

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Erik Taklai 174651IDAR

**X-TEE TEENUSTE JÕUDLUSTESTIMINE  
EESTI ÄRIREGISTRI NÄITEL**

Diplomitöö

Juhendaja: Kaido Kikkas

PhD

Tambet Artma

BSc

Tallinn 2020

## **AUTORIDEKLARATSIOON**

Kinnitan, et olen koostanud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Erik Taklai

12.04.2020

# SISUKORD

LÜHENDITE JA MÕISTETE LOETELU .....	7
TABELITE LOETELU .....	8
JOONISTE LOETELU .....	10
1. SISSEJUHATUS.....	11
2. ÄRIREGISTRIST JA X-TEEST.....	12
2.1 Äriregister.....	12
2.2 X-tee .....	13
3. TARKVARA TESTIMINE JA KOORMUSTESTIMINE .....	14
3.1 Tarkvara testimine .....	14
3.2 Jõudlustestimise üldpõhimõtted .....	15
3.2.1 Jõudlustestimine.....	15
3.2.2 Jõudlustestimise liigid.....	16
3.2.3 Testimise skoop .....	17
3.2.4 Koormuse profiil .....	17
3.2.5 Testimisel mõõdetavad parameetrid .....	18
3.3 Tarkvara jõudlust ja skaleeruvust mõjutavad faktorid .....	19
3.4 Tarkvara jõudlustestide määramatus .....	19
4. TÖÖ METOODIKA .....	21
4.1 Teenuste reaktsioonaja muutumine .....	21
4.2 Päringute mõju infrastruktuurile.....	22
5. X-TEE TEENUSTE JÕUDLUSTESTIMINE ÄRIREGISTRI NÄITEL .....	23
5.1 Koormustestimise tarkvara valimine .....	23
5.2 Koormustestide plaani koostamine.....	24
5.2.1 Jälgitavad süsteemi komponendid .....	24
5.2.2 Jälgitavate süsteemi komponentide omadused .....	25
5.2.3 Testitavad teenused .....	26

5.2.4 Teenuste sisendid .....	26
5.3 Päringu reaktsiooniaja mõõtmise erinevate sisendite korral .....	28
5.4 Kasutajate arvu kasvu mõju infrastruktuurile .....	30
5.5 Kasutajate arvu kasvu mõju teenuse reaktsiooniajale .....	33
5.6 Testimistulemuste võrdlus .....	34
5.6.1 Teenuse reaktsiooniaeg .....	34
5.6.2 Päringuga edastavate andmete maht .....	34
5.6.3 Teenuse sisendi erinevused .....	35
5.6.4 Kasutajate arvu kasv .....	35
5.6.5 Infrastruktuuri mõõtmistulemused .....	35
5.7 Teenuste analüüs ning soovitused .....	36
5.8 Võimalused süsteemi koormuse vähendamiseks .....	37
KOKKUVÕTE .....	38
KASUTATUD KIRJANDUS .....	39
LISAD .....	41
Lisa 1. Detailandmed_v3 teenuse päring turvaserverisse .....	41
Lisa 2. Detailandmed_v3 teenuse X-tee osaline näidisvastus. ....	42
Lisa 3. Detailandmed_v3 teenuse osaline näidisvastus JSON vormingus. ....	44

## ANNOTATSIOON

Töö eesmärgiks on äriregistri XML ja X-tee teenustele jõudlustestimise plaani koostamine, testide loomine ja käivitamine, testide tulemuste analüüsimine ning arendusosakonnale ettepanekute andmine, kuidas teenuseid efektiivsemaks muuta.

Jõudlustestide plaani koostamisel uuriti, milliseid infosüsteemi komponente ja komponentide omadusi peaks X-tee teenuste jõudlustestimisel jälgima. Samuti analüüsiti, milliseid süsteemi osi peaks esmajärjekorras ning suurema tähelepanuga testima. Plaanis määrati kindlaks ka teenused, mida peaks esmajärjekorras testima ning kuidas neid peaks testima.

Töö käigus loodi valitud tähtsamatele ja probleemsematele teenustele jõudlustestid ning käivitati need. Peale testide käivitamist koostati testiraportid ning testiraportite tulemuste järgi analüüsiti, millistele infosüsteemi osadele tuleks suuremat tähelepanu pöörata ning milliste teenuste töös esineb jõudlusprobleeme. Jõudlusprobleemidega teenustele koostati ka parandusettepanekud, mis võiksid probleemid lahendada või vähendada. Töö käigus uuriti, milliseid põhilisi reegleid võiks XML teenuseid luues jälgida, et vähendada võimalike jõudlusprobleeme teenuste töös.

Töö käigus tehti selgeks, et testides teenuseid peaks tähelepanu esmalt pöörama kõige populaarsematele teenustele ning kõige pikema keskmise reaktsiooniajaga teenustele. Mõlema teenuste komplekti testimisel peaks analüüsima, kas on võimalik teenuste tööd optimeerida ning miks võib teenuse reaktsiooniaeg liialt pikaks venida. Enamasti oli põhjuseks kas andmebaasi päringu kehv optimeeritus või päringu sisendi ja väljundi probleemid.

Infrastruktuuri komponentide koormamine jõudlustestidega viitas, et alati oli esmaseks probleemiks andmebaasiserveri jõudlus, mis lisab tähtsust teenuste andmebaasi päringute optimeerimisele.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 40 leheküljel, 5 peatükki, 5 joonist, 6 tabelit.

# ABSTRACT

## **Performance Testing of X-Road Services by the Example of Estonian Business Register**

The aim of this work is to create Estonian business registers X-Road and XML services performance testing plan. Create and execute tests according to the plan. Analyse the test results and give proposal to the development department how to optimize the services.

This study showed what infosystem components and what characteristics of these components should be monitored during performance tests of X-Road and XML services. In addition, what system components should be tested first and with greater regard. It was also investigated, what main rules should be applied, when designing XML and X-Road services in order to avoid possible performance issues with the services.

The results of this study showed that during the testing of services more attention should be given to the most popular services and services with the longest average response time. For testing both of these service sets, it should be analysed if it is possible to optimize services and why services response time can be longer. Usually the problem was with the bad optimization of the database query for the request or requests input and output problems. Most serious problems were lack of paging on the response results.

Performance tests for the infrastructure components showed that for the tested sample, it was always the database server that had the greatest workload. This emphasizes the importance of optimization of database queries when X-Road services are created. The main characteristic on all tested infrastructure components, that was the first to have problems was servers CPU.

The thesis is in Estonian and contains 40 pages of text, 5 chapters, 5 figures, 6 tables.

## LÜHENDITE JA MÕISTETE LOETELU

Adapterserver	Rakendusserver, mis töötleb turvaserverist ja internetist saabunud XML päringuid ning nende põhjal edastab päringu andmebaasile, koostab saadud päringule vastuse ning vastab sellele.
Ettevõtja	Äriseadustiku tähenduses füüsiline isik, kes pakub oma nimel tasu eest kaupu või teenuseid ning kellele kaupade müük või teenuste osutamine on püsiv tegevus, ning äriseaduses sätestatud äriühing.
JSON	Avatud inimloetava teksti standardvorming atribuudi ja väärtuse paaridest koosnevate andmeobjektide edastuseks, eeskätt serveri ja veebirakenduse vahel. Põhineb JavaScripti alamhulgal, kuid ei sõltu keelest.
Käideldavus	Iseloomustab teabe, IT-süsteemide, inimeste ja protsesside teovõimet ja kättesaadavust sel ajal, millal organisatsioon neid vajab.
Lõim	Lõimed on operatsioonisüsteemi või arvutiprogrammi omadus jaotuda mitmeks protsessiks, mis eeldab, et protsess operatsioonisüsteemis võib koosneda mitmest osast, mis täidetakse paralleelselt (sealhulgas täpsustamata on, mis järjekorras protsesse täidetakse).
REST	<i>Representational State Transfer</i> . Tarkvara arhitektuuri tüüp, mis defineerib nõuded veebiteenuste loomiseks. Veebiteenused, mis alluvad REST arhitektuurile kutsutakse RESTful veebiteenusteks.
RIK	Registrite ja Infosüsteemide Keskus. Eesti riigiasutus, mis haldab põhiliselt Justiitsministeeriumi haldusalas olevaid infosüsteeme ja registreid nagu näiteks äriregister, kinnistusraamat, e-toimik ja karistusregister.
SOAP	<i>Simple Object Access Protocol</i> . Sõnumiprotokoll XML struktureeritud informatsiooni vahetamiseks.
SoapUI	Vabavaraline rakendus SOAP ja REST veebiteenuste testimiseks.

Turvaserver	X-tee andmevahetuskihi oluline komponent. Turvaserver vahendab X-tee liikmete vahelist andmevahetust üle avaliku interneti.
WSDL	<i>Web Services Description Language</i> . Protokoll veebiteenuste kirjeldamiseks ja neile teenustele juurdepääsu võimaldamiseks. See Microsoft'i ja IBM'i koostöös valminud protokoll kirjeldab teenuse poolt kasutatavaid protokolle ja vorminguid.
XML	<i>Extensible Markup Language</i> . Andmete struktureerimiseks mõeldud märgistuskeel.
X-tee	Infosüsteemide andmevahetuskiht, mis tagab interneti põhise andmevahetust digitaalselt peetavate andmekogude vahel.



## TABELITE LOETELU

Tabel 1. Detailandmed_v3 päringu tekitatud koormused 10 lõimega testides.....	31
Tabel 2. Detailandmed_v3 päringu tekitatud koormused 20 lõimega testides.....	32
Tabel 3. Detailandmed_v3 päringu tekitatud koormused 30 lõimega testides.....	32
Tabel 4. 10 lõimega testides Detailandmed_v3 päringu mõju teenuse reaktsiooniajale.....	33
Tabel 5. 20 lõimega testides Detailandmed_v3 päringu mõju teenuse reaktsiooniajale.....	33
Tabel 6. 30 lõimega testides Detailandmed_v3 päringu mõju teenuse reaktsiooniajale.....	33

## JOONISTE LOETELU

Joonis 1. Infosüsteemi järjekorra võrgu mudel.....	18
Joonis 2. Päringute keskmine reaktsiooniaeg.....	28
Joonis 3. Päringute minimaalne reaktsiooniaeg.....	29
Joonis 4. Päringute maksimaalne reaktsiooniaeg.....	29
Joonis 5. Keskmine päringutega edastatud andmete hulk(baiti sekundis).....	30

# 1. SISSEJUHATUS

Eesti riigiasutuste vahel on loodud andmevahetuskiht, mida nimetatakse X-teeiks. Eesti äriregister on üks populaarsemaid X-tee teenuste pakkujaid. Eraettevõtjad saavad kasutada äriregistrist andmete pärimiseks tasulisi XML teenused. Äriregistri infosüsteemi juurde kuulub üle 70 X-tee päringu ning aastal 2019 tehti iga kuu äriregistrist üle 9 miljoni X-tee päringu. [1]

Suur kasutajate arv seab äriregistriga seotud IT infrastruktuurile ka suured nõudmised. Kui teenused töötavad ebaefektiivselt või kasutajate arv tõuseb hüppeliselt, siis kasvab ka infrastruktuuri komponentide koormus ning lõpuks võib süsteemide kasutamine võimatuks osutuda.

Äriregistri X-tee teenustele ei ole tehtud eraldi koormusteste. Selle pärast on läbi analüüsitud, kuidas töötavad teenused suurte koormuste korral ning milliste infosüsteemi komponentidega tekivad esmalt probleemid suuremate koormuste korral. Samuti on analüüsitud, kuidas teenuste tööd saab optimeerida, et vähendada koormust süsteemidele ning kasutaja jaoks vähendada teenuse reaktsiooniaega.

Käesolevas töös koostatakse äriregistri XML ja X-tee teenustele jõudlustestimise plaan, luuakse ja käivitatakse testid, analüüsitakse testiraporti tulemusi ning raporti analüüsi põhjal edastatakse arendusosakonnale ettepanekud, kuidas teenuseid efektiivsemaks muuta.

Töö tõstab äriregistri infosüsteemide töö efektiivsust vähendades koormust infrastruktuuri komponentidele, vähendades teenuse reaktsiooniaega kasutajale ning eemaldades teenustest võimalikud kitsaskohad.

Diplomitöö ülesanne on äriregistri X-tee ja XML teenustele jõudlustestimise plaani loomine, testide loomine ja käivitamine, testide tulemuste analüüsimine ning arendusosakonnale ettepanekute andmine, kuidas teenuseid efektiivsemaks muuta.

## 2. ÄRIREGISTER JA X-TEE

### 2.1 Äriregister

Äriregistris hoitakse andmeid Eestis registreeritud ettevõtjate ja asutuste kohta. Äriregistri registripidajaks on Maakohtu registriosakond ning tehniliseks haldajaks on RIK. Äriregister on üldnimetus, mis koondab endas erinevad alamregistrid nagu äriregister, mittetulundus- ja sihtasutuste register, korteriühistute register, maaparandusühistute register, riigi- ja kohaliku omavalitsuse asutuste register. [2]

Riigiasutused saavad äriregistrist andmeid pärida kasutades X-tee. Äriregister on ühinenud Eesti X-tee ja alates aastast 2017 pakub X-tee versioon 6 teenuseid. Äriregister pakub üle 70 erineva X-tee teenuse. Äriregistri alamsüsteemi teenused on üks populaarsemaid X-tee teenuste pakkujaid Eestis. X-tee faktilehe järgi tehti 2020 jaanuaris äriregistri X-tee alamsüsteemi üle 9 miljoni päringu, mis oli päringute arvult Eestis neljandal kohal olev alamsüsteem. [3]

Eraettevõtjad saavad äriregistrist andmeid pärida kasutades äriregistri XML teenuseid. Äriregistri XML teenuseid on kokku 14 ning need on ärioloogiliselt sama sisuga nagu XML teenused. Võrreldes X-tee teenustega, puudub XML teenustel päringu päises info X-tee alamsüsteemide kohta ning päringu sihtpunkt ei ole X-tee turvaserver, vaid adapterserver. [1]

Äriregistrist saab andmeid pärida kasutades erinevaid teenuseid nagu näiteks:

Lihtandmed\_v1 - X-tee ja XML teenus, mida kasutatakse ettevõtja üldandmete pärimiseks: registrikood, nimi, aadress, õiguslik vorm.

Detailandmed\_v3 - X-tee ja XML teenus, mida kasutatakse detailsemate andmete pärimiseks: üldandmed, isikuandmed, kommertspandiandmed, menetluses avalduse andmed, määruste andmed. See on äriregistri üks keskseid teenuseid andmete pärimiseks. Detailandmete teenuse päringu ja vastuste näidised on töö lisas 1, 2 ja 3.

EttevotjaMuudatused\_v1 - X-tee teenus, mida kasutatakse ettevõtjate kohta toimunud muudatuste info saamiseks. Tavaliselt kasutatakse koos detailandmete teenusega. Esmalt päritakse muudatuste info ning peale seda detailandmete teenusega muudatuste sisu.

Esindus\_v1 - X-tee teenus, mida kasutatakse ettevõtjate esindusõiguslike isikute pärimiseks.

Kõikidel teenustel on eraldi teenuse versiooni number lisatud teenuse nime lõppu selleks, et erinevaid versioone eristada. Kui teenuse sisend või väljund muutub uuenduste käigus, siis peavad ka teenuse kasutajad oma infosüsteemides täiendusi tegema või teenuse pakkuja peab avaldama uue teenuse versiooni. Äriregistris avaldatakse teenuse funktsionaalsuse täiendamise korral alati uus teenuse versioon, mis lihtsustab teenuste uuendamist. Teenuste kasutajad ei pea enda infosüsteemides muudatusi tegema. Soovituslik on kasutada kõige uuemat versiooni, kuid vanad versioonid on ka edaspidi kasutatavad, kuid vähemate ning uuendamata funktsionaalsustega. [4]

## **2.2 X-tee**

X-tee on aastal 2001 avaldatud andmevahetusteenus, mis võimaldab turvaliselt tarbida erinevate asutuste ja ettevõtjate poolt pakutavaid teenuseid läbi interneti. X-tee lihtsustatud struktuur koosneb teenust pakkuvast infosüsteemi andmebaasist, andmebaasiga seotud adapterserverist ja turvaserverist ning teenuse tarbija infosüsteemi rakendusest, adapterserverist ja turvaserverist.

X-teel kasutatakse andmete edastamiseks XML päringuid. Päringud jagunevad struktuurilt päiseks ning sisuks. Päises kirjeldatakse teenuse tarbija ja teenuse pakkuja andmed. Päringu sisuosas määratakse päringu parameetrid, mille aluse teenusepakkuja infosüsteemist infot päritakse. Vastuse päises on samuti teenuse tarbija ja teenuse pakkuja info ning sisuosas edastatakse soovitud info teenuse tarbijale. X-tee teenuste XML kujul näited on toodud käesoleva töö lisades 1 ja 2.

Päringud infosüsteemide ja nende kasutajate vahel toimuvad läbi turvaserverite. Nii teenuse pakkujal kui ka pärijal peab olema seadistatud oma turvaserver, et päringuid teha. Päring koostatakse teenuse tarbija infosüsteemis, millest päring liigub läbi tarbija turvaserveri teenusepakkuja turvaserverisse. Kui päringu tegemine on lubatud, siis teenusepakkuja turvaserverist tehakse päring teenusepakkuja adapterserverisse, mis koostab XML vastuse ning päring saadetakse sama teed mööda tagasi teenuse tarbija infosüsteemi.

X-tee ja selle teenuste ülesehituse, tehnoloogia ja andmevahetusega saab lähemalt tutvuda RIA X-tee andmeteenuste arendajate koolitusmaterjalidest. [5]

## **3. TARKVARA TESTIMINE JA KOORMUSTESTIMINE**

### **3.1 Tarkvara testimine**

Tarkvara testimine on protsess, mis kontrollib, et programmeeritud tarkvara teeks seda, milleks see on mõeldud ja samuti, et see ei teeks midagi milleks see ei ole mõeldud. Samuti võib öelda, et testimine on programmi käivitamise protsess, mille mõte on leida sellest programmist vigu.

Tarkvara testimist saab liigitada musta kasti ja valge kasti testimiseks. Musta kasti testimist tuntakse ka kui andmetel põhinevaks või sisendi ja väljundi testimiseks. Selle testimismeetodiga vaadeldakse programmi, kui musta kasti. Testijat ei huvita, mis protsessid ja struktuurid on programmi sees. Testija uurib ainult seda, millistes olukordades programm ei tööta vastavalt tema nõuetele. Valge kasti testimist tuntakse ka kui loogikal põhinevat testimist. Selle puhul uuritakse programmi sisemist struktuuri ning protsesse ning testiandmed võetakse programmi loogikast.

Kolmandaks saab testimist liigitada ka funktsionaalseks ja mittefunktsionaalseks testimiseks. Funktsionaalse testimise käigus jälgitakse, et programmi funktsioonid töötaksid vastavalt nõuetele. Mittefunktsionaalse testimise käigus testitakse süsteemi jõudlust, vastupidavust, stressitaluvust, skaleeruvust, turvalisust. Põhifookus on sellel, kuidas tõsta kasutajamugavust läbi selle, kui kiiresti süsteem vastab kasutaja tegevustele

Testimist saab liigitada ka manuaalseks ja automaattestimiseks. Manuaalne testimine on tarkvara käsitsi uurimine testija poolt selleks, et leida vigu. Testija testib programmi lõppkasutaja vaatepunktist ning kontrollib, et kõik tarkvara funktsionaalsused töötavad vastavalt nõuetele. Protsessi käigus töötavad testijad läbi testilood ning loovad testiraportid käsitsi ilma kasutamata automatiseerimisvahendeid. Automaattestimise käigus kasutatakse testimiseks automatiseerimise tööriistu ja skripte selleks, et leida vigu tarkvaras. Testijad loovad ja käivitavad testide skriptid, mis kontrollivad tarkvara funktsionaalsusi. [6]

## 3.2 Jõudlustestimise üldpõhimõtted

### 3.2.1 Jõudlustestimine

Üks mittefunktsionaalseid testimise liike on tarkvara jõudlustestimine. Tarkvara jõudlustestimise tegemiseks võib olla erinevaid põhjuseid:

- Määrata kindlaks, kuidas töötavad infosüsteemid suurte kasutaja hulkade korral. Seda tehakse tavaliselt enne rakenduse esmast töökeskkonda paigaldamist selleks, et tuvastada võimalike skaleeruvuse probleeme suurte kasutajahulkade korral.
- Rakenduse töö optimeerimiseks. Kui aja jooksul kasutajate hulk kasvab või mingil muul põhjusel leitakse, et võivad tekkida rakenduse jõudlusprobleemid, siis teostatakse jõudlustestid, et leida võimalusi infosüsteemi komponentide töö optimeerimiseks.

Jõudlustestimine jaguneb erinevateks infosüsteemi jõudluse testimise meetoditeks: koormuse testimine, stressi testimine, vastupidavuse testimine, mahu testimine. [7]

Infosüsteemi jõudluse mõõdikud saab jagada kaheks erinevaks grupiks:

- Teenusele orienteeritud mõõdikud
- Tõhususele orienteeritud mõõdikud

Teenusele orienteeritud mõõdikud on käideldavus ja teenuse reaktsiooniaeg ning need mõõdavad, kuidas infosüsteem vastab lõppkasutaja ootustele. Tõhususele orienteeritud mõõdikud on läbilaskevõime ja ressursside kasutus ning need mõõdavad, kuidas rakendus kasutab temale määratud ressursse. [8]

Infosüsteemi hea jõudluse eesmärgiks peaks alati olema vähendada päringu reaktsiooniaega samal ajal, kui süsteemi läbivate päringute ja vastuste hulk on võimalikult suur. Infosüsteemi optimeerimist peab alustama alati kõige aeglasemast komponendist või kõige rohkem kasutatava ressursiga komponendist selleks, et optimeerimise efekt kogu süsteemi kiirusele oleks suurem. Seda põhimõtet peaks jälgima kõikide tarkvara jõudluse optimeerimise tegevuste juures. [9]

Üldiseid täpsemaid jõudluse standardeid ei ole kokku lepitud. Teenuse reaktsiooniaja mõju kasutajale kohta on tehtud mitmeid uuringuid, kuid ühest standardit siiski ei ole kindlaks määratud. Ühtset standardit on raske kujundada, sest reaktsiooniaja pikkuse mõju

kasutusmugavusele on paljuski tunnetuslik. Erinevad inimesed tunnetavad aeglast reaktsiooniga erinevalt. Kui võrrelda erinevaid uuringuid, siis hinnatakse kasutajale mugavaks reaktsiooniaja piiriks 1-2 sekundit. [8]

### 3.2.2 Jõudlustestimise liigid

Jõudlustestimise saab jagada alaliikideks:

- **Alusandmete saamiseks testimine** - see paneb paika võrdlusmomendi järgmisteks testimistsükliteks. Enamasti mõõdetakse teenuse reaktsiooniga. Testi käivitatakse tavaliselt ühe lõimega (kasutajaga), mis teeb ühte päringut kindla ajavahemiku vältel. Muud koormust sellel ajahetkel süsteemile ei tohi olla. Testi tulemused saab võtta baasväärtusteks järgnevate testide tegemisel ning võrdlemisel, kuidas suurenenud koormus mõjutab süsteemi tööd.
- **Koormustestimine** - rakendust koormatakse, kuni jõutakse püstitatud eesmärgideni, kuid tavaliselt mitte süsteemi toimimise lõppemiseni. Eesmärk on leida, kas süsteem täidab sellele püstitatud eesmärgid käideldavuse, läbilaskevõime ja reaktsiooniaja kohta. Koormustestimine tavaliselt imiteerib kõige sarnasemalt töökeskkonna koormusi.
- **Stressi testimine** - testimise eesmärk on suurendada koormust, kuni testitav rakenduse või infrastruktuuri osa lõpetab toimimise ning teenus enam ei toimi. Stressi testimine eesmärk on teada saada süsteemi võimekus olemasoleva infrastruktuuri korral ning kui suur on hetkel ressursside varu, kuni tekivad probleemid.
- **Vastupidavuse või stabiilsuse testimine** - vastupidavuse testimist viiakse läbi selleks, et leida probleemid, mis tekivad pikaajalise koormuse korral.
- **Suitsutestimine** - testimisel keskendutakse infosüsteemi osadele, mida on võrreldes eelmise testiga muudetud, et leida, kuidas muudatused mõjutasid muudetud komponentide jõudlust.
- **Isolatsioonitestimine** - testimisel keskendutakse teadaolevale probleemsele osale. Sarnaseid kasutajate toiminguid korratakse kuni jõudlusprobleemi põhjused leitakse.

Testimismeetodi valimine sõltub suurel määral testitavast rakendusest, kui palju on aega testimiseks ning mis on testimise eesmärgiks. [8]



### 3.2.3 Testimise skoop

Üldised mittefunktsionaalsed tarkvara jõudluse ja skaleeruvuse testid saab jagada järgnevasse kategooriatesse:

- **Jõudluse regressioonitestimine** – teostatakse selleks, et saada teada, kuidas on mõjutanud uued tarkvara versioonid infosüsteemi jõudlust. See kindlustab, et klientidel ei ole ebameeldivaid üllatusi, kui töökeskkonnas toodet uuendatakse.
- **Jõudluse optimeerimise testimine** – teostatakse selleks, et kasutuses olevate infosüsteemide tööd optimeerida ning muuta tarkvara kiiremaks ja efektiivsemaks.
- **Jõudluse võrdlusuuringud** – teostatakse selleks, et oleks teada, millist jõudlust kasutajad võivad sellelt tarkvaralt oodata. Teostatakse tavaliselt samal riistvaral ja konfiguratsioonil, matkides realistlike pärismaailma töökoormusi, mis eeldatavalt on töökeskkonnas.
- **Skaleeruvuse testimine** – teostatakse selleks, et teada saada, kas tarkvara vastab kliendi kasvavatele ärimahtudele. Sarnaneb stressi testimisele: tarkvarale pannakse suur koormus, et näha kas see täidab nõuded või tekivad probleemid tarkvara töös. [9]

### 3.2.4 Koormuse profiil

Jõudlustestimise korral peab otsustama, kuidas koormus jaotatakse testimise käigus. Koormuse jaotamiseks on enamlevinud järgmised võimalused:

- Kogu plaanitud infosüsteemi koormamine algab ja lõpeb samaaegselt. Kasutajad teevad toiminguid korraga, kuid mitte sünkroonselt. See tähendab, et kasutajad ei tee täpselt sama toimingut täpselt samal ajal kogu testimise aja vältel, vaid kasutajad on oma tegevustega protsessi eri osades.
- Pidevalt kasvava koormusega - algset koormust suurendatakse pidevalt, kuni jõutakse soovitud eesmärgini.
- Pidevalt kasvava astmelise koormusega - algset koormust suurendatakse ajavahemike kindlate ajavahemike järel kindlaksmääratud suurusjärkude võrra. Näiteks iga viie minuti järel suurendatakse lõimede arvu 100, 200, 400, 800.
- Koormuse tõstmine ning vähendamine – algset koormust suurendatakse ajavahemike järel või pidevalt ning peale soovitud taseme saavutamist langetatakse koormust ajavahemike järel või sujuvalt.

Lisaks eelmainitud võimalustele kasutatakse tihti ka teisi lähenemisi, mis on tavaliselt tingitud testitava tarkvara iseärasustest ning testimise eesmärkidest. [8]

### 3.2.5 Testimisel mõõdetavad parameetrid

Infosüsteemidele on võimalik luua üldine sõnumivahetuse järjekorra mudel (Queueing Network Model - QNM), milles on kujutatud kõik suuremad jõudluse mõjutajad ühes infosüsteemis. Joonisel 1 esitatud QNM mudel kirjeldab, kuidas liiguvad teenuse päring ja vastus erinevate süsteemi komponentide vahel keskmises XML veebiteenuse infosüsteemis. [9]

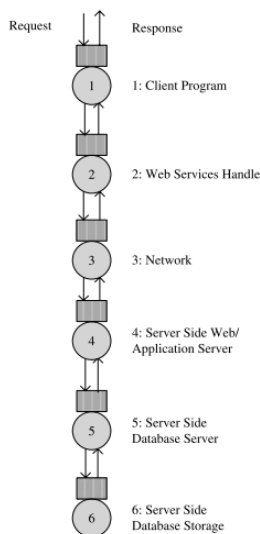


Figure 5.2 QNM for SOA-based applications.

Joonis 1. Infosüsteemi sõnumivahetuse järjekorra mudel. [9]

Lähtuvalt ülaltoodud mudelist on veebiteenuste kohta jälgitavad infosüsteemi komponendid:

- Klientrakendus
- Veebiteenuste tarkvara
- Arvutivõrk
- Rakendusserver
- Andmebaasiserver
- Andmebaasi andmesalvestusseadmed

Olenevalt testitavast infosüsteemist ja testimise eesmärkidest võib erinevaid komponente lisanduda või eemaldada. [9]

### 3.3 Tarkvara jõudlust ja skaleeruvust mõjutavad faktorid

Põhilised tarkvara jõudlust ja skaleeruvust mõjutavad faktorid on järgmised: [9]

- Infosüsteemi riistvara jõudlus ja skaleeruvus. See on üks kõige tähtsamaid faktoreid, mis mõjutab infosüsteemi jõudlust, kui sellel on koormus. Neli põhilist riistvara komponenti: keskprotsessor, mälu, andmesalvestusseadmed ja võrgu infrastruktuur.
- Riistvara seadistused - riistvara süsteemid on disainitud, et potentsiaalselt suurendada jõudlust ja skaleeruvust tarkvaral, mis töötab sellel riistvaral.
- Operatsioonisüsteem - kui samal riistavaral on rakendus paigaldatud erinevatele operatsioonisüsteemidele, siis mõjutab see rakenduse jõudlust.
- Andmebaasimootor - samal riistavaral ja operatsioonisüsteemiga võivad andmebaasist sõltuvad rakendused omada väga erinevat jõudlust.
- Andmebaasisüsteemi seadistused - näiteks, kas andmebaasiserveri päringu optimeerija on uuendatud. Tihti võib päringute kiirus kasvada, kui uuendada andmebaasi päringute statistikat.
- Tarkvara rakenduse seadistused – näiteks, kas rakendus saab töötada mitme lõimega või ühe lõimega, mitu lõime on seadistatud töötama mitmelõimelisel rakendusel või vahemälu kasutamise võimekus.
- Tarkvara disain ja rakendus - olemasoleva tarkvaraga ja seadistusvõimalustega on see kõige tähtsam faktor, mis määrab, kuidas infosüsteem töötab suuremate koormuste korral.

### 3.4 Tarkvara jõudlustestide määramatus

Tarkvara jõudlustestides mõõdetud tulemused on paljuski sõltuvad juhusest. Korrates sama testi samal tarkvaral koos sama seadistuse ja sama infrastruktuuriga saab mõõtmistulemustes tavaliselt erinevusi. Kui testi tulemus on alla 10% erinev eelmise testi tulemusest, siis ei saa testija väita kindlalt, et tulemuse paranemine või halvenemine oli seotud tarkvara või selle seadistuste muudatusega. Seega on tarkvara jõudlustestides statistiline viga üsna suur: suurt täpsust on raske saavutada olenevalt testitavad tarkvarast või selle komponendist.

Kui korrates sama testi on tulemuste erinevus alla 5% siis võib öelda, et vaadeldava testiga on võimalik jõudlusmõõdikuid hinnata. Kui tulemuste erinevus on üle 10%, siis peaks testi uuesti

disainima, et juhuslikust vähendada. Tuleks leida komponendid, mis võivad juhuslikust suurendada ning võimalusel need eemaldada testist.

Testitulemuste suurte kõikumiste korral tuleks rakendada abinõusid, et vähendada probleemseid testi osi, mis võivad tulemuste kõikumisi tekitada.

- Testid tuleb seadistada käima vähemalt 10 minutiks. Pikem testi aeg vähendab juhuslikust testide lõpptulemustes.
- Arvesse tuleb võtta andmete vahemälusse talletamine erinevatel tasemetel: rakenduses, andmebaasis, failisüsteemis, andmesalvestusseadmete tasemel. See mõjutab enim andmebaasidega tugevasti seotud rakendusi.
- Tuleb kindlaks teha, et testis kasutatakse iga kord täpselt sama protseduuri sama rakenduse seadistusega. [9]

## 4. TÖÖ METOODIKA

### 4.1 Teenuste reaktsiooniaja muutumine erinevate koormustega

Arvestades, kui tugevalt mõjutab süsteemide koormus teenuste reaktsiooniaega, siis valis autor üheks põhiliseks indikaatoriks, mida teenuste testimisel jälgida teenuste reaktsiooniaja. Kehvasti skaleeruvatel süsteemil kasutajate arvu kasvades tõuseb vastavalt ka teenuse reaktsiooniaeg. Hästi skaleeruvatel süsteemil võib kasutajate arvu kasvades teenuse reaktsiooniaeg kasvada, kuid see kasvab vaid kuni teatud tasemeni. Samuti ei teki hästi skaleeruvatel süsteemil suuri kõrvalekaldeid keskmisest teenuse reaktsiooniajast. [8]

Arvestades, et äriregistri teenused peavad skaleeruma ka suurte kasutaja hulkade korral, siis otsustas autor äriregistri viie kõige populaarsema teenuse töös võrrelda teenuse reaktsiooniaja ning kasutajate hulga omavahelist seost. Kas ja kui palju kasvab teenuste reaktsiooniaeg koos kasutajate hulga kasvuga?

Teiseks otsustas autor võrrelda, kuidas mõjutab teenuse reaktsiooniaega päringu sisendi ja väljundi muudatused. Kasutades sama teenust saab kasutaja pärida erinevaid andmekomplekte, mis muudavad suuresti teenuse reaktsiooniaega. Analüüsides sisendi ja väljundi muudatuste mõju teenuse reaktsiooniajale saab analüüsida, kas teenuse sees on vaja teha optimeerimist või muudatusi.

Kolmandaks otsustas autor võrrelda, kuidas mõjutab teenuste reaktsiooniaega erinevate süsteemi komponentide kasutamine. Kuna äriregistri teenuseid saab kasutada nii X-tee turvaserveri kui ka ainult adapterserveri, siis saab hinnata, kas ja kui palju mõjutab teenuse reaktsiooniaega turvaserveri lisandumine teenuse päringu ja vastuse teekonda. Samuti on võimalik adapterserverist pärida vastuseid JSON vormingus. Siinkohal on võimalik võrrelda, kuidas mõjutab teenuste reaktsiooniaega, kui vastatakse JSON vormingus sõnumiga.

Kui kõige enam teenuse reaktsiooniaega mõjutavad faktorid on selgeks tehtud, siis saab edasi analüüsida, kuidas optimeerida teenuste tööd või milliste muude võimaluste abil saab teenuse reaktsiooniaega vähendada.

Teenuse reaktsiooniaja muutumise testimise metoodika ja analüüs põhineb raamatu „The Art of Application Performance Testing: Help for Programmers and Quality Assurance“ autori Ian Molyneaux poolt kajastatud metoodikates. [8]

## 4.2 Päringute mõju infrastruktuurile

Infosüsteemi optimeerimist peab alustama alati kõige aeglasemast komponendist või kõige rohkem kasutatava ressursiga komponendist selleks, et optimeerimise efekt kogu süsteemi kiirusele oleks suurem. Seda põhimõtet peaks jälgima kõikide tarkvara jõudluse optimeerimise tegevuste juures. [9]

Selle tõttu on tähtis kontrollida, milliseid komponente koormab teenuse töö kõige enam ning millise komponendi ressursid saavad kasutajate kasvu olukorras esimesena ära kasutatud. Selleks otsustas autor kasutades QNM mudelit määrata põhilised teenuse töös kasutatavad infosüsteemi osad. Leitud infosüsteemi osadele simuleeritakse kasutajate poolt tekitatud koormust ning koormus kasvab ajas. Koormuse tõstmine lõpetatakse, kui on selgelt eristatav, millisel infosüsteemi osal lõppeb esimesena ressurss.

Kui kõige probleemsemad infosüsteemi osad on leitud, siis on võimalik edasi analüüsida, millised võivad olla edasised lahendused, et olukorda parandada.

Päringute mõju infrastruktuurile testimise ja analüüsi metoodika põhineb raamatus „Software Performance and Scalability: A Quantitative Approach“ autori Henry H. Liu poolt kajastatud metoodikatel. [9]

Autor valis eelmainitud viisid teenuste ja infrastruktuuri testimiseks selle pärast, et selline analüüsi viis on tehniliselt kerge ning kiire võimalus, kuidas leida, millised infosüsteemi komponendid, teenused ja teenuste osad võivad olla probleemsed. Teenuste ja infrastruktuuri komponentide testimiseks on palju erinevaid analüüsi võimalusi, kuid autor leidis, et selline lähenemisviis täidab töö eesmärgid kõige kiiremini.

## 5. X-TEE TEENUSTE JÕUDLUSTESTIMINE ÄRIREGISTRI NÄITEL

### 5.1 Koormustestimise tarkvara valimine

XML teenuste koormustestideks sobivale tarkvarale esitas autor järgmised nõuded:

1. Tarkvaraga vastama töös vajalikule põhilisele vajadusele: sellega peab olema võimalik luua koormusteste XML ja REST teenustele.
2. Koormustestide tulemused peab olema võimalik eksportida mõnesse enamlevinud failiformaati andmete töötlemiseks.
3. Tarkvara peab olema vabavaraline.
4. Tarkvara peab olema elujõuline - eeldatavalt jätkuvad tarkvara kasutajatugi ja uuendused järgneva 5 aasta jooksul.
5. Tarkvara kasutamine peaks olema võimalikult lihtne, nii et testide koostamiseks ei ole vaja tugevaid tehnilisi teadmisi ning testide koostamine oleks võimalikult kiire.

Autor analüüsis mitmeid XML teenuste testimiseks sobivaid programme nagu Insomnia ja Postman, kuid koormustestide loomine nendega polnud võimalik. Samuti analüüsis autor erinevaid enamlevinud koormustestimiseks sobivaid tasuta tarkvarasid nagu Gatling ja Eggplant, kuid need ei osutunud valituks kuna leidis lihtsamaid alternatiive. Lisaks juba mainitud tarkvaradele on väga palju erinevaid kommertstarkvarasid nagu näiteks Blazemeter, LoadImpact ja LoadUI Pro, kuid need ei sobinud tarkvara valikutingimustega.

Nõuetele vastasid ning täpsemaks analüüsiks valis autor programmid JMeteri ja SoapUI.

JMeteri positiivsed omadused:

- Üks tuntumaid teenuste testimise tarkvarasid, mille kohta leidub palju kirjandusmaterjali, artikleid ning õpetusi.
- Testitulemused saab kuvada graafikuna tarkvara siseselt ning sealt ka eksportida.
- Koormustestide loomine on võimalik kasutajaliidesest.
- Kasutajal erinevaid funktsionaalsusi tulemuste graafiliseks esitamiseks.

JMeteri negatiivsed omadused:

- Keeruline kasutajaliides - küllaltki vana tarkvara mistõttu ka kasutajaliides pole kõige kasutajasõbralikum.
- Teenuste WSDList pole võimalik teenuste XML automaatselt luua - kasutaja peab päringu XMLi koostama mujal.
- XMLi taandamist pole võimalik teha automaatselt. [10]

SoapUI positiivsed omadused:

- Teenuste WSDList saab luua automaatselt kõikide teenuste XMLid.
- Koormustestide loomine on võimalik kasutajaliidest.
- Testitulemused saab kuvada graafikuna tarkvara siseselt ning sealt ka eksportida.
- Üks tuntumaid teenuste testimise tarkvarasid, mille kohta leidub palju kirjandusmaterjali, artikleid ning õpetusi.

SoapUI negatiivsed omadused:

- Mõned funktsionaalsused on saadaval ainult tasulises versioonis. [11]

Varasemates uurimistöodes on võrreldud JMeteri ja SoapUI sobivust X-tee teenuste testimiseks ning leiti, et sobivam on SoapUI tänu lihtsamale kasutajaliidesele ja WSDList automaatselt teenuste XMLide loomise võimalusele. [12]

Analüüsidest mõlema tarkvara funktsionaalsusi, puudusi ja positiivseid külgi ning lähtudes varasematest analüüsides leidis autor, et tööks sobivamaks tarkvaraks on SoapUI. Põhilisteks argumentideks oleks siinkohal suurem kasutajamugavus. Näiteks on SoapUI kasutajaliides lihtsam ning XML vaikepäringud koostatakse automaatselt.

## **5.2 Koormustestide plaani koostamine**

### **5.2.1 Jälgitavad süsteemi komponendid**

Joonisel 1 on kirjeldatud kõik üldised XML päringuga seotud komponendid. Kui seda mudelit rakendada X-tee teenustele, siis saab täpsustada, mis komponente peaks jälgima. Kuna kliendi programmi ei testita ning teenused testitakse SoapUIs, siis kliendi programm puudub. Rakendusserver koosneb kahest osast: turvaserver ja adapterserver. Seega on jälgitavateks



komponentideks äriregistri veebiteenuste korral on X-tee turvaserver, adapterserver ja andmebaasi server.

Päring esitatakse esmalt turvaserverisse, kust see suunatakse edasi adapterserverisse, mis esitab andmebaasile päringu.

XML teenuste korral on päringuga seotud vähem komponente: adapterserver ja andmebaasiserver. Päring saadetakse adapterserverisse, mis esitab andmebaasile päringu. Seega peaks XML teenuse testimisel jälgima adapterserveri ja andmebaasi tööd.

Kõik mõõtmised teostati testkeskkonnas, millele on eraldatud vähem ressursse, kui töökeskkonnale. Selle tõttu ei ole töökeskkonna ja testkeskkonna testide mõõtmistulemused omavahel otseselt võrreldavad. Võrreldavad on testide tulemused. Kui töös koostatud test on kiirem teisest testist, siis sama tulemus saadaks ka töökeskkonnast .

### **5.2.2 Jälgitavate süsteemi komponentide omadused**

Autor otsustas lähtudes ülalmainitud XML teenuste mudelist ja X-tee teenuste omapäras, et tähtsamaim omadus, mida jälgida on teenuse reaktsiooniaeg.

Lisaks teenuse reaktsiooniajale tuleb jälgida erinevate infrastruktuuri komponentide tööd:

Turvaserver:

- Mälukasutus
- Protsessori koormus

Adapterserver:

- Mälukasutus
- Protsessori koormus

Andmebaasiserver:

- Mälukasutus
- Protsessori koormus

### 5.2.3 Testitavad teenused

Autor valis teenused, mida testida kahe põhilise kriteeriumi alusel:

- Enam kasutatavad teenused
- Kõige pikema reaktsioonijaga teenused

Info äriregistri päringute statistika kohta on pärit RIA poolt edastatud äriregistri X-tee teenuste igakuisest statistikast. Arvesse võeti 2019 aasta esimese 6 kuu statistika: 01.01.2019-31.06.2019.

RIA andmete järgi on äriregistri kõige rohkem kasutatavad teenused:

- esindus\_v1
- detailandmed\_v3
- arikeelud\_v1
- lihtandmed\_v1
- majandusaastaAruanneteKirjed\_v1

RIA andmete järgi on äriregistri kõige pikema keskmise reaktsioonijaga teenused:

- rkoarr\_asutused\_v1
- ettevotjaMuudatusedTasuta\_v1
- toimikuDokumendid\_v1
- detailandmed\_v3
- mtaBilSisu\_v1

Analüüsi esimeses etapis valiti 5 kõige enam kasutatavat ja 5 kõige pikema reaktsioonijaga teenust. Nende teenuste puhul analüüsiti eraldi sisendeid, väljundeid ning koormusi, mis nende päringute käivitamine infrastruktuurile avaldavad.

### 5.2.4 Teenuste sisendid

Teenuste sisendid peaksid jäljendama kasutajate poolt loodud koormust. Koormustestidesse sellist mitmekesisust sisse viia on keeruline ülesanne. Iga teenuse sisendit tuleb eraldi analüüsida.

Töös valiti meetodi näitlikustamiseks teenuse detailandmed\_v3 jõudlustestimise võimaluste tutvustamise. Järgnevalt on näiteks toodud detailandmed\_v3 teenuse sisendi ja soovitud vastuse analüüs. Detailandmete teenuse XML näited on käesoleva töö lisades 1 ja 2.

Lihtsaim variant teenuse kasutamiseks on registrikoodi järgi ettevõtja andmete pärimine. See kasutajalugu on ilmselt ka üks rohkem kasutatavaid:

1. Registrikood on ettevõtja nime kõrval üks põhilisi viise, kuidas kasutajad otsivad ettevõtja andmeid.
2. Välised süsteemid kasutavad andmete saamiseks põhiliselt registrikoodi järgi otsimist. Esmalt otsitakse ettevõtjamuudatused\_v3 teenusega, millistel ettevõtjatel on toimunud muudatus. Peale seda käivitatakse detailandmete päring, et ettevõtja muudatused enda infosüsteemi salvestada.

Teine erisus, mida uurida, oleks teenuse väljundi muudatused. Pärides ettevõtja andmeid saab küsida erinevaid andmeliike.

1. Kasutaja soovib ainult ettevõtja üldandmeid
2. Kasutaja soovib ettevõtja üld- ja isikuandmeid
3. Kasutaja soovib ettevõtja üld- ja isiku- ja kommertspandiandmeid
4. Kasutaja soovib ettevõtja üld-, isiku-, kommertspandi- ja määruste andmeid
5. Kasutaja soovib ettevõtja üld-, isiku-, kommertspandi-, määruste- ja menetluses avalduste andmeid

Koos päringu vastusesse lisanduvate andmetega suureneb info hulk, mida edastatakse ning andmebaasi päringu keerukus.

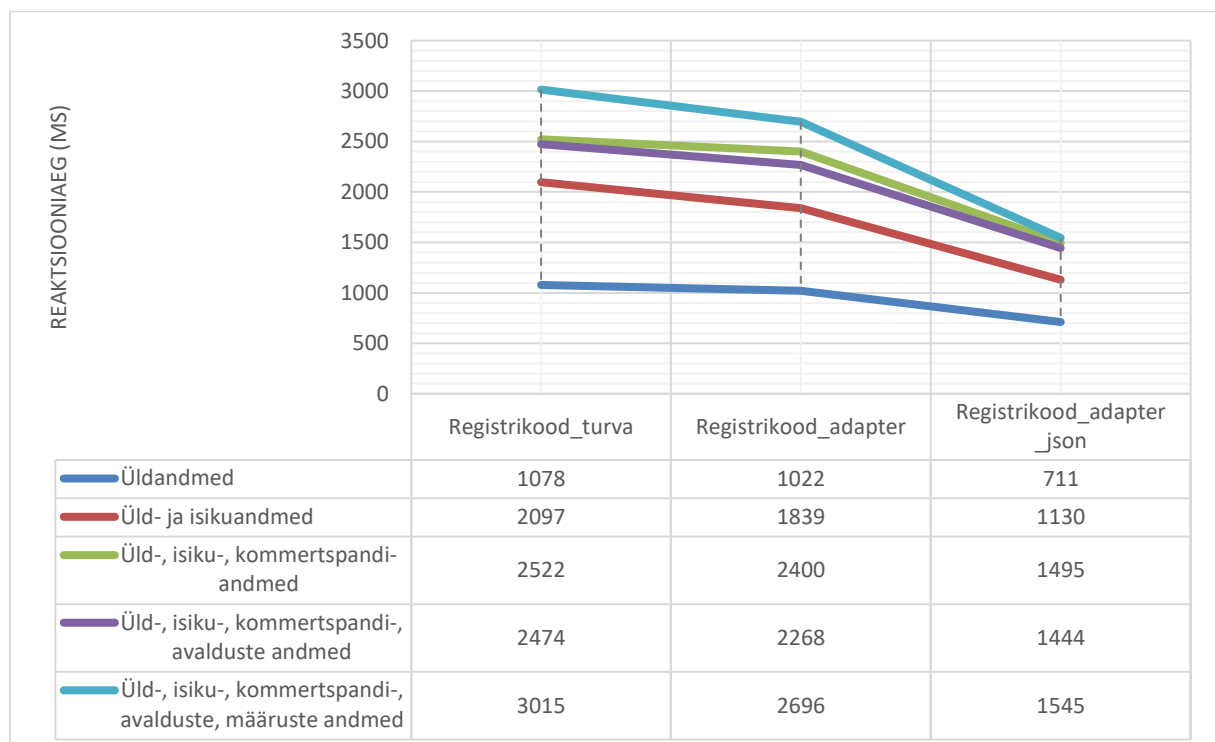
### **5.2.5 Testide käivitamise ja koormuse tekitamise strateegia**

Kuna analüüsis uuritakse lähemalt, kui pikk on teenuse reaktsiooniaeg ning kui suur koormus on infrastruktuuril äriregistri teenuste erinevate sisendite korral, siis oleks mõistlik valida võimalikult lihtne koormuse profiil, et mõõtmistulemuste hindamine oleks lihtne. Selleks sobivaim lahendus on kogu plaanitud infosüsteemi koormamine algab ja lõpeb samaaegselt. Kasutajad teevad toiminguid korraga, kuid mitte sünkroonselt.

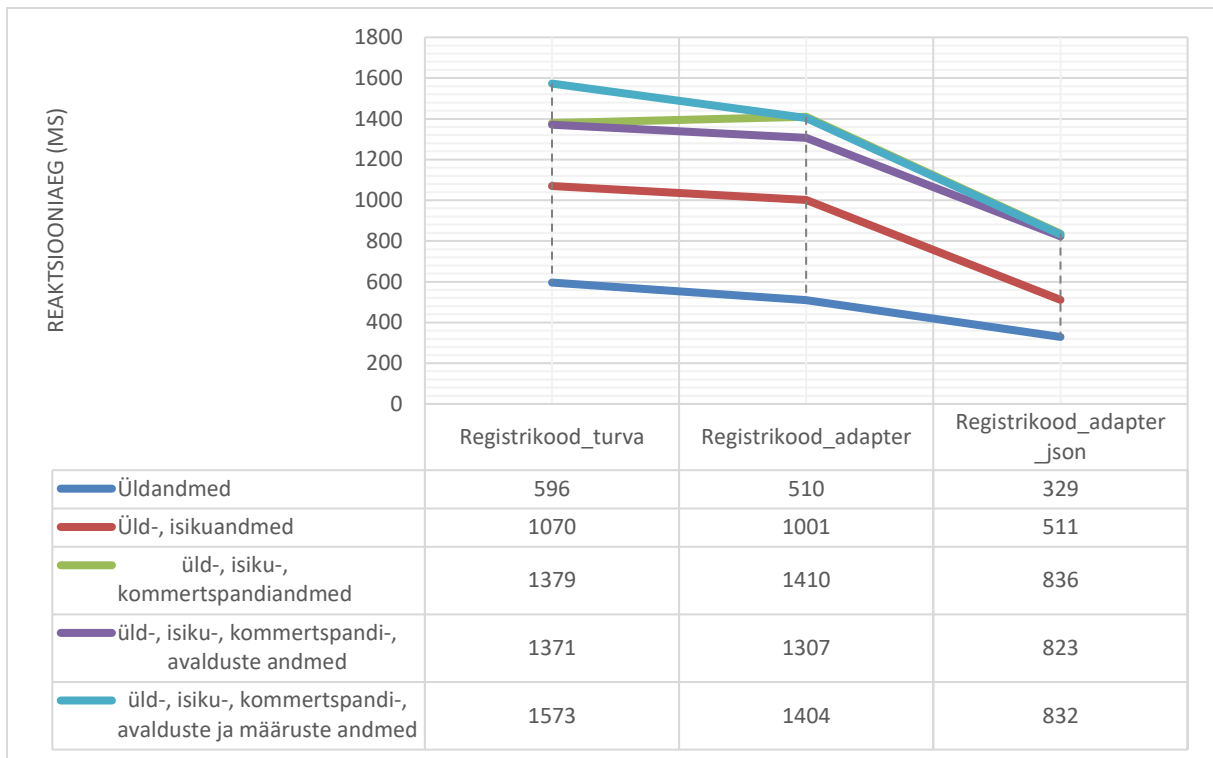
Kuna testkeskkonna infrastruktuur erineb töökeskkonna infrastruktuurist tunduvalt, siis koormuse alusmäära testimist ei tehtud. Indikaatoriks võeti vaid üksikute päringute keskmine reaktsiooniaeg.

### 5.3 Päringu reaktsiooniaja mõõtmine erinevate sisendite korral

Teenuse detailandmed\_v3 näitel valiti esimeseks testietapiks päringu registrikoodi järgi. Sama teenust kasutades saab pärida viite erinevat andmekomplekti: üldandmed, isikuandmed, kommertspandi andmed, menetluses avalduste andmed ja määruste andmed. Ühe sisendiga päringut käivitati 25 lõimega 600 sekundi jooksul. Esmalt tehti test üldandmete kohta ning seejärel lisati iga testiga uus päritav andmekomplekt juurde päringule. Kõikide päringute korral tehti neid ühes testietapis 3 erineval viisil: päring turvaserverisse, päring adapterserverisse, päring adapterserverisse ning vastuse tagastamisega JSON vormingus. Mõõdetavateks parameetriteks olid minimaalne teenuse reaktsiooniaeg, maksimaalne teenuse reaktsiooniaeg, keskmine teenuse reaktsiooniaeg ja edastatud andmehulk ajavahemikus – baiti sekundis. Mõõdetud tulemuste visuaalseks analüüsiks koostati järgnevad joonised.



Joonis 2. Päringute keskmine reaktsiooniaeg.



Joonis 3. Päringute minimaalne reaktsiooniaeg.



Joonis 4. Päringute maksimaalne reaktsiooniaeg.



Joonis 5. Keskmine päringutega edastatud andmete hulk(baiti sekundis).

## 5.4 Kasutajate arvu kasvu mõju infrastruktuurile

Teises testietapis analüüsis autor teenuse detailandmed\_v3 näitel, millised infrastruktuuri komponendid on enim mõjutatud teenuste poolt tekitatud koormusest. Ühe sisendiga päringut käivitati erinevates testides vastavalt 10, 20 ja 30 lõimega 600 sekundi jooksul. Päring suunati turvaserverisse, et koormus tekiks kõikides põhilistes komponentides: X-tee turvaserver, andmebaasi adapterserver ja andmebaasiserver. Kõik jälgitavad infrastruktuuri komponendid olid klasterdamata ühe serveriga süsteemid.

Turvaserveri tehnilised näitajad:

- 2.66 GHz Intel Xeon protsessori 1 tuum
- 12 GB muutmälu
- Võrguadapteri kiirus: 1000Mb/s

Adapterserveri tehnilised näitajad:

- 2.66 GHz Intel Xeon 1 protsessori tuum

- 11,5 GB muutmälu
- Võrguadapteri kiirus: 1000Mb/s

Andmebaasiserveri tehnilised näitajad:

- 2.66 GHz Intel Xeon 1 protsessori tuum
- 5,5 GB muutmälu
- Võrguadapteri kiirus: 1000Mb/s

Turvaserveri mõõdetavad väärtused:

- Keskprotsessori koormus
- Muutmälu kasutus

Adapterserveri mõõdetavad väärtused:

- Keskprotsessori koormus
- Muutmälu kasutus

Andmebaasiserveri mõõdetavad väärtused:

- Keskprotsessori koormus
- Muutmälu kasutus

Võrgukoormust otsustas autor mitte täpsemalt mõõta, sest arvestades tänapäevaseid võrgukiirusi ning esmaseid katsetulemusi, kus võrgukiirusi jälgiti, leidis autor, et XML päringute tekitatav koormus võrgule on nii madal, et probleeme see tekitada ei tohiks.

Tabel 1. Detailandmed\_v3 päringu tekitatud koormused 10 löimega testides.

	CPU koormatus koormuseta	CPU koormatus koormuse korral	CPU kasutus koormuse (idle time)	CPU kasutus koormuse all	Vaba mälu maht koormusega	Vaba mälu maht koormuse all
Turvaserver	7 %	33 %	94 %	34 %	3,48 GB	3,12 GB

Adapterserver	0,2 %	51 %	99 %	38 %	8,71 GB	8,68 GB
Andmebaasiserver	4 %	61 %	91 %	25 %	4,4 GB	4,37 GB

Tabel 2. Detailandmed\_v3 päringu tekitatud koormused 20 löimega testides.

	CPU koormatus koormuse ta	CPU koormatus koormuse all	CPU kasutus koormuseta(i dle time)	CPU kasutus koormuse ga	Vaba mälu maht koormuse ta	Vaba mälu maht koormuse all
Turvaserver	7 %	62 %	93,4 %	24,7 %	3,12 GB	3,10 GB
Adapterserver	0,4%	59,5 %	99,4 %	25,6 %	7,73 GB	7,53 GB
Andmebaasiserver	4 %	65,5%	91%	22,1 %	4,40 GB	4,34 GB

Tabel 3. Detailandmed\_v3 päringu tekitatud koormused 30 löimega testides.

	CPU koormatus koormuse ta	CPU koormatus koormuse all	CPU kasutus koormuseta(id le time)	CPU kasutus koormuse all	Vaba mälu maht koormuse ta	Vaba mälu maht koormuse all
Turvaserver	7 %	58 %	94 %	23 %	3,38 GB	2,87 GB
Adapterserver	0,2 %	67 %	99 %	18%	7,57 GB	7,37 GB
Andmebaasiserver	4 %	68 %	95 %	20%	4,41 GB	4,33 GB



## 5.5 Kasutajate arvu kasvu mõju teenuse reaktsioonijale

Punktis 5.4 testiti teenuse reaktsiooniaegu 10 lõimega pärides. Nüüd testitakse, kuidas muutub reaktsiooniaeg, kui kasutajate arv kasvab. Testiti teenust detailandmed\_v3, otsides ettevõtja üldandmeid ettevõtja registrikoodi järgi. Lõimede arvu kasvatati iga testiga kümne võrra: 10 lõime, 20 lõime, 30 lõime. Alumistes tabelites on esitatud testide tulemused.

Tabel 4. 10 lõimega testides Detailandmed\_v3 päringu mõju teenuse reaktsioonijale.

Testilugu	min	max	avg	bps
Registrikood_turva	596	4297	1078	172268
Registrikood_adapter	510	2325	1022	169239
Registrikood_adapter_json	329	1318	711	116353

Tabel 5. 20 lõimega testides Detailandmed\_v3 päringu mõju teenuse reaktsioonijale.

Testilugu	min	max	avg	bps
Registrikood_turva	759	6015	1728	198152
Registrikood_adapter	578	3216	1823	194668
Registrikood_adapter_json	392	2434	1461	133836

Tabel 6. 30 lõimega testides Detailandmed\_v3 päringu mõju teenuse reaktsioonijale.

Testilugu	min	max	avg	bps
Registrikood_turva	957	13773	2601	193856
Registrikood_adapter	1050	5122	2745	190448

Registrikood_adapter_json	330	3339	2358	130934
---------------------------	-----	------	------	--------

Kasutajate arvu kahekordistumisel, ehk siis võrreldes 10 löimega testi 20 löimega testiga kasvas keskmine päringu aeg turvaserverisse 650 ms (60%), adapterserverisse 801 ms (78%) ja adapterserverise JSON vormingus vastusega 750 ms (105%). Võrreldes 20 löimega testi 30 löimega testiga kasvas keskmine päringu aeg turvaserverisse 873 ms (50 %), adapterserverisse 922 ms (50 %), adapterserverisse JSON vormingus vastusega 897 ms (61 %).

## 5.6 Testimistulemuste võrdlus

Analüüsid detailandmete teenuse sisendeid, teenuse reaktsioonikiiruseid, infrastruktuuri koormuseid ning kasutajate arvu kasvu mõju, saab teha järgmise kokkuvõtte leitud erinevustest.

### 5.6.1 Teenuse reaktsioonikiirus

Tulenevalt Joonis 2. Päringute keskmine reaktsiooniaeg, esitatud andmetest saab järeldada, et päringute tegemisel turvaserveri ja adapteri vahel kasutajamugavuses suurt vahet ei teki. Kasutusmugavus kasutajal ei muutu, arvestades, et keskmine päringute reaktsiooniaegade vahe oli erinevate sisendite korral vastavalt testile kas 56, 258, 122, 206 või 319 ms adapteri päringust kiirem. Samas aga kui adapteri päringu vastus tagastatakse JSON vormingus, siis on päringu keskmine reaktsiooniaeg vastavalt testile kas 311, 709, 905, 824 või 1151 ms kiirem, kui XML formaadis vastus.

### 5.6.2 Päringuga edastavate andmete maht

Päringuga edastavate andmete maht ning sekundis edastavate andmete maht oli kõikides testides suurim läbi X-tee turvaserveri. XML päring, mis tehti adapterserverisse ning mille vastus oli samuti XML oli erinevates testides vastavalt 3029, 1678, 1265, 1357, 1155 baiti sekundis väiksema mahuga, kui turvaserverisse tehtud päringud. Adapterserverisse tehtud päring, mille vastus tagastati JSON vormingus, oli adapterserveri XML vastustest vastavalt 52886, 6326, 17899, 19743, 31514 baiti sekundis väiksema andmete mahuga.

### **5.6.3 Teenuse sisendi erinevused**

Teenuse sisendis lisati iga testiga uusi päritavaid andmeid, mis pikendas minimaalseid, keskmisi ja maksimaalseid reaktsiooniaegasid. Iga sisendi ning oodatava väljundi lisamisega suurendati andmebaasipäringut, mis selle põhjal koostati ning vastuse koostamine võttis kauem aega. Samuti kasvas üle võrgu saadetakse andmete hulk.

### **5.6.4 Kasutajate arvu kasv**

Kasutajate arvu kasvades kasvas vastavalt ka teenuse reaktsiooniaeg, edastatud andmehulk ning infrastruktuuri koormused. Tõstes kasutajate arvu kaks korda, testides 10 lõimega kasutajalt 20 lõimega kasutajale, kasvas keskmine teenuse reaktsiooniaeg erinevates testides vastavalt 650, 801 ja 750 ms. Ajas edastatud andmehulk vastavalt tõusis 25884, 25429, 17483 baiti sekundis. Tõstes kasutajate arvu veel poole võõra, ehk 30 lõimeni, kasvas keskmine teenuse reaktsiooniaeg vastavalt erinevates testides 873, 922 ja 897 ms. Samal ajal aga ajas edastatud andmehulk (baiti sekundis) natuke langes: -4296, -4220, -2902. Vaadeldavas testis oli langus nii väike, et ei saa öelda kindlalt, et toimus suur langus. Pigem saab väita, et edasist tõusu ei toimunud ning andmeid suudeti sama aja jooksul sama palju edastada.

### **5.6.5 Infrastruktuuri mõõtmistulemused**

Andmebaasiserveri komponent oli jälgitavatest infrastruktuuri komponentidest kõige probleemsem. Nimelt kasvas keskprotsessori kasutus ja koormus kõige kiiremini. Keskprotsessori koormatus kasvas 10 lõimega testis 61%, 20 lõimega 65% ja 30 lõimega 68%. Keskprotsessori kasutus kasvas 10 lõimega testis 66%, 20 lõimega testis 69% ja 30 lõimega testis 75%. Muutmälu kasutus kõikides testides oli marginaalne. Kõikides testides vähenes vaba mälu alla 0,1 GB mistõttu piisava varu korral ei tohiks probleeme tekkida.

Adapterserveri komponendi probleemiks oli sarnaselt andmebaasiserverile kõrge keskprotsessori koormus. Adapterserveri keskprotsessori koormatus kasvas 10 lõimega testis 51%, 20 lõimega 59% ja 30 lõimega 67%. Keskprotsessori kasutus kasvas 10 lõimega testis 61%, 20 lõimega testis 74% ja 30 lõimega testis 81%. Muutmälu kasutus kõikides testides oli marginaalne. Kõikides testides vähenes vaba mälu alla 0,3 GB mistõttu piisava varu korral ei tohiks probleeme tekkida.

Turvaserveri komponendi probleemiks oli sarnaselt teistele komponentidele kõrge keskprotsessori koormus. Adapterserveri keskprotsessori koormatus kasvas 10 lõimega testis

51%, 20 lõimega 59% ja 30 lõimega 67%. Keskprotsessori kasutus kasvas 10 lõimega testis 61%, 20 lõimega testis 74% ja 30 lõimega testis 81%. Muutmälu kasutus kõikides testides oli marginaalne. Kõikides testides vähenes vaba mälu alla 0,3 GB mistõttu piisava varu korral ei tohiks probleeme tekkida.

## 5.7 Teenuste analüüs ning soovitused

Analüüsidest punktides 5.3-5.5 kogutud infot detailandmed\_v3 ja teiste teenuste koormustestide kohta võib teha järgmisi järeldusi äriregistri X-tee teenuse kohta.

1. Andmebaasi päringute optimeerimine. Tähtis osa päringute vastustes on andmebaasipäringute kiirusel. Kui andmebaasipäring on aeganõudev, siis pole teistest teenuste optimeerimise meetmetest eriti kasu ning teenus jääb ikkagi kasutajale aeglaseks ning infrastruktuurile ebamõistlikult koormavaks. Seega on üks kindlasti kriitilisi komponente, mida jälgida teenuste optimeerimise juures andmebaasi päringud. Analüüsitud teenustest leiti teenuses mtaBilSisu\_v1, et päring oli ebamõistlikult aeglane ning kasutajatel tekkisid probleemid teatud sisendite korral. Analüüsidest teenust mtaBilSisu\_v1 leiti, et probleem oli andmebaasipäringus. Peale paranduste tegemist andmebaasipäringus vähenes teenuse reaktsiooniaeg teatud sisendite korral üle 10 sekundi.

2. XML asemel JSON vormingus päringute kasutamine. Punktis 5.3 esitatud tabelites ja joonistel tuli selgelt välja, et kasutaja vaatest on JSON vormingus päringute kasutamine efektiivsem. Nimelt võrreldes XML päringuga olid JSON vastusega päringud teenuse keskmises reaktsiooniajas keskmiselt 300-1100 ms kiiremad. Kusjuures mida rohkem edastati päringuga infot, seda suurem oli ka vahe JSON vastusega päringu kasuks. Samuti oli edastatud andmemahud väiksemad, mis eeldatavasti panid väiksema koormuse infrastruktuurile. Selle tõttu on soovituslik XML päringute asemel kasutada REST teenuseid. Alates 2019 aprillist on võimalik X-tee teenuseid arendada ka täielikult REST APIle, kui avaldati uus X-tee funktsionaalsus. REST APIga X-tee teenuste kasutamine vähendab teenuste reaktsiooniaega ning vähendab ka sõnumivahetuseks kulutatud ressursse. [13]

3. Teenuse vastuse lehekülgedeks jagamine. Analüüsidest teenuste sisendeid ja väljundeid tekkis selge probleem teenustega, millel puudus lehekülgede tugi. Analüüsitud teenustest oli selleks rkoarr\_asutused\_v1. Kui teenuses vastuses on vähe andmeid, näiteks vaid paari ettevõtja kohta, siis probleeme ei teki. Kui aga vastusega võib tulla sadu otsinguvasteid, siis selle

koostamine võib võtta liiga kaua aega ning kasutada liiga palju süsteemide mälu. Lahendus siinkohal oli teenuse rkoarr\_asutused\_v1 vastuse jagamine lehekülgedeks.

4. Kriitilisemate infrastruktuuri komponentide täiendamine, et jõudlusprobleemid nii kergelt ei tekiks. Põhiliselt peaks jälgima komponente, mille koormus kasvab kõige kiiremini, ehk siis vaadeldavas töös on selleks põhiliselt andmebaasiserveri, turvaserveri ja adapterserveri keskprotsessori võimekus. Kui andmebaasi serveri mälukasutuses tekib probleeme, siis võib sellest järeldada, et päringu andmebaasipäringu optimeerituses on probleeme.

## **5.8 Edaspidised võimalused süsteemi koormuse vähendamiseks**

Infrastruktuuri jõudluse täiendamine: näiteks võib infrastruktuuri komponent olla vananenud tehnikaga või mõnel muul põhjusel vähese ressursiga jäänud. Sellisel juhul on lihtsaim ja odavaim võimalus infrastruktuuri komponendi tehnoloogiline uuendamine või ressursi lisamine. [7]

Koormuse tasakaalustamine: infosüsteemi koormuse tasakaalustamine on levinud vahend, mis aitab koormust jaotada infrastruktuuris mitme serveri vahel. Tänu sellele väheneb võimalus, et teenused pole kätte saadavad täna serveri jõudlusprobleemidele või uude tehniliste probleemide tõttu. Samuti on koormuse ajal ka teenuste reaktsiooniaeg lühem. Koormuse tasakaalustamine on üks levinumaid võimalusi, kuidas tagada teenuste kättesaadavus ning piisavalt lühike teenuste reaktsiooniaeg. [14]

Tuleb analüüsida teenuste sisu. Võimalik, et teenusega väljastatakse andmeid, mis on ebavajalikud ning nende esitamiseks läheb liialt suur ressurss. Seega võiks vaadata üle oma teenuseportfelli ning ebavajalikud andmeväljad teenusest eemaldada. Näiteks leiti teenuse toimikuDokument\_v1 väljundit analüüsid, et teenuse vastusega antakse kaasa ka faili suurus baitides. Faili suuruse arvutamine võttis aega ning mäluresurssi, kuid teenuse kasutajad andmevälja ei kasutanud. Seega võis andmevälja eemaldada ning sellega vastas teenus kiiremini ning samuti vähenes mälukasutus.

Kasutajate juhendamine - ajakohased ning täpsed kasutusjuhendid. Süsteemide koormus võib tõusta ka kasutajate vähesest informeeritusest. Kasutajatel peavad olema ajakohased juhendid teenuste kohta, et nad teaksid, milliseid teenuseid soovitud andmete jaoks kasutada ning et kasutajate ärioloogika töötaks korrektselt.

## KOKKUVÕTE

Teenuste koormustestimise juures on keerulisem osa meetodika valik, millisel viisil probleemile läheneda. Vaadeldavas töös valis autor meetodika koormustestide loomiseks vastavalt töös viidatud autorite kogemustele. Kuid kindlasti saab teenuste ja infosüsteemi komponentide tööd koormuse all jälgida ka teisi meetodikaid kasutades. Autor leidis ka samade autorite poolt mitmeid teisi võimalusi, kuidas teenuseid jälgida, kuid antud meetodika paistis sobivamat kõige rohkematele teenustele ning on ka lihtsam teostada ja analüüsida.

Töös analüüsiti viite kõige populaarsemat ja viite kõige pikema reaktsioonijaga äriregistri teenust ning X-tee teenuste töös koormatavaid infosüsteemi komponente. Samuti analüüsiti, kuidas muutus teenuste reaktsiooniaeg erinevate süsteemi koormuste korral ning teenuste erinevate sisendi ja väljundi parameetrite muutumisel. Tulenevalt testide tulemustest leiti, et teenuse reaktsioonijale on kõige suurem mõju andmebaasi päringute efektiivsusel. Kui andmebaasi päringud võtavad kaua aega, siis pole teistel infosüsteemi komponentide efektiivsusel suurt väärtust. Seega peaks teenuste töö parendamisel suurt tähelepanu pöörama andmebaasi päringu optimeerimisele: näiteks suudeti teenust optimeerida nii, et teatud sisendi korral vähenes päringu reaktsiooniaeg rohkem kui 10 sekundit.

Autor leidis lisaks ka teisi võimalusi teenuste töö optimeerimiseks. Üks kindlaim viise, kuidas vähendada teenuse reaktsiooniaeg oli XML teenuste asemel REST teenuste kasutamine, mida osaliselt näitasid ka katsetulemused. Teiseks suuremaks faktoriks, mis võiks teenuste tööd mõjutada oli teenuse vastuse lehekülgedeks jagamine. Kui seda ei tehtud, siis suuremahuliste vastuste korral võis reaktsiooniaeg olla mitu minutit.

Infrastruktuuri komponentide koormamisel tekkisid suurimad probleemid andmebaasiserveril. Turvaserveri ja adapterserveri jälgitud omadustel nii suuri muudatusi ei täheldatud. Seegi viitab andmebaasipäringute tähtsale osale teenuse töös.

Käesolevas töös kasutatud meetodikat saab rakendada ka teiste X-teega liidestunud infosüsteemide poolt pakutavate teenuste jõudluse testimiseks. Üldised reeglid, mida jälgida teenuste ja infosüsteemide komponentide töös on laiendatavad ka teistele infosüsteemidele.

Järgmisteks sammudeks X-tee teenuste optimeerimisel on andmebaasipäringute optimeerimise analüüs ning tulevikus, kui X-tee toetab, siis ka REST teenuste kasutusele võtmine.

## KASUTATUD KIRJANDUS

- [1] Äriregistri XML teenused. [WWW] <https://www.rik.ee/et/e-ariregister/xml-teenus> (11.09.2019)
- [2] Äri- ja ühinguregister. [WWW] <https://www.just.ee/et/eesmargid-tegevused/ari-ja-uhinguregister> (11.09.2019)
- [3] X-tee teenuste faktiliht, [WWW] <https://www.x-tee.ee/factsheets/EE/> (25.02.2020)
- [4] Äriregistri XML päringute detailne kirjeldus. [WWW] [https://www.rik.ee/sites/www.rik.ee/files/elfinder/article\\_files/xml\\_paringute\\_detailne\\_kirjeldus.pdf](https://www.rik.ee/sites/www.rik.ee/files/elfinder/article_files/xml_paringute_detailne_kirjeldus.pdf) (12.04.2020)
- [5] X-tee andmeteenuste arendajate koolitusmaterjalid. [WWW] <https://moodle.ria.ee/course/view.php?id=14> (10.01.2020)
- [6] Glenford J. Myers, Tom Badgett, Corey Sandler. The Art of Software Testing. 3rd ed. USA: John Wiley & Sons, Inc, 2012.
- [7] Bernhard Klemm. Application Performance Testing: A Universal Performance Testing Methodology. Suurbritannia : Bernhard Klemm, 2013.
- [8] Ian Molyneaux. The Art of Application Performance Testing: Help for Programmers and Quality Assurance. USA : O'Reilly Media, Inc, 2009.
- [9] Henry H. Liu. Software Performance and Scalability: A Quantitative Approach. USA : 2009.
- [10] Apache JMeter. [WWW] <https://jmeter.apache.org/index.html> (28.09.2019)
- [11] SoapUI. [WWW] <https://www.soapui.org/open-source.html> (28.09.2019)
- [12] Rahvastikuregistri x-tee teenuste testimine arenduskeskkonnas Jmeter ja SoapUI näitel, [WWW] <https://digi.lib.ttu.ee/i/?8839> (12.04.2020)
- [13] X-tee blogi, [WWW] <https://blog.ria.ee/tag/x-tee/> (17.03.2020)

[14] IBM Cloud Learn Hub, [WWW] <https://www.ibm.com/cloud/learn/load-balancing>  
(12.02.2020)



# LISAD

## Lisa 1. Detailandmed\_v3 teenuse päring turvaserverisse.

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xro="http://x-road.eu/xsd/xroad.xsd"      xmlns:iden="http://x-
road.eu/xsd/identifiers"                      xmlns:prod="http://arireg.x-
road.eu/producer/">

  <soapenv:Header>

    <xro:protocolVersion>4.0</xro:protocolVersion>

    <xro:userId>EE11111111111</xro:userId>

    <xro:id>1234567890</xro:id>

    <xro:service iden:objectType="SERVICE">

      <iden:xRoadInstance>ee-test</iden:xRoadInstance>

      <iden:memberClass>GOV</iden:memberClass>

      <iden:memberCode>70000310</iden:memberCode>

      <iden:subsystemCode>arireg</iden:subsystemCode>

      <iden:serviceCode>detailandmed_v3</iden:serviceCode>

      <iden:serviceVersion>v1</iden:serviceVersion>

    </xro:service>

    <xro:client iden:objectType="SUBSYSTEM">

      <iden:xRoadInstance>ee-test</iden:xRoadInstance>

      <iden:memberClass>GOV</iden:memberClass>

      <iden:memberCode>70000310</iden:memberCode>

      <iden:subsystemCode>generic-consumer</iden:subsystemCode>

    </xro:client>

  </soapenv:Header>

  <soapenv:Body>
```

```

<prod:detailandmed_v3>
  <prod:keha>
    <prod:ariregistri_kood>70000310</prod:ariregistri_kood>
    <prod:yandmed>1</prod:yandmed>
    <prod:iandmed>0</prod:iandmed>
    <prod:kandmed>0</prod:kandmed>
    <prod:dandmed>0</prod:dandmed>
    <prod:maarused>0</prod:maarused>
  </prod:keha>
</prod:detailandmed_v3>
</soapenv:Body>
</soapenv:Envelope>

```

## Lisa 2. Detailandmed\_v3 teenuse X-tee osaline näidisvastus.

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xro="http://x-road.eu/xsd/xroad.xsd"          xmlns:iden="http://x-
road.eu/xsd/identifiers"                          xmlns:prod="http://arireg.x-
road.eu/producer/">
  <soapenv:Header>
    <xro:protocolVersion>4.0</xro:protocolVersion>
    <xro:userId>EE111111111111</xro:userId>
    <xro:id>1234567890</xro:id>
    <xro:service iden:objectType="SERVICE">
      <iden:xRoadInstance>ee-test</iden:xRoadInstance>
      <iden:memberClass>GOV</iden:memberClass>
      <iden:memberCode>70000310</iden:memberCode>
      <iden:subsystemCode>arireg</iden:subsystemCode>
      <iden:serviceCode>detailandmed_v3</iden:serviceCode>
    </xro:service>
  </soapenv:Header>

```

```

    <iden:serviceVersion>v1</iden:serviceVersion>
  </xro:service>
  <xro:client iden:objectType="SUBSYSTEM">
    <iden:xRoadInstance>ee-test</iden:xRoadInstance>
    <iden:memberClass>GOV</iden:memberClass>
    <iden:memberCode>70000310</iden:memberCode>
    <iden:subsystemCode>generic-consumer</iden:subsystemCode>
  </xro:client>
</soapenv:Header>
<SOAP-ENV:Body xmlns:ns1="http://arireg.x-road.eu/producer/">
  <ns1:detailandmed_v3Response>
    <ns1:paring>
      <ns1:ariregistri_kood>70000310</ns1:ariregistri_kood>
      <ns1:yandmed>true</ns1:yandmed>
      <ns1:iandmed>false</ns1:iandmed>
      <ns1:kandmed>false</ns1:kandmed>
      <ns1:dandmed>false</ns1:dandmed>
      <ns1:maarused>false</ns1:maarused>
    </ns1:paring>
    <ns1:keha>
      <ns1:ettevotjad>
        <ns1:item>
          <ns1:ariregistri_kood>70000310</ns1:ariregistri_kood>
          <ns1:ettevotja_id>9000176081</ns1:ettevotja_id>
          <ns1:nimi>Registrite ja Infosüsteemide Keskus</ns1:nimi>
          <ns1:kmkkr_number>EE100523377</ns1:kmkkr_number>
        </ns1:item>
      </ns1:ettevotjad>
    </ns1:keha>
  </ns1:detailandmed_v3Response>
</SOAP-ENV:Body>
</soapenv:Body>

```

```

    <ns1:yldandmed>
      <ns1:esmaregistreerimise_kpv>1998-09-
23Z</ns1:esmaregistreerimise_kpv>
      <ns1:staatus>R</ns1:staatus>
      <ns1:staatus_tekstina>Registrisse
kantud</ns1:staatus_tekstina>
      <ns1:piirkond>5</ns1:piirkond>
      ...
    </ns1:keha>
  </ns1:detailandmed_v3Response>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

### **Lisa 3. Detailandmed\_v3 teenuse osaline näidisvastus JSON vormingus.**

```

{"keha":
  {"ettevotjad":
    {"item":
      [{"ariregistri_kood": 70000310,
        "ettevotja_id": 9000176081,
        "nimi": "Registrite ja Infos\u00f0fcsteemide Keskus",
        "isikuandmed": {}},
        {"kmkr_number": "EE100523377"},
      ...
    ]
  }
}

```