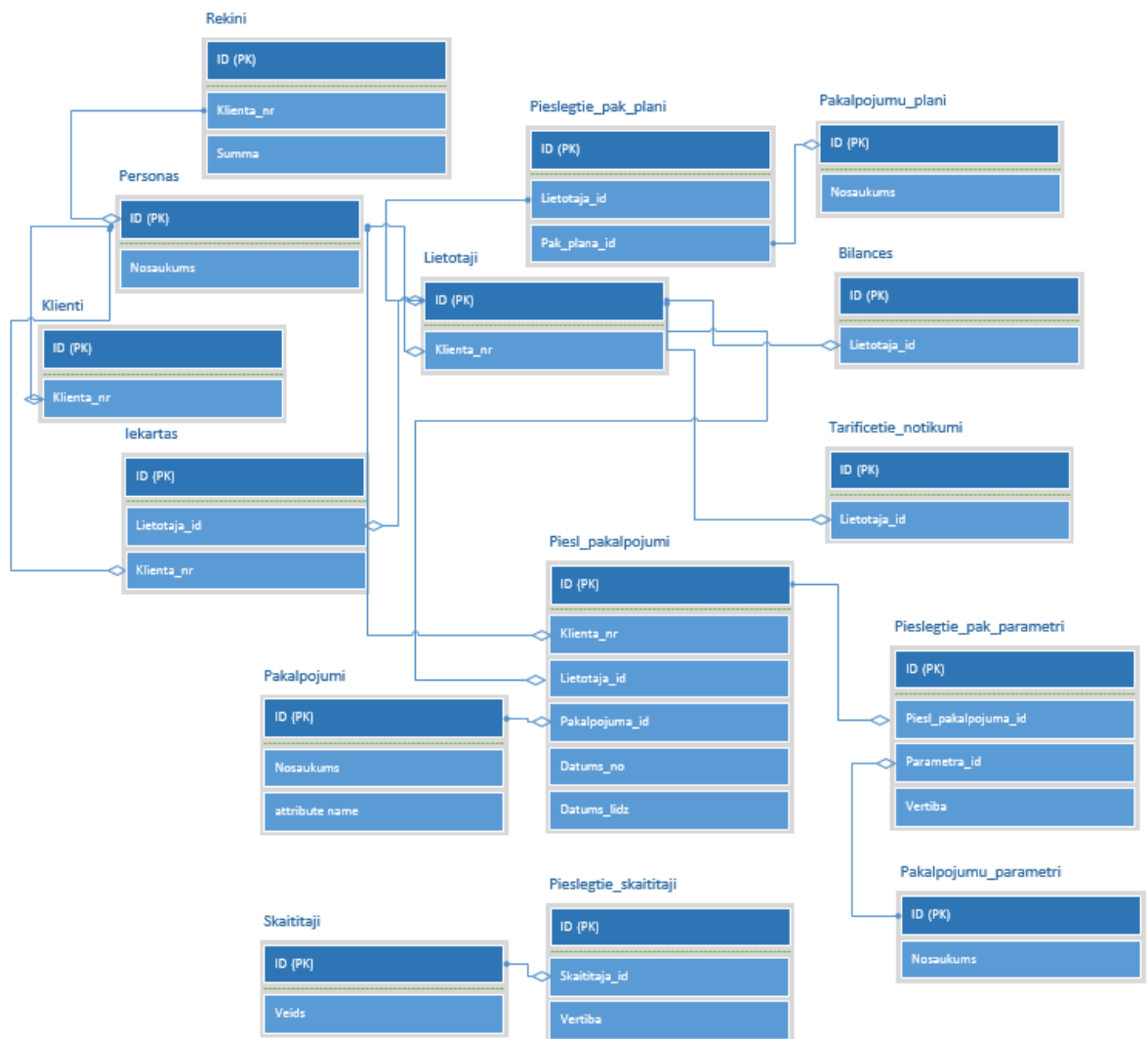
Darbs sastāv no 2 anotācijām, autoreferāta, apzīmējumu saraksta, 7 nodaļām, secinājumiem un izmantotā literatūras saraksta ar 11 tiešsaistes avotiem.

1. ESOŠĀ VAAC SISTĒMA

1.1. Vispārējā datubāzes uzbūve

Pirms tiek veikti kādi uzlabojumi, primārais ir izpētīt esošo sistēmas arhitektūru, lai būtu zināms kā darbs tiek organizēts pašlaik un kuras ir risinājuma vājās vietas. Uzņēmumā galvenā datubāze ir Oracle Database 10g. Tā ir sīkāk sadalīta loģiskās shēmās(schema), lai būtu atsevišķi sagrupētas līdzīgās tabulas.

- Pakalpojumu konfigurācijas shēma
- Klientu, pakalpojumu un pieslēgumu shēma
- Rēķinu shēma
- Iekārtu shēma
- Sistēmas lietotāju un darbinieku shēma

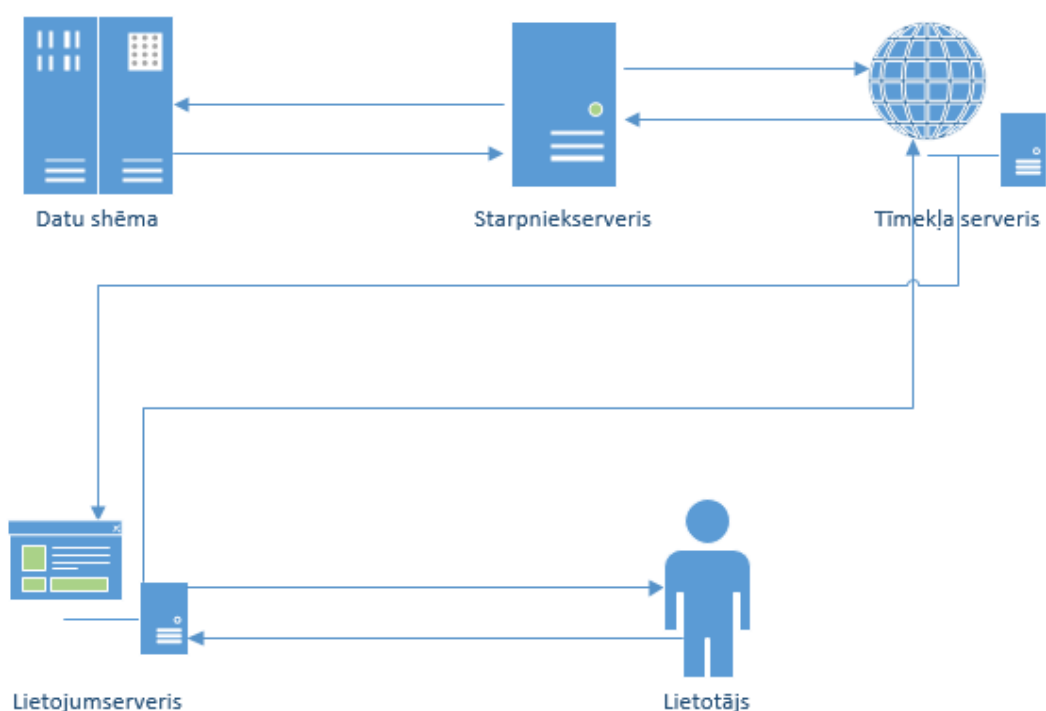


Attēls 1.1. Vienkāršots centrālās datu bāzes modelis

Kā arī shēmas katrai integrētajai ārējai sistēmai, piemēram:

- Piedāvājumu vadības sistēmas shēma
- IVR shēma
- Kontaktu vadības sistēmas shēma
- Incidentu vadības sistēmas shēma
- Pašapkalpošanās (VAAC) sistēmas shēma

1.2. VAAC sistēmas uzbūve



Attēls 1.2. VAAC sistēmas dizains

Datu shēma ir tā, kur atrodas visi VAAC sistēmas dati, no turienes tie tiek caur starpniekserveri padoti tīmekļa serverim, kur tos apstrādā un caur lietojumserveri lietotājs ar tiem var strādāt. Datu satiksme notiek abos virzienos – vienā virzienā datu atrādīšanai lietotājam, otrā virzienā ir lietotāja veidoto iesniegumu nogādāšana datu shēmā apstrādei.

No lietotāja puses, tas izskatās šādi



Attēls 1.3. Sistēmas lietotāja ceļš pieteikšanās brīdī

Ja lietotājs autorizējas ar pieslēguma numuru, tad piekļuves tiesību līmenis tiek noteikts pēc pakalpojuma esamības, ja tam ir pieslēgts pakalpojums “Galvenais pieslēgums”, tad ir pieeja visiem klienta datiem un ir tiesības rediģēt klienta datus un veikt darbības ar visiem pieslēgumiem. Ja lietotājam nav šāda pakalpojuma, tad ir pieejami dati tikai par savu pieslēgumu. Ja lietotājam ir pakalpojums “VAAC lietošanas aizliegums”, tad lietotājam pieejas sistēmai nav. Šāds pakalpojums, gan, tiek pieslēgts reti un tikai pēc paša klienta lūguma.

1.2.1. Datu shēmas struktūra

VAAC sistēmas datu shēma sastāv no šādām tabulām, kur glabājas visi dati, kuri ir redzami lietotājam

VAAC_KLIENTI

Informācija par klientiem, to nosaukums vai vārds un uzvārds, norādītās rēķinu un e-pasta adreses, pēdējā rēķina summa un apmaksas statuss.

VAAC_LIETOTAJI

Informācija par lietotājiem, to vārds un uzvārds, pieslēguma numurs, pakalpojumu plāns, reģistrēšanas datums, pēdējās aktivitātes datums.

VAAC_KLIENTU_REKINI

Informācija par klienta rēķiniem, to summa, izveidošanas un apmaksas datumi.

VAAC_PAKALPOJUMI

Lietotāju un klientu pieslēgtie pakalpojumi, to pieslēgšanas datums, pakalpojuma nosaukums un detalizētāka informāciju par parametriem.

VAAC_BILANCES

Informācija par lietotāju iztērētajiem skaitītājiem, to derīguma termiņu un par iztērētās naudas summu.

VAAC_SESIJAS

Klientu sesijas, kad tie ir logojušies iekšā un sesijas numurs.

VAAC_IESNIEGUMI

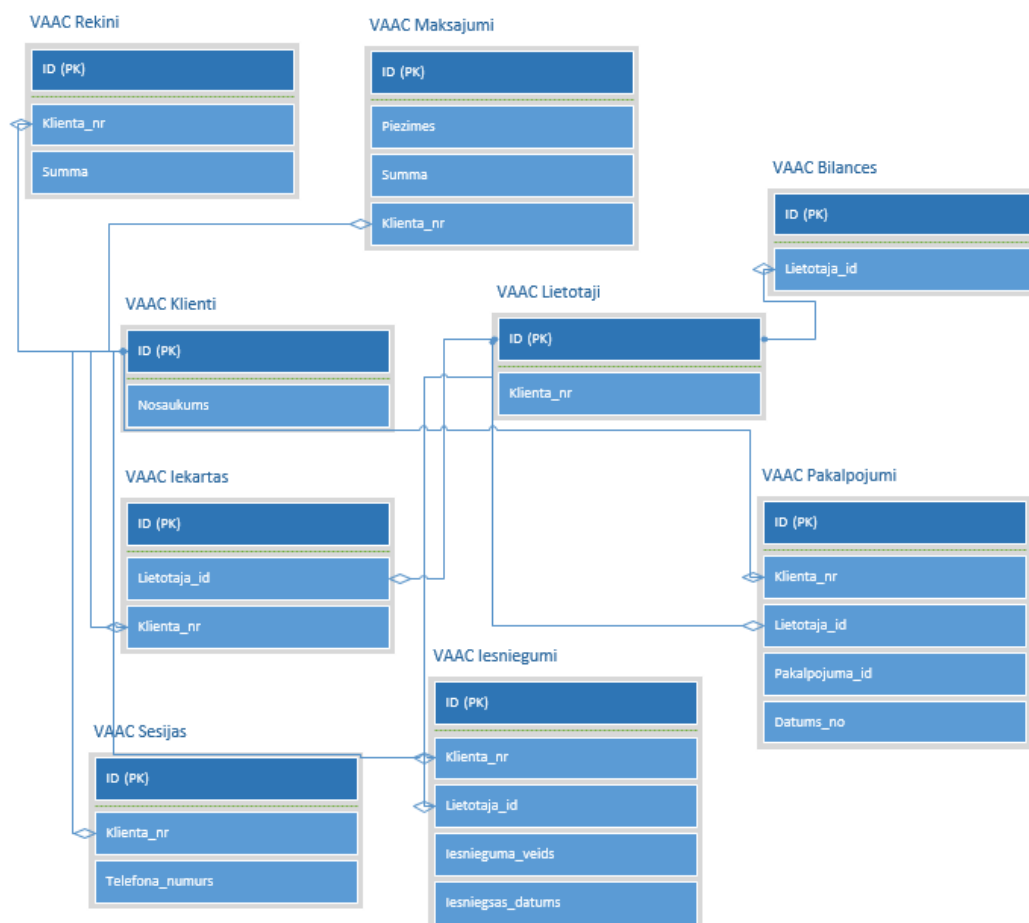
Klienta veiktie iesniegumi VAAC sistēmā par dažādām tēmām, piemēram, pakalpojumu pieslēgšana un atslēgšana, pakalpojumu plāna maiņas, lietotāju izveidošana, piedāvājumu izmantošana, lietotāja atslēgšana uz laiku.

VAAC_MAKSAJUMI

Informācija par klienta veiktajiem maksājumiem, to summa un veikšanas datums.

VAAC_IEKARTAS

Informācija par klientu un lietotāju iekārtām, summa, līguma termiņš, iekārtas modelis



Attēls 1.4. VAAC shēma

1.2.2. Datu iegūšana

Visi klientu dati tiek rēķināti vienreiz 24 stundu laikā, naktī, izmantojot Oracle sistēmas paralēlos procesus, kuri atlasa informāciju no atbilstošajām tabulām pārējās shēmās un saglabā tos atbilstošajās VAAC tabulās. Rezultātā tiek iegūti strukturēti un pārskatāmi dati, kas gan ir 0-24 stundas veci.

Atsevišķs sistēmas process ir paredzēts iesniegumu apstrādei, kas veic pakalpojumu pieslēgšanu un atslēgšanu, kā arī pakalpojumu maiņu lietotājiem un klientiem, automātiski. Pārējie klientu iesniegumi ir informatīvi klientu atbalsta daļas vajadzībām, tos sistēma apstrādāt nespēj.

1.3. Sistēmas trūkumi

Kā jau jebkurai sistēmai, trūkumi var traucēt jebkurai sistēmas lietošanā vai uzturēšanā iesaistītajai pusei. Šajā nodaļā tiks pastāstīts par to iedalījumu atkarībā no puses, kurai konkrētā problēma traucē visvairāk.

1.3.1. Biznesa un lietotāju trūkumi

1.3.1.1. Novecojuši dati

Lielākā problēma no lietotāja puses ir tas, ka dati vienmēr ir zināmā mērā veci, jo sistēmas process, kas šos datus rēķina un saglabā, tiek darbināts reizi dienā, kas nozīmē, ka nekādas izmaiņas lietotāja datos nebūs redzamas līdz nākamās dienas rītam, neatkarīgi no tā, vai ir mainījies kāds pakalpojums, parādījies jauns rēķins vai veikts maksājums. Tāda pati situācija ir arī ar lietotāja skaitītājiem un iztērētās naudas bilanci.

1.3.1.2. Iesniegumu apstrādes ilgums

Daļēji saistīts ar iepriekšējo punktu. Tā kā lietotājam nav iespējas redzēt izmaiņas savos pakalpojumos, pakalpojumu plānos līdz nākamās dienas sākumam, tad lietotājam rodas sajūta, ka nekas nav mainījies, līdz viņš to neredz VAAC sistēmā. Papildus tam, tie iesniegumi, kuri nav automatizēti, tiek apstrādāti daudz ilgākā laikā, jo tur nepieciešama operatora iesaiste.

1.3.1.3. Vēla rēķinu pieejamība

Esošajā sistēmā rēķins tiek izvadīts (izveidots PDF fails) un izsūtīts pa e-pastu, taču lietotājam tas nav pieejams līdz brīdim, kamēr visiem klientiem rēķins ir izveidots, izsūtīts un ir beidzies datu rēķināšanas process. Šeit rodas tā dīvainā situācija, ka lietotājs ir saņēmis rēķina apstiprinājumu, taču pašapkalpošanās sistēmā to neredz un caur turieni to apmaksāt vēl nevar.

1.3.1.4. Automatizācijas trūkums

Galvenokārt biznesa speciālistu problēma, jo tiek patērēti lieli cilvēkresursi, lai apstrādātu visus tos iesniegumus, kuriem nav izveidota automātiska apstrāde. Cilvēka kļūdas gadījumā

klients nav dabūjis ko vēlējies un iesniegums ir jāveido vēlreiz, šoreiz jau, iespējams, ar rēķina pārrēķinu.

1.3.2. IS uzturētāju problēmas

1.3.2.1. Aprūtināta datu izmaiņu vadība

Jebkuras izmaiņas atgriežamajos datos prasa iziet pilnam piegādes ciklam, sākot ar izstrādi, testēšanu un piegādāšanu produkcijā, kas laika ziņā aizņem vidēji 14 kalendārās dienas. Tas notiek neatkarīgi no tā, vai tiek pievienots viens jauns lauks vai 10 jauni lauki. Izmaiņas ir jāveic gan tabulu struktūrā, gan datu rēķināšanas sistēmas procesā, prasot nozīmīgu tehnisko un administratīvo resursu apjomu. Savukārt sistēmas morālās novecošanas dēļ, nav iespējams pašapkalpošanās tīmekļa vietnē piedāvāt vairākas jaunas lietas

1.3.2.2. Papildus drošības slāņa trūkums

Esošajā sistēmā ir tikai viens drošības slānis starpniekserverī. Visa sesijas pārraudzība notiek tur. Lietotājam pierakstoties VAAC sistēmā, tiek uzģenerēts sesijas identifikators, kurš tiek izmantots sasaistē ar lietotāja identifikatoru. Lai arī datu pieprasījumi no tīmekļa servera caur starpniekserveri nenotiek pa taisno, teorētiski, uzlaužot starpniekserveri, var būt iespējama situācija, ka dati nonāk pie nepareizā īpašnieka. Būtu daudz korektāk, ja drošības pēdējo valni nodrošinātu tā sistēma, no kuras dati tiek atlasīti.

1.3.2.3. Ienākošo datu kodējums

Visi lietotāja ievadītie dati, kas nonāk iesniegumu tabulā, ir ievadīti, izmantojot unikodu (Unicode), taču VAAC sistēmas būvēšanas laikā konkrētā Oracle versija šo kodējumu neatbalstīja, tāpēc starpniekserverī nākas pārkodēt šo informāciju uz heksadecimālo kodējumu, kas pēc tam ir jāatkodē atpakaļ uz unikodu, lai tā būtu cilvēkam lasāma. Tādējādi, jebkura testēšana, tikai apskatoties uz sistēmā saglabātajiem lietotāja ievadītajiem datiem, nav iespējama bez datu atkodēšanas.

1.3.2.4. Datu dublēšanās

Esošajā risinājumā tie paši dati, kas jau glabājas centrālajās shēmās, tiek būtībā dublēti un pārrakstīti uz VAAC shēmu, tādējādi radot nevajadzīgi aizņemtu vietu. Šī gan ir problēma

vairāk no loģikas, nekā no atmiņas viedokļa, jo pārdesmit papildus aizņemti megabaiti neko sevišķu sistēmai nenodara.

2. JAUNĀS SISTĒMAS PRASĪBAS

Kā jau jebkuras sistēmas uzlabošanā, nepietiek tikai ar esošo trūkumu mīkstināšanu, tāpēc sistēmai arī tika uzdotas papildus prasības no biznesa speciālistu puses. Prasības pēc tipa tika dalītas funkcionālajās un nefunkcionālajās prasībās. Ir uzskaitīti arī galvenie izaicinājumi, kas rodas šo prasību izpildē un problēmu risināšanā

2.1. Funkcionālās prasības

2.1.1. *Provizoriskais rēķins*

Bieži gadās situācijas, kad klients iztērē lielu naudas summu, pats to neapzinoties, un tad, saņemot rēķinu, rodas pārsteigums. Izveidojot sadaļu, kur klients var redzēt tiešsaistē aprēķinātu rēķinu, kāds viņam pēc pašreizējās situācijas būs mēneša sākuma, ir veids, kā klientiem šos pārsteigumus samazināt. Veidojot rēķinu, mēneša maksas tiek rēķinātas pa dienām – ja lietotājs ir lietojis konkrētu pakalpojumu plānu pusmēnesi, un citu plānu otru pusmēnesi, tad mēneša beigās cena tiek aprēķināta proporcionāli katra plāna izmantoto dienu skaitam. Tāda pati situācija ir arī ar pakalpojumiem. Maksu grupas iedalās

- Vietējā satiksme
- Ārzemju satiksme
- Paaugstinātas maksas pakalpojumi
- Pakalpojumu plāna maksas
- Pakalpojumu maksas
- Citi pakalpojumi

Pēc aprēķināšanas, klientam neko vairs nemainot, aprēķinātajai summai vajadzētu arī būt redzamai rēķinā.

Papildus tam, ka tiek rēķināts tekošā mēneša provizoriskais rēķins, šai informācijai jābūt arī redzamai nākamā mēneša sākumā, līdz rēķina izveidošanas brīdim.

2.1.2. *Klienta iesniegumu apstrādes automatizācija*

Nav pieļaujama situācija, ka uzņēmuma darbiniekam ir jāapstrādā klienta iesūtītais iesniegums manuāli, jo tas prasa laiku un resursus. Saņemto iesniegumu sistēmai ir jāapstrādā automatiski un jādod lietotājam redzēt rezultātu uzreiz pēc tā izpildes, vai tā būtu lietotāja atslēgšana uz laiku, lietotāja anulēšana vai pakalpojumu plāna maiņa.

2.1.3. Lietotāja anulēšana

Līdz šim nav iespējas VAAC sistēmā klientam anulēt kādu no saviem lietotājiem, taču klientu ērtības labad būtu nepieciešams, lai to būtu iespējams izdarīt, tādējādi izvairoties no lieka gājiena uz klientu centru, tad ieguvējs ir gan klients, kurš ir sev aiztaupījis lieku braukāšanu, gan klientu centrs, kuram samazinās rindas, jo jāapkalpo mazāk cilvēku.

2.1.4. Masveida pakalpojumu pieslēgšana

Sevišķi aktuāla prasība tieši lielajiem un biznesa klientiem, kuriem, tā vietā, lai zvanītu konsultantam un lūgtu izpildīt katru no prasībām ar roku, jābūt iespējai pieslēgt/atslēgt pakalpojumus pašam, vai nu visiem saviem lietotājiem, vai vairākiem vienlaicīgi, kā arī jābūt iespējai izdarīt to pašu ar pakalpojumu plāniem.

2.1.5. Ilgtermiņa lietotāja paradumu statistika

Lietotājam jāspēj sekot līdzi saviem lietošanas paradumiem, lai saprastu, kuri no pakalpojumiem vai pakalpojumu plāniem ir mazāk vai vairāk piemēroti, un jāspēj reaģēt uz savu paradumu izmaiņām, lai izvairītos no nevajadzīga rēķina pieauguma, tāpēc nepieciešams izveidot sadaļu, kur klients var redzēt pēdējo 12 mēnešu pārskatu par visiem viņa pakļautībā esošajiem lietotājiem, kur ir redzama to aktivitāte dažādos kritērijos tā mēneša griezumā, piemēram

1. Vietējais patēriņš
 - Zvanu skaits
 - Zvanu ilgums
 - SMS skaits
 - Datu patēriņš pa tīkla grupām
2. Ārzemju patēriņš
 - Zvanu skaits ārzemēs
 - Zvanu ilgums ārzemēs
 - SMS skaits ārzemēs
 - Datu patēriņš ārzemēs

Izejot no šīs informācijas, klients var pieņemt lēmumu palielināt vai samazināt sev pieejamos limitus zvaniem vai datu patēriņam, tādējādi padarot iztērēto naudu efektīvāku.

2.1.6. Top10 zvanītie numuri

Lietotāja līmenī nepieciešams atrādīt top10 numurus mēneša ietvaros, uz kuriem ir ticis zvanīts, kā arī sarēķināt kopējo zvanu ilgumu uz šiem numuriem, tādējādi parādot lietotājam statistiku ar kuriem pieslēguma numuriem ir ticis visvairāk komunicēts.

2.1.7. Jauna lietotāja izveidošana

Nepieciešama funkcionalitāte, kas ļauj klientam izveidot jaunu lietotāju, ar nosacījumu, ka pie viņa atrodas darboties spējīga sim karte. Izvēloties no numuru saraksta jaunu numuru un ievadot sim kartes numuru, kā arī pakalpojumu plānu, sistēmai jāspēj uzģenerēt jaunu aktīvu lietotāju, kurš ir pieslēgts var veikt tīkla notikumus.

2.2. Nefunkcionālās prasības

2.2.1. Reāllaika datu iegūšana

Jaunajā sistēmā jālabo viens no lielākajiem vecā VAAC trūkumiem, novecojuši dati. Pēc iespējas visiem savāktajiem datiem klienta pieteikšanās brīdī sistēmā ir jābūt tik svaigiem, cik tie sistēmā atrodas, piemēram, nav pieļaujama situācija, ka pirms 10 minūtēm pieslēgtais pakalpojums klientam nav pakalpojumu sadaļā redzams, vai ka veiktais zvans, kuram ir beigusies tarifikācija pirms 15 minūtēm, nav ticis pieskaitīts pie kopējā zvanu apjoma. Tā arī rēķiniem jābūt redzamiem un apmaksājamiem VAAC sistēmā tiklīdz tie ir izveidoti un izvadīti (nosūtīti klientam uz norādīto epastu). Tāpat arī visām šīm izmaiņām ir jāatspoguļojas provizoriskā rēķina skatā – lietotājam pieslēdzot pakalpojumu, uzreiz ir jāredz, kā tas ietekmēs viņa rēķinu mēneša beigās. Izņēmumi ir situācijas, kad centrālajā datu bāzē dati vēl nav redzami, piemēram tīkla notikums nav pabeidzis tarifikāciju.

2.2.2. Datu iegūšanas ātrums piesakoties

Klientam piesakoties sistēmā, nepieciešams panākt, ka dati tiek savākti ne vairāk kā 5 sekunžu laikā, no cilvēka viedokļa tas nozīmē pirmās lapas ieraudzīšanu ne vēlāk kā 10 sekunžu laikā ar visu datu pārnesi, pēcapstrādes un lapas ģenerēšanu.

2.2.3. Iesniegumu apstrādes ātrums

Vecajā sistēmā iesniegumu apstrādes ātrums variē no 2-3 sekundēm, pakalpojumu pieslēgšanas/atslēgšanas gadījumos, kā arī pakalpojumu plānu maiņas gadījumos, līdz pat 2-3 dienām, gadījumos, kad iesnieguma izpildē nepieciešams iesaistīties darbiniekam. Jaunajā VAAC nepieciešams iesnieguma apstrādei patērēt ne vairāk kā 1 sekundi, lai pīķa stundās, kad tiek iesniegti 20-30 iesniegumi sekundē, nenotiktu manāmas aiztures klientu pusē.

3. IZAICINĀJUMI PRASĪBU IZPILDĒ

3.1. Esošā projektējuma uzlabošana

3.1.1. Uzlabot datu ieguves ātrumu

Atlasot datus no plakanām tabulām, iespējams panākt ievērojamu datu ieguves ātrumu, neskatoties uz klienta lietotāju skaitu. Datu savākšanas ātrums sevišķi nemainās atkarībā no tā, vai pierakstīšanos veic klients ar vienu lietotāju vai klients ar 1000 lietotājiem, jo viena lietotāja informācijas ieguves ātrums ir niecīgs, savukārt lielam klientam tas pārvēršas vien par nelielu.

Esošajā sistēmā tas ir panākts ar to, ka visi dati ir izrēķināti jau pirms laika, un pieteikšanās brīdī tiek tikai veikts selekts plakanā tabulā, taču jaunveidojamajā sistēmā šādas iespējas nav, visi bilances un provizoriskā rēķina dati ir jāizrēķina pēc pieprasījuma, kā arī iekārtu līgumu datu vākšana ir salīdzinoši lēns process, jo tur ietilpst gan iekārtas, gan paša līguma, gan to datu iegūšana.

Mazam lietotāju skaitam visu datu atlase nav liela problēma, taču klientiem ar daudz un ļoti daudz lietotājiem tā sāk kļūt par lielu problēmu.

3.1.2. Saglabāt resursu patēriņu esošajā līmenī

Rēķinot datus vienreiz dienā, lai arī apjomīgu daudzumu, ir iespējams smagākos resursu tēriņus veikt tieši naktī, kad sistēmas noslodze kopumā ir stipri mazāka, tādējādi izlīdzinot kopējo dienas resursu līkni. Tieši bilances un skaitītāju rēķināšana, kas aizņem vislielāko resursu apjomu, ja tiktu veikta ļoti regulāri, patērētu milzīgus sistēmas resursus.

Ieviešot jauno sistēmu, nepieciešams panākt, ka resursu patēriņš nepieaug, lai gan datus nepieciešams iegūt reāllaikā.

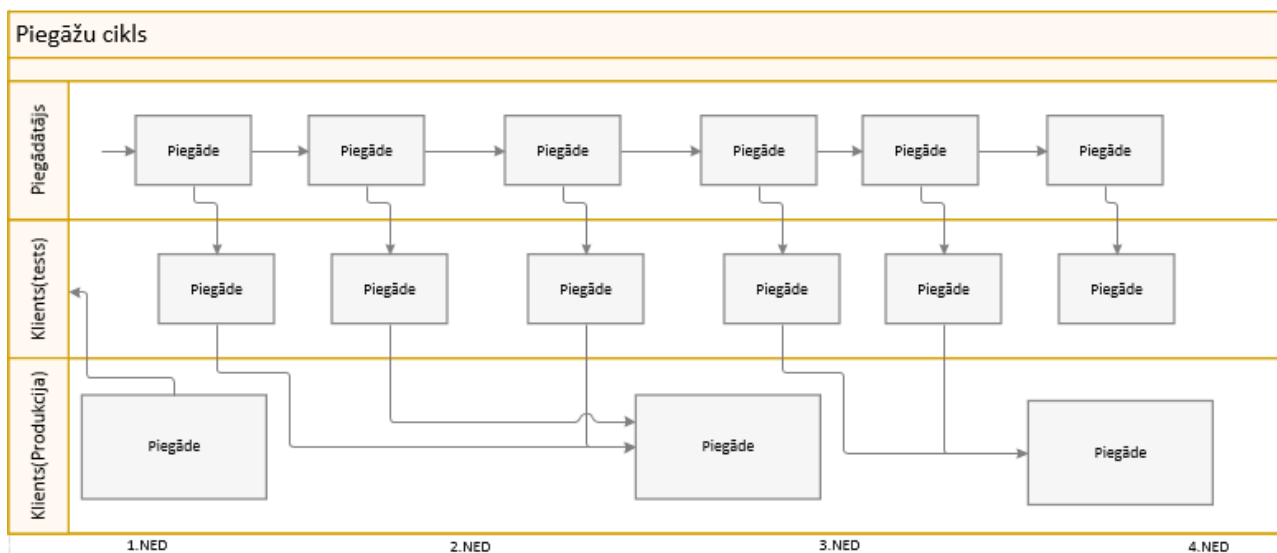
3.1.3. Uzlabot drošības līmeni

Situācijā, kad drošības pārbaudes notiek tikai vienā vietā, ir iespējams labi nodrošināties pret to, ka starpniekservera Oracle lietotājs netiek klāt pie visās pārējās shēmās esošajiem datiem, taču, ja šis Oracle lietotājs veic tiešos SQL vaicājumus pie VAAC sistēmas shēmas, tad tajā brīdī nenotiek nekādas pārbaudes datu bāzes pusē, vai prasītājam vispār ir tiesības iegūt datus par konkrēto lietotāju.

3.2. Jaunievedumu ieviešanas izaicinājumi

3.2.1. Paātrināt izmaiņu ieviešanu

Esošajā sistēmā, tāpat kā visās citās shēmās, ieviest izmaiņas produkcijas vidē ir apjomīgs un laikietilpīgs process, pat izdarot pavisam nelielas izmaiņas. Konkrēti VAAC sistēmas gadījumā, lai tīmekļa serverim nogādātu vienu papildus datu lauku, nepieciešams pārrakstīt datu vākšanas sistēmas procesu, kas nozīmē koda izmaiņas, kā arī pārbūvēt tabulu, kur šie dati glabājas. Ļoti liels izaicinājums ir atrast veidu, kā no tā visa izvairīties, lai maznozīmīgas izmaiņas datu atlasē būtu iespējams izdarīt ātrāk, neiesaistot lielus cilvēkresursus, jo avāriju gadījumā piegādes laiku var saīsināt līdz pāris stundām.



Attēls 3.1. Piegāžu cikls uzņēmumā

Kā redzams attēlā, piegādes no piegādātāja tiek saņemtas ar regulāru intervālu, kā arī tiek liktas testa datubāzē gandrīz uzreiz pēc saņemšanas, taču ir lielāka aizture pirms konkrētā piegāde nonāk produkcijas datubāzē. Galvenie iemesli ir:

- Rēķinu veidošana

Katra mēneša pirmā dekāde tiek atvēlēta iepriekšējā perioda slēgšanai, kas nozīmē konfigurācijas pārbaudes un visu klientu rēķinu veidošanu. Šajā laikā piegādes produkcijā netiek liktas, lai neaizkavētu vai neiztraucētu šo procesu.

- Testēšanas resursu pieejamība

Tā kā piegādes no piegādātāja tiek saņemtas vidēji divas reizes nedēļā, uzņēmumam būtu nepieciešami ievērojami resursi tikai uz testētāju komandu, lai piegāžu uzlikšanu produkcijā

būtu iespējams ievērojami paātrināt. Papildus tam, zinot, ka pirmajā mēneša dekadē no tā nebūtu jēgas, tas būtu nevajadzīgs uzņēmuma resursu patēriņš.

3.2.2. Rēķinu pieejamība uzreiz pēc izveidošanas

Iespējams, kā jebkurā lielā uzņēmumā ar daudz klientiem, ilgu laiku aizņem rēķinu veidošana. Esošajā sistēmā rēķins lejupielādējamā formātā ir pieejams salīdzinoši vēlu, tikai tad, kad tas ir sagatavots pilnīgi visiem klientiem. Galvenais iemesls tam ir tāds, ka rēķina izveidošana vēl nav vienīgā pazīme, ka tam ir jābūt klientam pieejamam. Kad rēķini masveidā tiek izveidoti, tie tiek kārtīgi pārbaudīti vai nav ieviesušās kādas datu kļūdas un tikai tad, kad tas ir izdarīts, tam būtu jānonāk pie klienta, tas tiek darīts ar rēķinu “izvadīšanu”. Papildus problēmas rodas tajā brīdī, kad maksātājs ir izvēlējis rēķinam netipisku šablonu, piemēram, nevis PDF, bet gan teksta failu, kas nozīmē, ka sistēmai ir jānāk atrast ne tikai īsto rēķinu, bet arī īsto datnes veidu, kas ir diezgan grūti izdarāms, ja rēķins, tā dati un pati lejupielādējamā datne ir jāiegūst tiešsaistē, negaidot rēķinu procesa beigas.

3.2.3. Klientu dalījums pēc lietotāju skaita/atrādāmās informācijas

Esošajā sistēmā nav nekāda dalījuma starp lieliem/maziem klientiem vai privātpersonām/juridiskām personām. Problēma rodas brīdī, kad rādāmo informāciju nepieciešams nodalīt, piemēram, privātpersonām drīzāk interesēs informācija par pakalpojumiem, pakalpojumu plāniem un citām maksām ar PVN, savukārt juridiskām personām visas šīs summas būtu vairāk nepieciešams atrādīt bez PVN. Tāpat arī juridiskām personām būtu lielāka nepieciešamība redzēt ne tikai pēdējos pāris rēķinus, bet vismaz gadu vecus rēķinus, grāmatvedības nepieciešamībām. Klientam izveidotās pilnvaras privātpersonām nav aktuālas, tās interesē tikai korporatīvos klientus, lai tie var redzēt, kam ir šobrīd aktīva pilnvara redzēt klienta datus.

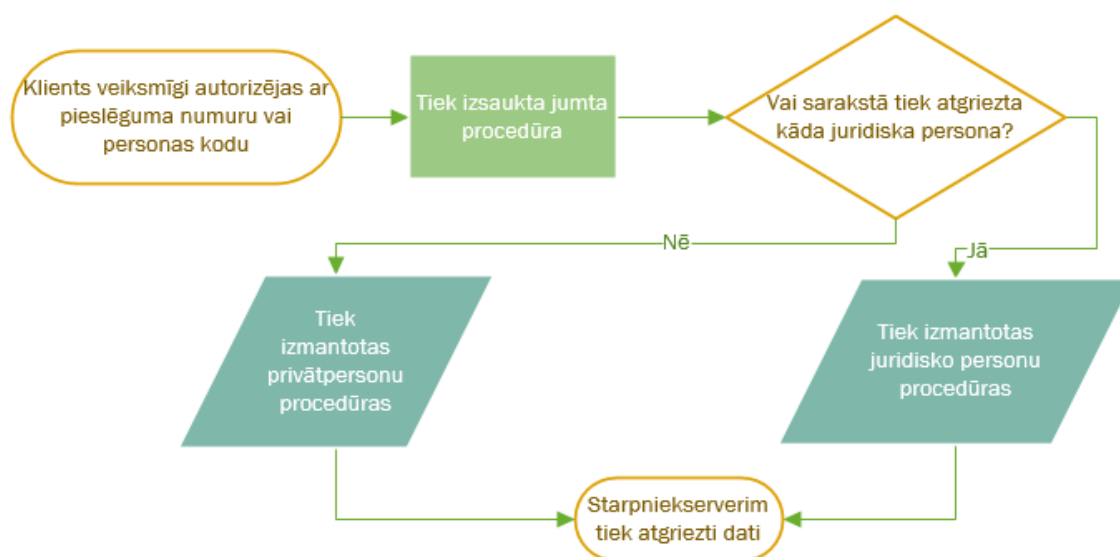
Liela starpība atrādāmajā informācijā arī rodas tad, kad klienta lietotāju skaits ir nevis 5 vai 10, bet gan 10000 un vairāk. Klientam vairs neinteresē atsevišķu lietotāju bilances tik ļoti, cik svarīga ir kopsumma. Esošajā sistēmā tas nesagādā problēmas tikai dēļ tā, ka dati netiek vākti reālajā laikā, bet gan pieteikšanās brīdī tiek iegūti iepriekš izrēķinātie dati no plakanām tabulām. Tā kā vienīgais veids kā panākt nepieciešamo ātrdarbību tik atšķirīgām lietotāju vajadzībām ir sistēmas nodalīt, kas nozīmē veidot daļu no datiem, kas tiek iegūta visiem klientiem vienādi, savukārt otru daļu savākt atkarībā no klienta tipa.

Galvenokārt, šajā apakšpunktā aprakstīto iemeslu dēļ tika pieņemts lēmums ieviest pašapkalpošanās sistēmu divās kārtās, nodalot privātpersonu un juridisko personu sistēmas

4. PRIVĀTPERSONU SISTĒMAS PROJEKTĒJUMS

Projektēšanas laikā tika pieņemts lēmums izmantot PL/SQL procedūras, kuras starpniekservierim izsaucot, XML formātā tiek atgriezti strukturēti dati, kuri tiek apstrādāti un pārveidoti par masīvu, ar kuru tālāk strādā PHP. Papildus tam, tika nodalītas privātpersonu, jeb, salīdzinošu mazu klientu procedūras, no juridisku personu, jeb, lielo klientu procedūrām. Šajā nodaļā skaidrots, kāpēc tika pieņemti tieši tādi lēmumi un kādu iespaidu tas atstāja uz sistēmu, kā arī kā tika apmierinātas darījumprasības.

4.1. Datu ieguves shēma



Attēls 4.1. Lēmuma pieņemšana par procedūru kopas izmantošanu

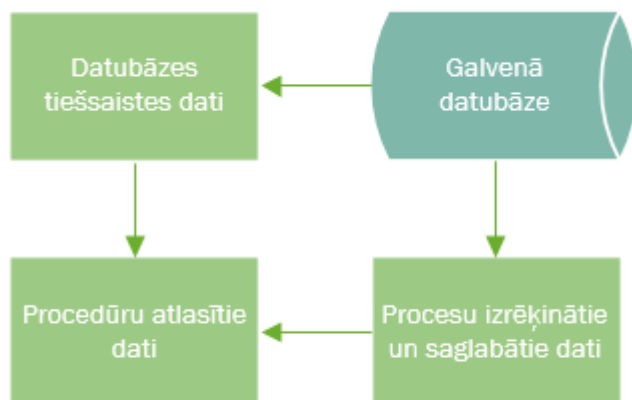
Globālā datu savākšanas shēma ir šāda. Visiem klientiem pieteikšanās lapa ir vienāda, taču brīdī, kad klients autorizējas, tiek izsaukta jumta procedūra, kas atgriež visus klientus, kas konkrētajam pieslēguma numuram vai personas koda īpašniekam ir pieejami, izvēloties klientu ar kuru strādāt, notiek viena vai otra procedūru kopa.

Jumta procedūras ieejošie parametri ir – pieslēguma numurs/personas kods un unikāls sesijas id, kas tiek uzģenerēts starpniekservierī pēc veiksmīgas pieteikšanās.

Visu procedūru atgrieztais rezultāts tiek ierakstīts NCLOB datu tipa mainīgajos XML formātā.

Pašu procedūru saturs ir SQL vaicājumu sajaukums ar PL/SQL kodu, kas nepieciešamības gadījumā procedūras ietvaros veic aprēķinos lietām, kuras nav pa taisno atlasāmas no tabulām.

Pašu procedūras datu kopums ir kombinācija no svaigiem datiem un asinhrono procesu izrēķinātiem vai iegūtiem datiem.



Attēls 4.2. Datu ieguves avoti

4.2. Datu ieguve (procedūras)

Speciāli šī veida klientiem ir paredzēta viena apvienojošā procedūra, kurā kā ieejošie parametri tiek padoti klienta id un jumta procedūrā uzģenerētais sesijas id. Veiksmīgu pārbaužu rezultātu gadījumā, tiek izsauktas 3 procedūras, kurās atgrieztie dati ir loģiski sagrupēti pēc to pielietojuma

- Klienta procedūra

Tiek atgriezti visi dati par klientu – vārds, uzvārds, norādītās pasta un e-pasta adreses, klienta pakalpojumi un to parametri, klienta iekārtas, kredītkontroles statuss, kā arī klienta rēķini un veiktie maksājumi.

- Lietotāju procedūra

Tiek atgriezti visi dati par klienta lietotājiem – vārds, uzvārds, pieslēguma numurs, pakalpojumi un to parametri, pakalpojumu plāns, saraksts ar pakalpojumiem un pakalpojumu plāniem, kas ir pieslēdzami kā arī lietotājam piesaistītās iekārtas un informācija par tām.

- Bilances procedūra

Tiek atgriezti visi dati par klienta lietotāju bilancēm – provizoriskais lietotāju rēķins, visa informācija par izlietoto tīkla satiksmi, to ilgums, apjoms un naudas summa, kā arī atlikums no lietotājam pieslēgtajiem datu/zvanu/sms skaitītājiem.

Šo 3 procedūru atgrieztais rezultāts ir viss, kas klientam nepieciešams no centrālās datubāzes, lai būtu iespējams strādāt ar VAAC sistēmu.

4.3. Asinhronie procesi

4.3.1. Iesniegumu apstrādes process

Jaunā projektējuma ietvaros tika pilnīgi pārstrādāts iesniegumu apstrādes process. Jaunais process ir spējīgs visus iesniegumus apstrādāt automātiski, neiesaistoties darbiniekiem. Procesā ātrdarbība tika uzlabota, pārtaisot to par mūžīgo procesu, kas visu laiku gaida jaunas komandas, tādējādi saņemot iesniegumu tas var reaģēt uzreiz, tādējādi uzlabojot vidējo iesnieguma apstrādes laiku līdz apmēram 1 sekunde.

Papildus tika vienkāršotas pakalpojumu plānu maiņas, ja līdz šim katram jaunajam pakalpojumu plānam vajadzēja veidot jaunu iesnieguma veidu, tad tagad šis pakalpojuma plāna identifikators tiek padots kā parametrs iesniegumam, tādējādi vienkāršojot konfigurācijas uzturēšanu.

Tāpat tika paredzēts atbalsts masveida pakalpojumu pieslēgšanai, atslēgšanai, pārveidojot iesniegumu tabulu par divām tabulām, no kura pirmā tiek izmantota kā interfeisa tabula, kurā iekrīt katrs individuālais ieraksts, savukārt, atkarībā no tajā prasītajām darbībām, tiek izveidots šī iesnieguma apakšieraksts, kurā ir redzama konkrētā komanda, kas procesam jāveic. Piemēram, ja parasts lietotājs vēlas sev nomainīt pieslēgt pakalpojumu, tad interfeisa tabulā būs redzams viens ieraksts un darbību tabulā arī būs redzams viens ieraksts, taču, ja interfeisa tabulā būs ieraksts, kurā būs norādīti, teiksim, 5 numuri atdalīti ar komatiem, tad, sasaistīti ar virsējo ierakstu, tiks izveidoti 5 ieraksti darbību tabulā, no kuriem tiks izpildīts katrs atsevišķi.

4.3.2. Lietotāju paradumu statistikas aprēķina process

Reizi mēnesī par katru lietotāju aprēķina tā pēdējā mēneša paradumu statistiku un saglabā tabulā. Tā kā tarifikācija tīkla notikumiem notiek tikai konkrētā mēneša ietvaros, kam beidzoties periods tiek slēgts, ir iespējams izvairīties no visu 12 mēnešu statistikas pārrēķina katru mēnesi. Lai arī apjomīgs un ilgs process, tas tiek darbināts tikai reizi mēnesī, tāpēc globālais resursu patēriņš ir zems.

4.3.3. *Top10 zvanīto numuru process*

Reizi mēnesī, pēc periodu slēgšanas, tiek darbināts process, kas katra lietotāja ietvaros sagrupē visus tā zvanus uz katru no numuriem un atgriež 10 numurus, uz kuriem ir zvanīts visvairāk un kopējo šo zvanu ilgumu par pēdējiem 12 mēnešiem, savukārt šo informāciju saglabā tabulā, no kurienes tā tiek padota kopā ar bilances informāciju.

4.4. Datu strukturēšana

Tā kā drošības aspektu dēļ autors pieņēma lēmumu strādāt ar procedūrām, tiešo SQL vaicājumu vietā, parādījās problēma par to, kur rakstīt datus, jo tabulas struktūrā šos datus ierakstīt nav iespējams. Bija nepieciešams izdomāt kādā formātā šos datus mainīgajā ierakstīt, tā lai tie būtu viegli lasāmi un viegli apstrādājami automātiskai sistēmai, pārveidojot to par masīvu. Izvēle bija par labu XML formātam, kas nodrošina abus šos kritērijus, taču xml formātā šie dati aizņem pārāk daudz vietas, lai to ievietotu nvarchar mainīgajā, tāpēc tika nolemts izmantot NCLOB (National Character Large Object)[9] mainīgo, lai būtu iespējams gan ierakstīt liela apjoma datus, gan arī izmantot latviešu garumzīmēm un mīkstinājumzīmēm nepieciešamo UTF-8 kodējumu.

Neliela daļa no (lietotāja procedūras) atgriežamā mainīgā vērtības satura, ar neīstiem datiem:

```
<MSISDN>29999999</MSISDN>  
<CTR_ID>524563</CTR_ID>  
<SSR_ID>23453637</SSR_ID>  
<FIRST_NAME>Vārds</FIRST_NAME>  
<LAST_NAME>Uzvārds</LAST_NAME>  
<SSU_IC>ACTIVE</SSU_IC>  
<SPN_ID>65345</SPN_ID>
```

4.5. Datu drošība

Galvenais iemesls strādāt ar procedūrām bija tieši drošība. Tās līmenis tika uzlabots izmantojot sesijas identifikatoru, kurš gan bija arī iepriekšējā sistēmā, taču tur tas tika pārbaudīts tikai starpniekserverī, projektējumā paredzētais variants paredz, ka klientam pierakstoties sistēmā tiek izveidots sesijas identifikators, kurš ir derīgs vienu stundu, sesija tiek saistīta ar visiem klientiem, ar kuriem cilvēkam ir tiesības strādāt un katrs procedūras izsaukums VAAC shēmā sevī iekļauj sesijas identifikatoru, kurš tiek pārbaudīts pret saglabāto,

ja saglabātais nesakrīt ar padoto, tad procedūras atbildē tiek ierakstīts, ka sesijas īpašniekam nav tiesības skatīties datus.

Šāds formāts palīdz pilnīgi izsargāties pret cilvēciskām kļūdām, kas varētu rasties starpniekserversa programmētāju dēļ, taču tagad datu atgriešana ir ļoti cieši sasaistīta ar konkrētās sesijas identifikatoru.

4.6. Datu konfigurācija

Viena no apjomīgākajām VAAC sistēmas daļām ir konfigurācija. Konfigurācija tiek izmantota lai piedāvātu katram klientam un lietotājam tikai tos pakalpojumus un pakalpojumu plānus, kas tam ir pieslēdzams vai atslēdzams, izslēdzot tehniskos pakalpojumus, kuri lietotājam nav jāredz, taču tāpat viņam ir pieslēgti. Tāpat konfigurācija tiek izmantota arī skaitītāju iegūšanai un sasaistei ar noteiktu pakalpojumu, kā arī provizoriskā rēķina rindu izveidošanai un iesniegumu apstrādei tiem iesniegumu veidiem, kas satur sevī komplicētāku informāciju par parastu pakalpojumu pieslēgšanu/atslēgšanu.

VAAC_PAKALPOJUMI

Konfigurācijas tabula satur visu to pakalpojumu sarakstu, kuriem ir kaut kāda saistība ar VAAC sistēmu. Tabulas lauki ir:

- PAKALPOJUMA_ID – lai atpazītu pakalpojumu
- IR_REDZAMS – J/N vērtība pasaka to, vai pakalpojums ir iekļaujams lietotāja pakalpojumu sarakstā un vai tas lietotājam ir jāredz
- IR_PIESLEDZAMS – J/N vērtība pasaka to, vai pakalpojumu lietotājam ir ļauts pieslēgt
- IR_ATSLEDZAMS – J/N vērtība pasaka to, vai pakalpojumu lietotājam ir ļauts atslēgt
- IR_MAINAMS – J/N vērtība pasaka to vai lietotājam ir ļauts pakalpojumam mainīt parametrus

VAAC_PAKALPOJUMU_PLANU_GRUPAS

Konfigurācijas tabula satur visus pakalpojumu plānus, kādi lietotājam varētu būt pieslēgti, kur katrs no tiem ir ievietots pakalpojumu plānu grupā, kur lietotājam ir iespējas mainīt plānus tikai grupas ietvaros, lai lietotājs nevarētu, piemēram, nomainīt savu fiksētā telefona plānu uz mājas interneta plānu, jo viņam var nebūt pieejama atbilstoša ierīce. Tāpat katrai no grupām ir arī papildus grupa ar galotni _OLD, kur šīs grupas pakalpojumu plānu

izmantotājiem, kuri lielākoties ir arhaiski plāni, būtu iespēja pārslēgties uz jaunā veida šīs grupas plānu, taču virzienā atpakaļ to izdarīt nebūs iespējams. Tabulas lauki ir:

- PLANA_ID
- PLANA_NOSAUKUMS
- GRUPAS_NOSAUKUMS

VAAC_SKAITITAJI

Satur visus skaitītājus, kādi lietotājam ir parādāmi VAAC sistēmā. Tabulas lauki ir:

- SKAITITAJA_KODS – atpazīstam skaitītāju
- PAKALPOJUMA_ID – atpazīstam pakalpojumu, kam šis skaitītājs tiek pievienots
- LIETOTAJA_GRUPA – atpazīstam lietotāja grupu, lai skaitītāju piedāvātu tikai īstajai lietotāja grupai, jo dažām lietotāju grupām var būt redzami dažādi skaitītāji.

VAAC_IESNIEGUMU_VEIDI

Ja normālā gadījumā pakalpojumu pieslēgšana notiek pēc principa ‘Pieslēgt’ un pakalpojuma identifikators, kas vienmēr ir pozitīvs skaitlis, tad pārējie iesniegumu veidi tiek apstrādāti pēc principa, ka katrs negatīvais skaitlis apzīmē savādāku iesnieguma veidu, kuri ir nedefinēti šajā tabulā. Piemēram, iesnieguma veids “pakalpojumu plāna maiņa” ir apzīmēts ar -13 identifikatoru, un ja iesniegums ar šādu identifikatoru parādās ienākošo iesnieguma pieprasījumu tabulā, tad pēc tā iespējams to atpazīt un iesniegumu izpildīt.

Tā kā nav vajadzības konfigurāciju sarežģīt un veidot šādu iesnieguma veidu maiņai uz visiem dažādajiem plāniem, pats plāna identifikators tiek padots iesnieguma parametros. Tabulas galvenie lauki ir:

- ID
- IESNIEGUMU_VEIDA_NOSAUKUMS
- IZPILDES_TIESĪBAS

Izpildes tiesības kolonnas vērtība nosaka to, vai konkrēto iesniegumu drīkst izpildīt tikai klienta galvenais numurs vai arī negalvenais, pārbaudīts tas tiek pēc iesnieguma saņemšanas, lai būtu dubultā pārbaude gan validējot lietotāja iesūtītos datus, gan pēc apstrādē.

VAAC_IESNIEGUMU_PARAMETRI

Šajā tabulā ir nedefinēts kuriem “negatīvajiem iesniegumiem” atbilst kuri parametru tipi. Piemēram, jau minētajam iesnieguma veidam ‘-13’, ir tikai viens parametrs, kurš pasaka uz kādu pakalpojumu plānu iesniegumu procesam ir jāveic maiņa konkrētajam lietotājam. Citiem iesniegumu veidiem, piemēram, “klienta informācijas maiņa” ir paredzēti 2 parametri:

EPASTA_ADRESE, kurā tiek norādīta e-pasta adrese, uz kuru centrālajā datubāzē ir jāveic maiņa

KLIENTA_ADRESE, kurā klients norāda savu faktisko adresi, rēķinu vai sūtījumu saņemšanai.

Tādējādi ir iespējams sasaistīt konkrēto iesnieguma veidu ar parametriem, kurus procesam ir jāizmanto iesnieguma izpildē. Tabulas lauki ir:

- ID
- IESNIEGUMU_VEIDA_ID
- PARAMETRA_NOSAUKUMS
- DATU_TIPS
- VAI_OBLIGATS

5. DINAMISKU RISINĀJUMU IEVIEŠANA

Šajā nodaļā tiks pastāstīts par galvenajiem argumentiem par labu dinamiskiem risinājumiem, kā arī vietas, kur konkrēti tie tiks izmantoti.

5.1. Atgriežamo datu izmaiņu vadība

5.1.1. *Privātpersonu datu atlasēs shēmas trūkumi*

Ieviešot testa režīmā privātpersonu sistēmu, tika atklāts būtisks trūkums, kas uzņēmumā vēl nebija risināts. Tik vien, kā pievienot jaunu lauku, kas tiek atrādīts VAAC sistēmā, ir nepieciešams iesaistīt vairākus IT darbiniekus un laika ziņā, no lēmuma pieņemšanas brīža, līdz ieviešanai, aizņem vismaz 2 nedēļas[3.2.1.]. Galvenā problēma bija tā, ka visi datu atlasēs pieprasījumi bija ierakstīti procedūras kodā, kur izmaiņas veikt ir lēni, lai arī izmaiņas būtu šajā gadījumā jāveic tikai pašos datu atlasīšanas pieprasījumos. Palielinot pie šādas shēmas, sistēma būtu diezgan statiska, un steidzamas izmaiņas ieviest prasītu ievērojamus administratīvos resursus. Redzot šos trūkumus, kopā ar kolēģiem Gunti Vaivodu un Jāni Kalniņu tika izdomāts risinājums vaicājumu izmaiņu uzlabošanai.

5.1.2. *Dinamiskā datu vaicājumu izmaiņu vadība*

Idejas ietvaros tika pieņemts lēmums pēc iespējas nodalīt datu atlasēs vaicājumus, jeb SQL vaicājumus no PL/SQL procedūru koda.

Kas līdzīgs ir darīts šajā[10] foruma rakstā, lai gan koda piemērā netika aprakstīti veiksmīgi piemēri, kā šos dinamiskos vaicājumus papildināt ar mainīgajiem parametriem, bez kuriem pašapkalpošanās sistēmas vajadzībām, iztikt nevarēja. Nekur citur tieši šāds risinājums netika manīts. Tika apkopoti visi procedūrās izmantojamie SQL vaicājumi un katram piešķirts unikāls kods. Rezultātā iegūta dinamiska vaicājumu tabula, kur vaicājums ir sasaistīts ar unikālo kodu, kas tiek lietots procedūrā. Procedūra griežas pie vaicājumu tabulas un iegūst aktuālo vaicājuma tekstu, kura atgriežamie lauki tiek integrēti procedūras izejas parametrā XML formā.

Piemērs:

Lietotāja procedūra

vaicājuma kods: USER_SELECT

vaicājuma teksts: *select first_name, last_name from users*

procedūras atbildē tas izskatītos šādi

```
<user>
<user_info>
<first_name>Jānis</first_name>
<last_name>Bērziņš</last_name>
<user_info>
</user>
```

Padarot iepriekšējo vaicājumu ar kodu `USERS_SELECT` par neaktīvu un ievietojot jaunu rakstu ar šādu kodu, kur vaicājuma teksts ir

```
select first_name, last_name, middle_name from users
```

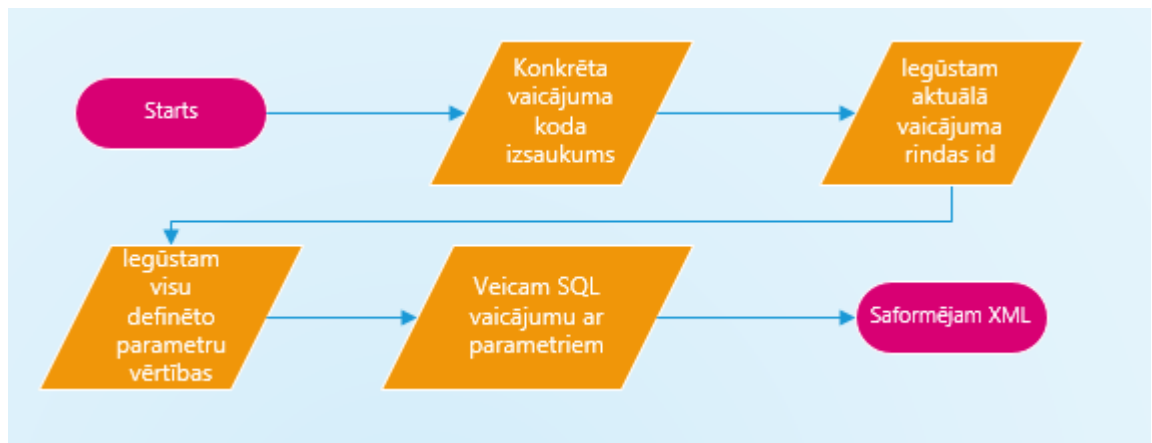
procedūras atbilde ir šāda

```
<user>
<user_info>
<first_name>Jānis</first_name>
<last_name>Bērziņš</last_name>
<middle_name>Māris</middle_name>
<user_info>
</user>
```

Mēs iegūstam jaunu lauku XML atbildē, nemainot procedūras kodu un neiesaistot izmaiņu vadībā visus cilvēkus, kas tiktu iesaistīti citā veidā, tādējādi padarot vienkāršas atgriežamo lauku izmaiņas ātras un nesāpīgas un viegli pārvaldāmas.

SQL vaicājumiem ir arī dinamiski parametri, kuri tiek padoti PL/SQL procedūrā veicot šī vaicājuma izsaukumu. Vienam vaicājumam var būt no viena līdz trijiem parametriem, un tie šobrīd tie var būt kādi no sekojošiem veidiem – klienta id, lietotāja id, personas kods, sim kartes numurs vai iekārtas IMEI kods. Visi šie parametru veidi ir iepriekš definēti un procedūras ietvaros šīm vērtībām ir jābūt zināmām, tas ir vienīgais ierobežojums parametru izvēlē. Katram vaicājumam nedefinētie parametri atrodas tabulas ieraksta kolonnās kā nosaukumi. Vaicājums ar parametriem izskatās, piemēram, šāds

```
select first_name, last_name, middle_name from users
where user_id = :P_USER_ID
```



Attēls 5.1. SQL atbildes iegūšana dinamiski

Savukārt procedūrai veicot darbu, ir redzams apakšprocedūras izsaukums, kurā ir nodefinēts nepieciešamā vaicājuma kods, kurš savukārt griežas pie tabulas, kur atrod vaicājumam paredzētos dinamiskos parametrus, aizpilda tos ar procedūrā esošo mainīgo vērtībām un veic SQL izpildi, kas beigās saformatē XML struktūras atbildi.

Šādai pieejai, gan, ir arī savi mīnusi. Lai arī atgriezties atpakaļ pie iepriekšējās vaicājuma versijas ir viegli un ātri, ievietojot jaunu SQL vaicājuma versiju, pārbaudīt vai izmaiņas ir sekmīgas, ir iespējams tikai izsaucot procedūru. Parametru sintakses dēļ, vaicājuma gala versiju nav iespējams pilnīgi notestēt, un nomainot parametra vērtību pret parametra nosaukumu, ir iespējams kļūdīties un padarīt procedūru par nestrādājošu. Līdzīga situācija var gadīties arī pievienojot vaicājumā tabulu, uz kuru VAAC datubāzes shēmai nav piešķirtas tiesības, arī šādā gadījumā procedūra kļūst nederīga un izsaukums beidzas ar kļūdu.

Kopumā, gan, šādas vaicājumu izmaiņu vadības pieejas plusi vairākkārt atsver mīnusos, ja galvenais mērķis ir padarīt sistēmu par dinamisku izmaiņu veikšanā.

5.2. Provizoriskais lietotāja rēķins

5.2.1. Risinājuma izvēle

Lai arī prasībās nebija specificēts, ka lietotāja tekošā rēķina detalizāciju nepieciešams izveidot dinamiski pieteikšanās brīdī, atkarībā no tajā brīdī spēkā esošajiem pakalpojumiem un pakalpojumu plāniem, tika pieņemts lēmums provizoriskā rēķina datus izrēķināt uzreiz pie autorizācijas. Alternatīvs variants bija pēc lietotāja pieteikšanās piedāvāt datu atjaunināšanas pogu, kas dotu komandu uz datubāzi izveidot rēķinu un saformēt rēķina rindas, kuras tiktu padotas uz VAAC sistēmu, savukārt pats rēķins tiktu anulēts, kas būtu daudz vieglāks un ātrāk ieviešams, taču lietotājam mazāk draudzīgs.

5.2.2. Dinamiska prognozējamā rēķina izveidošana

Lai izveidotu prognozējamo rēķinu tekošajam mēnesim ar izmaksu skaidrojumiem lielākā problēma bija izveidot atbilstošu konfigurāciju, kas pareizi interpretē bilanču tabulā esošās summas un to veidus. Izmantojot bilanču tabulas datus bija nepieciešams izveidot rēķina rindu veidus, kuros informācija tiek sagrupēta zem attiecīgajām rindām.

Bilanču tabula ir vieta, kur tiek kopā sagrupēti visas konkrētā tīkla notikuma veida instances zem viena raksta. Galvenie svarīgie bilanču tabulas lauki šai atpazīšanai ir

- Klienta un lietotāja id
- Konta id, kas nosaka bilances konta veidu, pēc kā var noteikt vai tas ir vietējais vai viesabonēšanas notikums
- Finanšu pakalpojuma id, no kura noskaidrojams vai konkrētā bilances rinda ir piesaistīta konkrētam pakalpojumam
- Virzienu no un virzienu uz, kas parāda vai tas ir iekštīkla notikums vai nē
- Summu naudas izteiksmē, kopējo bilanču rindas notikumu ilgumu, skaitu un datu patēriņa apjomu, kuri tiek izmantoti

Tā kā ne vienmēr var būt iespējams savākt visus datus no bilanču tabulas tiešsaistē, nodaļā 4.2.1. minētais process, kā arī tiešsaistes procedūra izrēķinātās rindas ievieto tabulā, no kurienes tās tiek atlasītas par konkrēto lietotāju un pievienotas klāt konkrētā lietotāja bilances procedūras atbildes datiem XML formātā.

Protams, arī šādam risinājumam ir savi mīnusi, no izstrādes viedokļa ir daudz vieglāk izveidot rēķinu simulatoru, kurš izveidos šādas rēķina rindas ar 100 % precizitāti, taču tas notiek salīdzinoši lēni un nav konfigurējams VAAC sistēmas vajadzībām. Otrs mīnuss ir tas, ka šī rēķināšana patērē salīdzinoši lielāku laiku ar pārējiem datiem, kur sevišķi nekas rēķināts netiek, tādējādi paildzinot lietotāja datu savākšanas ilgumu piesakoties, kas varētu būt pavisam niecīgs, ja vienmēr tiktu izmantoti tikai naktī aprēķinātie dati, taču tas neietekmē pieteikšanās procesu tik ļoti, lai atteiktos no reāllaika datu rēķināšanas.

5.3. Konfigurējamo plānu lietošanas integrācija

Izmēru ziņā lielākajiem klientiem ik pa laikam tiek veidoti specializētie, tieši klientam piemērotie pakalpojumu plāni. Tā kā pašā plāna izveidē nepieciešams iesaistīt lielu cilvēkresursu un tas prasa kādu laiku, šāds speciālo plānu skaits nav sevišķi liels, taču no VAAC sistēmas puses problēma rodas tad, kad ar šiem plāniem, kas ir piesaistīti konkrētam klientam,

ar speciāli izveidotiem pakalpojumu skaitītājiem, jāveic darbības, piemēram, jāizveido jauns lietotājs, jāmaina esošam lietotājam plāns uz šādu, vai jāparāda lietotājam pieslēgtā plāna, tieši datu pārraides megabaitu atlikušās vienības.

Lai gan arī vecajā sistēmā bija pieejams elementārs šo plānu pielietojums – bija iespējams veikt maiņu uz šo plānu, ja plāns bija izveidots pirms asinhronā procesa darbības beigšanas, nebija iespējams ne uzzināt plāna skaitītāju atlikumu, ne arī veikt jebkādas darbības ar šo plānu, ja līgums par plāna izveidi bija parakstīts nevis ar konkrēto klienta numuru, piemēram, konkrēto degvielas uzpildes staciju, bet gan ar virsklienta numuru, piemēram, degvielas uzpildes stacijas zīmola tiesību turētāju.

Šo, lielāko klientu, kas lielākoties ir juridiskas personas, virtuālo plānu pilnu atbalstu atsevišķā sistēmā realizēt ir izaicinoši, jo katrs no šiem plāniem var būt pilnīgi individuāls, kas nozīmē pilnīgi atsevišķus skaitītājus, nosacījumus un cenu. Lai arī skaitītāji, lielākoties tiek atkalizmantoti, kopējais unikālais skaitītāju veidu apjoms tieši šiem konfigurējamajiem plāniem jau pārsniedz 40. Ieviešot šo konfigurējamo plānu atbalstu VAAC sistēmā, pastāv risks ievērojami samazināt ātrdarbību, jo, kā jau minēts, šī speciāli veidotā plāna līgums un nosacījumi var atrasties zem pavisam cita klienta numura, tāpēc nav iespējams vadīties tikai pēc tā, kādi pakalpojumu plāni klienta lietotājiem ir pieslēgti šobrīd, jo līgums var būt stājies spēkā pirms pavisam neilga laika.

6. JURIDISKU PERSONU SISTĒMAS PROJEKTĒJUMS

6.1. Privātpersonu un juridisko personu dalījums

Izpētes laikā tika noskaidrots, ka izveidotie procedūrās esošo vaicājumu prototipi strādāja ļoti labi ar salīdzinoši nelielu lietotāju skaitu klientā (1-10), savukārt klientiem ar 10+ lietotājiem, kā bija paredzēts, vaicājumi strādāja lēnāk, proporcionāli lietotāju skaitam, kur vairs netika apmierināta prasība, par laiku, kādā jāsaņem dati, klientam piesakoties sistēmā. Autors pieņēma lēmumu nevākt lielajiem klientiem pilnīgi visus datus par tā lietotājiem pieteikšanās, bet gan tikai pamatdatu. Tādējādi abām klientu grupām tiek nodrošināta datu savākšana pieteikšanās laikā līdz 4 sekundēm, savukārt lielajiem klientiem datu savākšana par pārējiem lietotājiem notiek fonā. Brīdī, kad klients izdomā šo lietotāju apskatīt detalizētāk, uzreiz tiek izsaukta procedūra, kura šos datus sarēķina un atgriež, un klientam tie ir pieejami pēc aptuveni 2 sekunžu aiztures.

6.2. Datu ieguve (procedūras)

Juridiskām personām tiek pielietota cita shēma. Tā kā ar privātpersonu procedūrām nav iespējams panākt nefunkcionālo prasību izpildi, jo netiek nodrošināts pietiekami ātra datu ieguve, tad juridiskām personām procedūru kopa izskatās mazliet savādāka.

Izsaucot klienta virsējo procedūru, starpniekserverim tiek atgriezts XML saraksts ar juridiskās personas klienta lietotājiem, to vārds, uzvārds, pieslēguma numurs un lietotāja numurs, kā arī tiek izsaukta tā pati klienta procedūra, kas atgriež datus privātpersonām, tādējādi tiek nodrošināta visa informācija, kas klientam ir nepieciešama pirmajā lapā. Juridiskām personām ir divas papildus procedūras, kuras tiek izsauktas pēc pieprasījuma, ja fonā šī informācija pēc pierakstīšanās vēl nav savākta.

- Klienta datu procedūra

Šeit tiek izmantota tāda pati procedūra kā privātpersonām, taču ar nelielām izmaiņām. Tiek vākti nevis pēdējie 3 rēķini, bet pēdējie 18, kā arī šīs procedūras atbildē ir sagaidāms daudz vairāk datu, jo tiek vākta arī informācija par pakalpojumiem, kas ir klienta līmenī. Ja neskaita nelielās izmaiņas, tad būtībā būtu bijis iespējams izmantot vienu procedūru abās pieteikšanās shēmās.

- Lietotāja un bilances procedūra

Līdzīgi kā privātpersonām, arī juridiskas personas lietotājam šīs procedūras ietvaros tiek atgriezti visa informācija par lietotāju, atšķirība ir tā, ka šeit vienā procedūrā ir šie dati apvienoti ar bilances datiem.

Papildus bilances datu atlasei, jaunākās bilanču datu vērtības tiek aizstātas ar bilanču procesa aprēķinātajām, vai citā veidā iegūtajām, pēdējām vērtībām

- Lietotāja pakalpojumu procedūra

Unikāla tieši juridiskām personām ir lietotāja pakalpojumu procedūra, kur tiek atgriezti visi lietotāja pakalpojumi un to parametri. Iemesls tam ir tāds, ka juridisko personu lietotājiem potenciālais pieslēdzamo pakalpojumu saraksts var būt ļoti liels, tāpēc tika pieņemts lēmums šo procedūru rezultātus nodalīt.

6.3. Asinhronie procesi

6.3.1. Lietotāju bilanču process

Process nakts laikā aprēķina visu lietotāju VAAC bilances un skaitītāju atlikumus. Aprēķinātā informācija tiek izmantota gadījumos, kad tarifkācija ir lietotāju nobloķējusi, vai arī rezultāts tiek rēķināts ilgāk, nekā tam vajadzētu notikt, tādās reizēs bilances informācija tiek rēķināta no saglabātās informācijas par lietotāju. Ja lietotāja bilances procedūra nostrādā veiksmīgi, tad procesa izrēķinātā informācija tiek aizstāta ar tikko aprēķināto klienta pierakstīšanās brīdī.

6.3.2. M2M lietotāju datu process

“Machine to Machine”, jeb “iekārta pie iekārtas” lietotājiem ir paredzēts atsevišķs process, kas šo lietotāju atgriežamos datus izrēķina reizi 6 stundās. M2M pakalpojumu plāni ir paredzēti automatizētai saziņai starp iekārtām, piemēram, novērošanas kameras, viedie elektrības skaitītāji vai gps iekārtas, un šo plānu lietotāji gandrīz vienmēr ir juridiskas personas, savukārt, komunicē šie lietotāji izmantojot sms vai datu pakalpojumus. Bez asinhrona procesa šo lietotāju sakarā nebija iespējams izvairīties, jo klienti, kuriem ir daudz šāda tipa lietotāju, vēlas tos pārredzēt tabulas veidā, kas nozīmē, ka šiem datiem vienmēr jābūt relatīvi jauniem, taču nav obligāti jābūt tiešsaistes datiem, jo M2M lietotāju datu sesijas ir garas un centrālē tiek sadalītas pa daļām mākslīgi.

Šis asinhronais process aprēķina konkrētai lietotāju grupai šādas lietas

- Klienta nosaukums
- Vietējais datu patēriņš(MB)
- Viesabonēšanas datu patēriņš(MB)

- Nosūtīto vietējo SMS skaits
- Nosūtīto viesabonēšanas SMS skaits
- Iekārtas IMEI kods
- SIM kartes numurs
- SIM kartes IMSI
- Tīkls, kurā lietotājs pēdējo reizi reģistrējies
- Pēdējā tarificētā notikuma datums
- Datu atjaunošanas laiks
- Statuss (aktīvs/uz laiku slēgts)
- Statusa izmaiņu datums

Lielākā daļa no šiem datiem ir iegūstama ļoti ātri vienam lietotājam, taču ir klienti ar vairākiem desmitiem tūkstošu šādu lietotāju un nav iespējams sagaidīt, kamēr visi šie dati tiktu atlasīti tiešsaistē un tas arī nav vajadzīgs, jo, kā jau minēts, dati šiem lietotājiem mainās vidēji reizi pāris stundās.

6.3.3. Iekārtu datu savākšanas process

Lielajiem klientiem ir savādāka shēma iekārtu datu savākšanai. Tā kā sistēmas lietotāju saskarnē ir paredzēta atsevišķa sadaļa iekārtām, kas reģistrētas uz klienta vārda, rodas problēma, ja ir jāsavāc 20, 50 vai vairāk iekārtas un to informācija, tāpēc speciāli lielajiem klientiem šī informācija par iekārtām tiek savākta reizi dienā vai biežāk, kas satur visus izrēķinātos datus uzreiz, lai pie klienta pieteikšanās viss nav jāērķina no sākuma. Līdz brīdim, kad jāsavāc 5-10 iekārtu dati, laika patēriņš nav sevišķi nozīmīgs, taču katra papildus iekārta paildzinātu lietotāja pieteikšanās laiku.

Process aprēķina katram klientam piesaistīto iekārtu datus, kur galvenie lauki ir šādi:

- KLIENTA_ID
- LIETOTAJA_ID
- IEKARTAS_MODELIS
- IEKARTAS_CENA
- IEKARTAS_MENESA_MAKSAJUMS
- IEKARTAS_MENESA_MAKSAJUMS_BEZ_PVN
- PIESAISTITAIS_NUMURS/LIGUMA_NUMURS
- IEGADES_DATUMS
- MENESU_SKAITS

- APREKINA_LAIKS

6.3.4. Pēdējā rēķina rindu process

Juridisko personu numuriem, kuriem nav pieejas tiesību apskatīt visus numurus un kuri, piemēram, atrodas zem konkrēta klienta, kurš ir viņu darba devējs, arī nepieciešams zināt par sava numura patērēto mēneša griezumā. Varbūt vienkārši intereses pēc, varbūt darba devējs par daļu no iztērētās summas prasa norēķināties. Šādiem pašapkalpošanās sistēmas lietotājiem ir paredzēta sadaļa, kur var apskatīties iepriekšējā rēķina rindas par konkrēto numuru, tādējādi uzzinot, kāda summa ir patērēta par pakalpojumiem, pakalpojumu plāniem un tajos neietilpstošajām sarunām, kā arī papildus informācija, kura var noderēt lietotājam plānojot savas izmaksas. Process tiek darbināts reizi mēnesī pēc visu rēķinu izveidošanas un izvadīšanas un saglabā katra numura iepriekšējā rēķina rindas teksta laukā.

6.4. Papildus datu ieguve

6.4.1. Lietotāju pamatdatu saraksts

Lai nodrošinātu pēc iespējas mazāku pieteikšanās laiku sistēmā juridiskajiem klientiem, kas biežāk ir izmēros lieli nekā privātpersonu klienti, autorizācijas brīdī papildus klienta datiem, kas tiek vākti ar procedūras palīdzību, starpniekservers veic vaicājumu, kas atgriež pamatdatus par visiem lietotājiem. Šajos datos ietilpst sekojoši lauki

- Lietotāja pieslēguma numurs
- Lietotāja vārds
- Lietotāja uzvārds
- Klienta numurs
- Statuss
- Statusa datums
- Rēķinā neiekļautā trafika summa bez PVN

Atgriežot šos datus, VAAC sistēmas tīmekļa saskarnei ir pietiekami daudz datu, lai būtu iespējams veikt meklēšanu, jo, tā kā tīmekļa serverim nav tiešas piekļuves pie datubāzes datiem, meklēšanu pa taisno nodrošināt nav iespējams, tāpēc visi parametri, pēc kuriem ir nepieciešams lietotājus meklēt, tiek padoti pieteikšanās brīdī.

6.4.2. M2M lietotāju pamatdatu saraksts

M2M lietotājiem, kas gan ir iekļauti arī parasto lietotāju sarakstā, lietotāja saskarnes specifiku dēļ, kur M2M lietotājiem ir atsevišķa sadaļa, tiek veikts atsevišķs vaicājums sistēmā, kurš iegūst informāciju no nodaļā 6.3.2. aprakstītā procesa aprēķinātajiem datiem. Šī veida lietotāju pamatdatu sarakstā tiek atgriezta visa tā pati informācija, kas ir procesa aprēķināta.

6.5. Datu konfigurācija

Lielākā daļa no juridisko personu konfigurācijas pārklājas ar privātpersonu konfigurācijas tabulām, tās ir

- VAAC_PAKALPOJUMI
- VAAC_PAKALPOJUMU_PLANU_GRUPAS
- VAAC_IESNIEGUMU_VEIDI
- VAAC_IESNIEGUMU_PARAMETRI

Veidotas primāri šī veida klientiem, taču, ir arī konfigurāciju tabulas, kuras bija nepieciešams ieviest vai papildināt līdz ar juridisko personu sistēmas veidošanu.

VAAC_PAKALPOJUMU_PLANU_GRUPAS

Šī tabula tika papildināta ar vairākiem plāniem, kas paredzēti tieši juridiskajām personām, kā arī ar lauku PLANA_PIEEJAMIBA, kas norāda to, kuram klienta veidam šis plāns ir paredzēts: tikai privātpersonām, tikai juridiskām personām vai abiem.

VAAC_VIRTUALIE_PLANI

Speciāli juridisko klientu vajadzībām tika veidota šī konfigurācijas tabula, kurā tiek nodefinēti konfigurējamo plānu veidi ar kolonnām

- ID

Tiek izmantota kā primārā atslēga šajā tabulā un šī atslēga tiek izmantota arī virtuālo plānu skaitītāju veidu tabulā

- PLANA_VEIDS

Balss/dati/M2M

- DATUMS_NO

No kura datuma šāda veida plāns ir pieejams

- DATUMS_LIDZ

Līdz kuram datumam šāda veida plāns ir pieejams

VAAC_VIRTUALO_PLANU_SKAITITAJI

Konfigurācijas tabula paredzēta skaitītāju sasaistei ar to, kuram virtuālajam plānam tie var būt piesaistīti

- SKAITITAJA_NOSAUKUMS

Kā skaitītājs sauksies VAAC sistēmā

- PLANA_ID

Parāda kurā virtuālajā plānā skaitītājs ir pieejams

7. PRODUKCIJAS SISTĒMAS ĀTRDARBĪBA

Nodaļā tiks pētīta jaunās sistēmas ātrdarbība, kā arī salīdzināta ar veco, atsevišķi tiks skatīta gan privātpersonu sistēma, gan juridisku personu sistēma.

7.1. Datu ieguves ātrdarbība piesakoties

Datu ieguves ātrdarbība tiks pētīta tikai datubāzes galā, netiks skatīts kopējais pieteikšanās laiks sistēmas lietotāju saskarnē, jo šajos rezultātos vēl tiek iekļautas pēcapstrādes datiem, kas gan vecajā gan jaunajā shēmā ir aptuveni vienādas, kā arī tīmekļa servera galā dati netiek atrādīti uzreiz kā tiek savākti, bet gan reizi pāris sekundēs notiek pārbaudes izsaukums uz starpniekserveri par to, vai dati ir gatavi.

7.1.1. Privātpersonas

Privātpersonu sistēmai datu ieguves ātrdarbību pārbaudīt ir pavisam vienkārši, notiek jumta procedūras izpilde, kas atgriež konkrētajam numuram vai personas kodam pieejamo klientu sarakstu un tad noteikt konkrētā klienta datu savākšana, izsaucot vienu galveno procedūru, kas savāc visus datus par klientu un tā lietotājiem. Datu ieguves ātrums ir atkarīgs no klienta lieluma un, galvenokārt, no sistēmas noslodzes, tāpēc šī metrika tika mērīta dažādos diennakts laikos ar dažāda izmēra klientiem.

laiks\klients	Klients1(2 liet.)	Klients2(2 liet.)	Klients3(7 liet.)	Klients4(7 liet.)	Klients5(15 liet.)	Klients6(16 liet.)
09:30	0.373	0.243	0.752	0.432	0.941	0.706
09:32	0.288	0.250	0.743	0.543	0.948	0.854
16:00	2.876	2.948	2.911	1.931	4.839	5.112
16:02	3.212	2.698	3.012	2.462	4.260	3.765
16:25	1.198	1.265	0.984	0.790	1.433	0.916
23:00	0.241	0.267	0.635	0.387	0.820	0.703
23:02	0.235	0.227	0.610	0.402	0.708	0.667

Attēls 7.1. Datu ieguves laiki privātpersonām

Dažādos laikos piefiksējot procedūras izpildes laiku rīkā PL/SQL developer, tika iegūta augstāk redzamā tabula. Visi laiki ir fiksēti sekundēs un, pārskatāmības nolūkos, ar 3 cipariem aiz komata. Izpildes ātrums tika mērīts 2 dažādiem klientiem ar 2,7 un 15 lietotājiem, lai iegūtu pilnīgāku pārskatu pār to, kā lietotāju skaits ietekmē datu ieguves ātrumu. Ar zaļu iekrāsoti labie rezultāti, ar dzeltenu neitrālie un ar sarkano sliktie.

Globāli, tabulā uzreiz redzams, ka pat klientiem ar vienādu lietotāju skaitu, ātrdarbība konsekventi ir lielāka vai mazāka nekā tikpat lielam klientam, ir vairāki faktori, kas to varētu ietekmēt:

- Klienta vecums – jo tas ir senāks klients, jo vairāk katram lietotājam var būt vēsturiskie dati, rēķinu skaits utt.
- Klienta lietotāju aktivitāte – ja klientam visi lietotāji ir regulāri balss numuri, tad ierakstu skaits bilancēs, provizoriskajā rēķinā, pakalpojumos, var būt stipri lielāks nekā tad, ja lietotāji ir tikai statistiski datu pārraides numuri.

Skatoties tālāk uz dienas periodiem, kas saistās ar procedūru izpildes laikiem, ir redzams, ka tā sauktajos klusajos periodos, neilgi pēc darba laika sākuma un neilgi pirms pusnakts, izpildes laiki ir diezgan līdzīgi un nelieli, kas varētu nozīmēt aptuveno pieteikšanās laiku 2 sekundes. Taču laika posmā pēc 16:00 procedūru izpildes laiki sasniedz neakceptējamus ilgumus un klientam varētu nozīmēt pieteikšanās ilgumu 4-6 sekundes. Sākotnēji tika veikti divi ātruma mērījumi, taču salīdzinot laikus ar rītu, tika nolemts veikt vēl vienu mērījumu pēc neliela pārtraukuma, lai saprastu vai tā ir bijusi tikai īslaicīga anomālija. Veicot atkārtotu mērījumu pēc 20 minūtēm, tika secināts, ka tik lieli izpildes laiki tiešām ir bijuši kā īslaicīga anomālija. Pēc datu apkopošanas tika meklēti skaidrojumi, daži no iespējamajiem ir šādi:

- Datubāzes anomālija ir visticamākais iemesls, kas 20 minūtes vēlāk normalizējās un neatkārtojās arī nākamajā dienā pēc sākotnējo mērījumu veikšanas. Atkārtotā mērījumā nākamajā dienā, šajā pašā laikā, procedūras izpildes laiki bija līdzīgi kā tie, kas tika tabulā iezīmēti ar dzeltenu.
- Lielāka vidējā sistēmas noslodze, gan no klientu gan no sistēmas iekšējo lietotāju puses. Ar to arī skaidrojami ~40-300 % palielinājums procedūru izpildes laikos.

Kopā privātpersonu sistēmas ātrdarbība var tikt atzīta par labu. Zema starprezultātu skaita dēļ autors nevēlas rēķināt un saukt vidējo pieteikšanās laiku, taču, pēc atkārtota mērījuma laika posmam 16:00-16:05 nākamajā dienā, autors secina, ka ar klientu radītu noslodzi šiem laikiem nebija sakara.

7.1.2. Juridiskas personas

Lai pārbaudītu juridisko personu sistēmas ātrdarbību ir jāizmanto mazliet savādāka shēma, jo, ātrdarbības dēļ nav iespējams izmantot visas privātpersonu procedūras, tādējādi iegūstot visus datus par visiem klienta lietotājiem uzreiz. Šiem klientiem datu atlasē pie pieteikšanās tiek izsaukta klienta procedūra un lietotāju saraksts, kā arī atsevišķais M2M

saraksts, ja šādi lietotāji klientam ir, atšķirība starp lietotāju sarakstiem un procedūras atbildi ir tā, ka tikai procedūras atbilde tiek formēta kā XML struktūras teksts, savukārt lietotāju saraksts tiek apstrādāts starpniekserverī uzreiz. Tā kā viena klienta ietvaros, izsaukumi notiek vienā sesijā, tad procedūras izpildes laiks jāskaita kopā ar abu lietotāju sarakstu atlasē laiku. Klientam ar daudz M2M lietotājiem pieteikšanās laikam būtu jābūt ilgākam, jo tie atrodas arī parasto lietotāju sarakstā, dubultojot laika patēriņu saraksta izveidē. Tika izvēlēti 3 dažāda lieluma klienti, no kuriem vienam bija arī liels daudzums ar M2M lietotājiem. Lietotāju skaits klientam ir norādīts iekavās, kur pirmais skaitlis ir kopējais lietotāju skaits, savukārt otrais skaitlis ir M2M lietotāju skaits.

laiks\klients	Klients1(100+0)	Klients2(1019+0)	Klients3(8296+8271)
10:00	0.268	0.418	1.589
10:03	0.233	0.421	1.374
16:30	0.489	0.658	1.950
16:33	0.437	0.642	1.759
23:30	0.298	0.363	1.429
23:33	0.214	0.402	1.348

Attēls 7.2. Datu ieguves laiki juridisko personu klienta datu procedūrai

Klienta datu atlasē nekādu anomāliju nav, ja neskaita nelielu ātrdarbības samazinājumu laikā, kas sakrīt ar darba laika beigu daļu. Fakts, ka datu atlasē ilgums palielinājies neproporcionāli lietotāju skaitam, var būt skaidrojams gan, ar jau minēto, klienta vecumu, gan klienta lietotāju aktivitāti.

laiks\klients	Klients1(100+0)	Klients2(1019+0)	Klients3(8296+8271)
10:00	0.293	0.307	1.203
10:03	0.284	0.304	1.113
16:30	0.304	0.392	1.346
16:33	0.347	0.370	1.440
23:30	0.341	0.348	1.184
23:33	0.291	0.362	1.193

Attēls 7.3. Datu ieguves laiki juridisko personu lietotāju sarakstam

Klienta lietotāju saraksta atlasē visinteresantākais ir tas, ka 100 lietotāju klientam saraksta atlasē aizņēmusi aptuveni tikpat ilgu laiku kā 1019 lietotāju klientam. Tā kā datu kopums un kolonnu skaits ir identisks, autoram nav skaidrojuma kāpēc tas tā ir noticis, jo 8296 lietotāju

klientam lietotāju saraksta atlase ir aizņēmusi vidēji 4 reizes vairāk nekā pirmajiem diviem klientiem.

laiks\klients	Klients1(100+0)	Klients2(1019+0)	Klients3(8296+8271)
10:00	0.000	0.000	0.324
10:03	0.000	0.000	0.338
16:30	0.000	0.000	0.354
16:33	0.000	0.000	0.382
23:30	0.000	0.000	0.419
23:33	0.000	0.000	0.321

Attēls 7.4. Datu ieguves laiki juridisko personu M2M lietotāju sarakstam

M2M lietotāju saraksta atlasē aizpildītas vērtības ir tikai lielākajam klientam, jo pārējiem šī veida pieslēgumu nav.

laiks\klients	Klients1(100+0)	Klients2(1019+0)	Klients3(8296+8271)
10:00	0.561	0.725	3.116
10:03	0.517	0.725	2.825
16:30	0.793	1.050	3.650
16:33	0.784	1.012	3.581
23:30	0.639	0.711	3.032
23:33	0.505	0.764	2.862

Attēls 7.5. Summētie datu ieguves ilgumam juridiskām personām

Sasummētajiem datiem ir redzams, ka līdz ar lietotāju skaita palielināšanos pieaug datu savākšanas laiks, taču tas nav tik izteikti manāms ar klientiem, kuriem ir tikai balss lietotāji, savukārt klientam, kuram ir M2M lietotāji, paiet manāmi ilgāks laiks, lai iegūtu uz pieteikšanās brīdi nepieciešamos datus.

7.1.3. Vecās sistēmas ātrdarbība

Tā kā vecajā sistēmā nebija nodalīta datu atlase starp privātpersonām un juridiskajām personām, visa testēšana notiks ar dažāda lieluma abu veidu klientiem. Testēšanai tika izveidota procedūra ar ieejošo parametru klienta id, kura izpildīja visus vaicājumus(kopā 8 dažādi vaicājumi), kas tika veikti vecajā sistēmā pieteikšanās brīdī un tika piefiksēts tās izpildes laiks apmēram tajos pašos laikos, kuros tika testētas abas jaunās sistēmas. Atsevišķi netiks pētīti arī klienti, kuriem ir M2M pieslēguma veida lietotāji, jo vecajā sistēmā tiem nebija nodalīta datu

atlase. Lai atvieglotu salīdzināšanu, tika izmantoti tie klienti, ar kuriem tika testētas jaunās sistēmas.

laiks\klients	PKlients1(2 liet.)	PKlients2(7 liet.)	PKlients3(15 liet.)	JKlients1(100 liet.)	JKlients2(1019 liet.)	JKlients3(8296 liet.)
10:40	1.316	1.410	1.423	2.739	3.325	4.842
10:43	1.190	1.286	1.434	2.503	3.947	5.025
17:00	1.424	1.392	1.514	2.470	3.674	4.985
17:03	1.338	1.460	1.389	2.698	4.147	5.182
23:50	1.267	1.291	1.493	2.732	3.780	4.697
23:53	1.194	1.388	1.305	2.863	3.671	4.728

Attēls 7.6. Datu ieguves ātrumi vecajā sistēmā

Ar P burtu ir apzīmēti privātpersonu klienti, savukārt ar J burtu ir apzīmēti juridisku personu klienti, savukārt iekavās ir norādīts konkrētā klienta lietotāju skaits. Tabulā redzams, ka procedūras izpildes ātrumus nav manāmi ietekmējusi datubāzes noslodze, jo ilgums visas dienas garumā ir saglabājies diezgan līdzīgs. Otrs secinājums par datu ieguves ilgumiem ir tāds, ka pamatdatu ieguves ilgums ir bijis diezgan konstants, nedaudz virs 1 sekundes, savukārt katra papildus lietotāja datu atlase, vidēji ir aizņēmusi salīdzinoši īsu laiku. Nedaudz negaidīti vislielākā klienta visu datu atlases ātrums arī nav proporcionāli lielāks stipri mazākiem klientiem, pat ja tiek atņemta bāzes datu atlases 1 sekunde. Iespējams skaidrojums tam varētu būt tas, ka 3. juridiskā klienta lietotāji, galvenokārt ir M2M lietotāji, tāpēc ir mazāka datu dažādība.

7.1.4. Vecās un jaunās sistēmas ātrdarbības salīdzinājums

Lai saprastu, vai jaunizveidotās sistēmas ātrdarbības ziņā ir labākas par veco, vizuāli redzēt salīdzinošā tabulā, kur tiem pašiem klientiem veikta datu atlase jaunajā un vecajā sistēmā. Privātpersonu datu atlases ātrumu salīdzinājums izskatās šādi.

laiks\klients	Klients1(2 liet.)	Klients4(7 liet.)	Klients5(15 liet.)	laiks\klients	PKlients1(2 liet.)	PKlients2(7 liet.)	PKlients3(15 liet.)
09:30	0.373	0.432	0.941	10:40	1.316	1.410	1.423
09:32	0.288	0.543	0.948	10:43	1.190	1.286	1.434
16:00	2.876	1.931	4.839	17:00	1.424	1.392	1.514
16:02	3.212	2.462	4.260	17:03	1.338	1.460	1.389
16:25	1.198	0.790	1.433				
23:00	0.241	0.387	0.820	23:50	1.267	1.291	1.493
23:02	0.235	0.402	0.708	23:53	1.194	1.388	1.305

Attēls 7.7. Datu ieguves ātrumi privātpersonām

Uzreiz redzams, ka datu atlases ātrums jaunajā sistēmā ir labāks, ja neskaita datubāzes anomāliju. Vecajā sistēmā izpildes laiks ir diezgan konstants un sevišķi neietekmējas no diennakts laika, savukārt jaunajā sistēmā ir spilgtāk redzams tas, ka datu atlases ātrums samazinās tieši pīķa stundās, kas arī ir viena no lietām, kuru būtu vērts optimizēt.

laiks\klients	JKlients1(100 liet.)	JKlients2(1019 liet.)	JKlients3(8296 liet.)	laiks\klients	Klients1(100+0)	Klients2(1019+0)	JKlients1(8296+8271)
10:40	2.739	3.325	4.842	10:00	0.561	0.725	3.116
10:43	2.503	3.947	5.025	10:03	0.517	0.725	2.825
17:00	2.470	3.674	4.985	16:30	0.793	1.05	3.65
17:03	2.698	4.147	5.182	16:33	0.784	1.012	3.581
23:50	2.732	3.780	4.697	23:30	0.639	0.711	3.032
23:53	2.863	3.671	4.728	23:33	0.505	0.764	2.862

Attēls 7.8. Datu ieguves ātrumi juridiskām personām

Salīdzinot juridisko personu datu ieguves ātrumus redzams, ka jaunajā sistēmā procedūru izpildes laiks 2 mazākajiem klientiem ir bijis 4-5 reizes ātrāks nekā vecajā sistēmā, savukārt lielākā klientam tas ir gandrīz 2 reizes mazāks. Jāņem gan vērā ir tas, ka mērīts datu savākšanas ilgums pieteikšanās brīdī, jo, dati par konkrētu lietotāju tiek sagatavoti uz sistēmas izmantotāja pieprasījumu lietotāja saskarnē, šo procedūru izpildes laiki tiks pētīti atsevišķi.

Kopumā jāsaprot, ka jaunajās sistēmās ir panākts ievērojams sistēmas ātrdarbības uzlabojums pieteikšanās brīdī, lai arī tas varētu būt mazliet labāks visaktīvākajās stundās.

7.2. Pārējo procedūru ātrdarbība

Tā kā juridisko personu sistēmai pieteikšanās brīdī tiek atlasīti vien klienta dati un lietotāju saraksti, tipiskā VAAC sistēmas lietošanas sesijā lietotāja saskarnes izmantotājs apskatīsies datus par vairākiem klientam piederošajiem lietotājiem. Brīdī, kad tiek nospiests uz konkrētu numuru, uz datubāzi aiziet izsaukums, kas atgriež vai nu lietotājam pieslēdzamos un pieslēgtos pakalpojumus, vai arī vispārējos un bilances datus.

Sākumā tika pārbaudīta vispārējo lietotāja datu procedūra

laiks\lietotājs	Lietotājs1(balss)	Lietotājs2(M2M)	Lietotājs3(Dati)
15:15	1.340	0.879	1.174
20:15	1.265	0.784	1.391

Attēls 7.9. Lietotāja datu procedūras ātrumi dažādiem lietotājiem

Lai arī sākotnēji patērētais laiks izskatās liels, jāņem vērā, ka procedūras atbildē ir visi dati par lietotāju, kā arī bilanču dati, kas savākti tiešsaistes režīmā. Balss un datu lietotājiem

atgriežamo datu daudzums ir līdzīgs, taču M2M lietotājam tas ir mazliet mazāks, jo bilanču skaits ir mazāks un tīkla notikumi ir mazāk daudzveidīgi.

laiks\lietotājs	Lietotājs1(balss)	Lietotājs2(M2M)	Lietotājs3(Dati)
15:00	0.275	0.159	0.187
20:00	0.289	0.147	0.262

Attēls 7.10.. Pakalpojumu datu procedūras ātrumi dažādiem lietotājiem

Pakalpojumu datu procedūras izpildes laiks ir ievērojami mazāks, jo tiek atlasīti diezgan statistiski dati, kas sastāv no lietotājam pieslēgtajiem pakalpojumiem un to parametriem, kā arī tiem pakalpojumiem, kas šobrīd ir pieslēdzami. Arī šeit M2M lietotājam atlases laiks ir manāmi mazāks, kas skaidrojams ar mazāku klāstu ar pieslēdzamiem pakalpojumiem.

Kopumā jāsecina, ka arī ātrdarbība pēc pieteikšanās ir apmierinoša un sevišķas aiztures klientam strādājot ar saviem datiem nav. Izveidojot datu savākšanu šādā shēmā, ka paši svarīgākie dati tiek savākti pieteikšanās brīdī, bet pārējie, pēc pieprasījuma, tiek apmierinātas klientu vajadzības ātri sākt darbu ar sistēmu un neapgrūtināt sistēmu ar nevajadzīgiem datu pieprasījumiem, kas klientam nav vajadzīgi.

7.3. Asinhrono procesu ātrdarbība

Šajā nodaļā tiks apskatīta to procesu ātrdarbība, kas darbojas vismaz reizi dienā, tādējādi, potenciāli visvairāk radot slodzi uz datubāzi. Procesi, kas darbojas reizi mēnesī, šeit netiks apskatīti.

7.3.1. Lietotāju bilanču process

Visapjomīgākais un ilgākais no VAAC sistēmas procesiem ir lietotāju bilanču process. Tas izveido rezerves datus visiem lietotājiem ar informāciju par bilancēm un provizorisko rēķinu. Sākotnēji tika plānots šo procesu laist bez paralēlām instancēm, taču tika saprasts, ka šādā veidā nevar nodrošināt pietiekamu ātrdarbību, tāpēc tika pieņemts lēmums sadalīt lietotājos intervālos un laist 5 paralēlas instances, lai šos rezerves datus būtu iespējams izveidot nakts laikā un neslogotu datubāzi dienas laikā, kad tā tiek izmantota visvairāk. Visi 5 paralēlie procesi sāk darbu 03:00 katru nakti, izņemot pirmais, kas sāk darbu 15 minūtes ātrāk, lai izdzestu iepriekšējās nakts rezerves bilances. Procesu ātrdarbība tika mērīta ņemot vidējo katras instances ilgumu pa mēneša dekadēm un vidējais ilgums ir attēlots stundās.

dekāde\instance	Process1	Process2	Process3	Process4	Process5
1-10	4.72	4.36	4.98	3.78	4.33
11-20	5.28	5.12	5.86	4.40	5.05
21-31	6.11	5.80	6.74	5.45	5.89

Attēls 7.11. Lietotāja rezerves bilanču aprēķina procesa ilgums

Redzams, ka mēneša sākumā process strādā ievērojami ātrāk, tas ir skaidrojams ar to, ka mēneša sākumā bilanču apjoms ir stipri mazāks, savukārt jo tuvāk nāk mēneša beigās, jo lielāks ierakstu apjoms procesam jāapstrādā. Jāpiebilst, ka mēneša pēdējās pāris dienās process reizēm darbojās pat virs 7 stundām. Otrs secinājums no šiem datiem ir tāds, ka intervāli varētu būt izveidoti labāk, jo šobrīd tie ir sadalīti pēc lietotāju numuriem, piemēram, pirmā instance apstrādā lietotājus ar identifikatoriem 1-200'000, nākamā instance ņem nākamās. Ideālā gadījumā būtu tā, ka procesu ilgums būtu ļoti līdzīgs, vismaz mēneša sākumā.

7.3.2. M2M lietotāju process

Tā kā šis process darbojas salīdzinoši bieži, bija diezgan svarīgi tā izpildes laiku samazināt pēc iespējas vairāk, tādējādi neslogojot sistēmu dienas laikā, kad to daudz dara citi datubāzes lietotāji. Tāpat kā bilanču procesam, ātrdarbība tika mērīta pa dekādēm, izrēķinot visu procesa darbības reišu vidējo ilgumu stundās.

dekāde\process	M2M datu process
1-10	0.45
11-20	0.53
21-31	0.67

Attēls 7.12. Pakalpojumu datu procedūras ātrumi dažādiem lietotājiem

Tāpat kā iepriekšējam procesam, arī šeit redzams, ka mēnesim tuvojoties beigām, vidējais laiks aizvien palielinās, taču kopējais procesa ilgums ir pietiekami mazs, lai procesu būtu iespējams darbināt reizi 6 stundās un pat biežāk, ja būtu nepieciešams.

7.4. Iesniegumu apstrādes ātrums

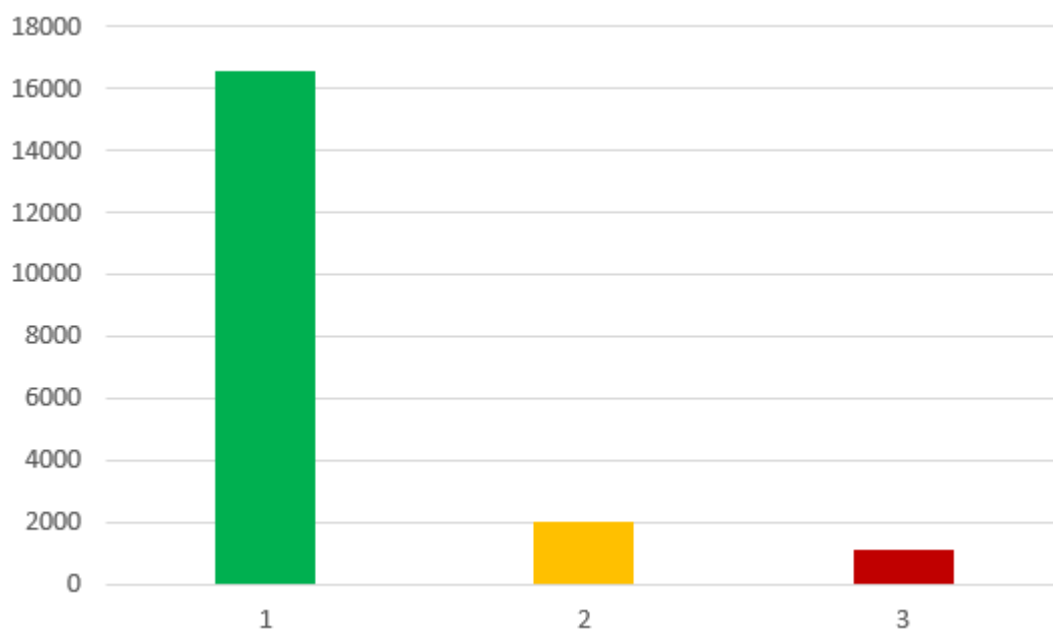
Iesniegumu apstrādes ātrumu tiešā veidā salīdzināt ar veco sistēmu īsti nebūtu korekti, jo tajā iesniegumi netika pildīti ar mūžīgo procesu, kas tos apstrādā uzreiz pēc saņemšanas, bet gan ar procesu, kas tiek palaists reizi 7 minūtēs.

Tika paņemti 2016. gada aprīļa visu saņemto iesniegumu izpildes laiki, un tabula izskatās šādi

ILGUMS SEKUNDES	SKAITS
0	10959
1	2788
2	2849
3	887
4	290
5	355
6	474
7	548
8	329
9	125
10	51
11	19
12	9
13	10
14	8
15	5
16	4
17	2

Attēls 7.13. Iesniegumu izpildes ātrumi

Lai arī no vēsturiskiem datiem nav iespējams iegūt precīzo konkrētā iesnieguma izpildes laiku, un tas ir iegūts no apstrādes beigšanas datuma atņemot saņemšanas datumu, arī no šādiem datiem var diezgan labi izdarīt secinājumus. Redzams, ka pārliecinoši lielākā daļa ir apstrādājusies 0-2 sekunžu laikā, kas nozīmē, ka reālais izpildes laiks šiem iesniegumiem ir bijis 0,01-2,99 sekundes. Neliela daļa iesniegumu ir izpildījusies 3-6 sekunžu laikā kopš iesūtīšanas, kas nav slikti, ja tādi ir mazākums, tāpat, apmēram uz pusi mazāk ierakstu ir izpildījusies 7-17 sekundēs, kas vairs nav sevišķi labs rādītājs. Izrēķinot ar Oracle piedāvāto funkciju vidējo izpildes laiku no visiem iesniegumiem, tiek iegūta vērtība 1,30 sekundes, kas gan ir nedaudz virs plānotās vidējās vērtības.



Attēls 7.14. Iesniegumu ātrumi pa grupām

Šajā tabulā parādīts kā iesniegumu skaits ir izdalīts pa grupām, kur zaļā grupa ir ar labu ātrdarbību, dzeltenā ar viduvēju un sarkanā ar sliktu.

Kopumā jaunās sistēmas iesniegumu apstrāde vērtējama pozitīvi un, lai arī pašas apstrādes ilgums nav sevišķi mainījies pret veco sistēmu, lietotāja veiktās izmaiņas notiek un arī ir tam redzamas daudz ātrāk.

SECINĀJUMI

Jebkuras sistēmas uzlabošanā vai pārveidošanā primārais ir kārtīgi izpētīt esošo sistēmu no visām pusēm un identificēt gan trūkumus, gan pozitīvās iezīmes, tādējādi mēģinot padarīt lietotni labāku ne tikai aspektos, kur ir bijušas problēmas, bet arī nezaudēt jau esošos labos risinājumus.

Ir ļoti grūti atrast balansu starp augstu resursu patēriņu un labu sistēmas lietojamību, datu kontekstā. No vienas puses, jo mazāk dati svaigi dati ir nepieciešami, jo mazāka būs sistēmas noslodze, taču ja reāllaikā tiek iegūti visi vai gandrīz visi dati, arī kompleksi aprēķini no daudzmiljonu ierakstu tabulām, tad nefunkcionālās prasības apmierināt būs tik pat kā neiespējami.

Uzņēmumā, kas strādā ar klientu datiem, ļoti svarīga ir šo datu drošība, pat ja drošības dēļ nākas pielietot risinājumus, kas sistēmu nedaudz var ietekmēt lietojamības ziņā uz sliktu pusi.

Veidojot telekomunikāciju uzņēmuma pašapkalpošanās sistēmu nekādus “no kastes izņemamus”, gatavus risinājumus atrast nav iespējams, nemainot pārējās sistēmas uzbūvi. Vienīgais veids kā veikt inovācijas ir pašizglītoties un spēt saprast kā dažādus mehānismus pielietot konkrētās sistēmas ietvaros.

Projektējuma ieviešanas procesa laikā nāksies saskarties ar apsvērumiem, kas var likt projektējumu izmainīt un tas var prasīt lielus IT resursus.

Sistēmas ātrdarbība pēc ieviešanas atbilst gaidītajam, lai arī bija nepieciešams sistēmu sadalīt divās loģiskās daļās, lai to panāktu. Atsevišķās sistēmas daļās gan to būtu iespējams un ieteicams uzlabot, it sevišķi sistēmas noslodzes pīķa stundās, kad sistēmas ātrdarbība cieš visvairāk.

Maģistra darba mērķis ir sasniegts, jo ir veiksmīgi izdevies izpētīt novecojušo sistēmu un samazināt vai novērst tās trūkumus, ieviešot arī dinamiskus risinājumus, kā arī ir izveidots projektējums, kas apmierina visas funkcionālās darījumasprasības un, lielākoties, arī nefunkcionālās prasības.

PATEICĪBAS

Vilnis Veips, tiešais vadītājs.

Jānis Kalniņš, “SIA AUTENTICA” vecākais programmētājs.

Guntis Vaivods, “SIA AUTENTICA” sistēmu analītiķis.

IZMANTOTĀ LITERATŪRA UN AVOTI

Tīmekļa vietnes un elektroniskie izdevumi (publikācijas, kas pieejamas tikai Tīmeklī)

1. Wikipedia [tiešsaiste] [atsauce 21.05.2016.].
Pieejams https://en.wikipedia.org/wiki/Database_schema
2. Wikipedia [tiešsaiste] [atsauce 21.05.2016.]. Pieejams
https://en.wikipedia.org/wiki/Portable_Document_Format
3. Wikipedia [tiešsaiste] [atsauce 21.05.2016.]. Pieejams
<https://en.wikipedia.org/wiki/Unicode>
4. Vikipēdija [tiešsaiste] [atsauce 21.05.2016.]. Pieejams
<https://lv.wikipedia.org/wiki/SQL>
5. Vikipēdija [tiešsaiste] [atsauce 21.05.2016.]. Pieejams
<https://lv.wikipedia.org/wiki/PL/SQL>
6. Vikipēdija [tiešsaiste] [atsauce 21.05.2016.]. Pieejams
<https://lv.wikipedia.org/wiki/PHP>
7. Vikipēdija [tiešsaiste] [atsauce 21.05.2016.]. Pieejams
<https://lv.wikipedia.org/wiki/XML>
8. Vikipēdija [tiešsaiste] [atsauce 21.05.2016.]. Pieejams
<https://lv.wikipedia.org/wiki/UTF-8>
9. Orafaq [tiešsaiste] [atsauce 21.05.2016.]. Pieejams
<http://www.orafaq.com/wiki/NCLOB>
10. Stackoverflow [tiešsaiste] [atsauce 21.05.2016.]. Pieejams
<http://stackoverflow.com/questions/3466320/how-can-i-use-an-sql-statement-stored-in-a-table-as-part-of-another-statement>
11. Docs.oracle [tiešsaiste] [atsauce 21.05.2016.]. Pieejams
https://docs.oracle.com/cd/B28359_01/server.111/b28318/datatype.htm#i14946

Maģistra darbs: Dinamisku risinājumu ieviešana telekomunikāciju pašapkalpošanās sistēmas uzlabošanā.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: _____
(Autora paraksts)

Ar savu parakstu apliecinu, ka esmu lasījis augstāk minēto maģistra darbu un atzīstu to par **pieņemrotu / nepieņemrotu** (nevajadzīgo svītrot) aizstāvēšanai Latvijas Universitātes datorzinātņu maģistrantūrā.

Darba vadītājs: _____
(Vadītāja paraksts)

Darbs iesniegts **maģistrantūras sekretariātā** _____.
(Iesniegšanas datums)

Ar šo es apliecinu, ka darba elektroniskā versija ir augšupielādēta LU informatīvajā sistēmā.

Studiju metodiķe: _____
(Metodiķes paraksts)

Recenzents: Prof., Dr.habil.dat. Juris Borzovs
(Akad.amats, zin.grāds, vārds, uzvārds)

Darbs aizstāvēts maģistra gala pārbaudījuma komisijas sēdē

_____ prot. Nr. _____
(Darba aizstāvēšanas datums)

Komisijas sekretārs: _____
(Sekretāra paraksts)