TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Risto Põldsalu 176963IAPM

EHITISREGISTRI ARHITEKTUURI VIIMINE MIKROTEENUSTE ARHITEKTUURILE

magistritöö

Juhendaja: Ants Torim
PhD

Tallinn 2020

Autorideklaratsioon

Kinnitan, et olen koostanud antud 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: Risto Põldsalu

07.01.2020

Annotatsioon

Käesoleva töö eesmärk on alustada Ehitisregistri arhitektuuri üle viimist mikroteenustel põhinevale arhitektuurile. Selle jaoks luuakse lõputöö käigus pilootrakendus, millega asendatakse osa senisest süsteemist ja see paigaldatakse uuele arhitektuurile. Seejärel leitakse kriteeriumid, et hinnata valitud teenusevõrgustikke ning leida sobiv teenusevõrgustik uue arhitektuuri jaoks.

Töö esimeses osas kirjeldatakse pilootrakendust ja senist süsteemi. Pilootrakenduse paigalduse järel tehakse järeldused, mis annavad aluse teenusevõrgustiku valimiseks.

Teises osas leitakse ja valitakse välja olulisemad kriteeriumid, mille järgi valida teenusevõrgustikku. Nende kriteeriumite järgi hinnatakse autori poolt ära 4 erineva teenusevõrgustiku varianti.

Kolmandas osas koostatakse küsitlus, mille käigus kriteeriumitele antakse kaalud, võrreldes neid omavahel tähtsuse järgi skaalal 1 – 9 punkti. Leitud kaaludega arvutatakse läbi eelnevalt tehtud hinnangud ning leitakse sobivaim teenusevõrgustik.

Lõputöö tulemuseks on Ehitisregistris kasutatav uuel arhitektuuril põhinev rakendus ning hindamismeetod koos näitliku lahendusega teenusevõrgustiku valimiseks.

Lõputöö on kirjutatud Eesti keeles ning sisaldab teksti 37 leheküljel, 7 peatükki, 3 joonist, 20 tabelit.

Abstract

Taking building registry architecture to microservices architecture

This thesis mission is to build the first pilot application for the Building registrys new microservice based architecture and with retrospection from that find the correct criteria to evaluate service meshes. The aim is to give the chosen criteria weights that can be used for evaluating service meshes.

In the first part of the thesis the current state of the system will be explained and the development of the new application. Conclusions will be made from this and they will be used to start the second part of the thesis.

In the second part of the thesis the important criteria will be found by using the Architecture Tradeoff Analysis Method (ATAM). Specialist will find scenarios that are important for them in the service mesh view and through evaluation and feedback the most important ones will be chosen. With these criterias 4 different solutions will be graded by each category so later the one with the best score can be found out.

In the last part of the thesis the weighted criteria will be used to evaluate the results for the possible solutions that were previously graded by the same criteria. The results can be used in the Building Registrys development and the weighted criteria method can be taken and used to evaluate a different project with a similar aim to find a solution from many possibilities.

The thesis is in Estonian and contains 37 pages of text, 7 chapters, 3 figures, 20 tables.

Lühendite ja mõistete sõnastik

AHP	*Analytical hierarchy process*, Analüütiline otsustusprotsess
mTLS	mutual Transport Layer Security
TLS	Transport Layer Security
CLI	Command-line interface
ATAM	Architecture Tradeoff Analysis Method
VRM	Võrgu Rajatiste moodul
EHR	Ehitisregister
DNS	Domain Name System
MKM	Majandus- ja Kommunikatsiooniministeerium
MTA	Maksu- ja Tolliamet

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Mikroteenuste arhitektuur [1] on kinnitanud oma koha tarkvara maailmas. Paljud ettevõtted on mikroteenuste arhitektuuri peale üle kolinud juba mitu aastat tagasi.

Käesoleva magistritöö eesmärgiks on leida Ehitisregistri portaali uue mikroteenuste arhitektuuri lahendus koos pilootrakendusega arendusega. Pilootrakendus on reaalne osa olemasolevast Ehitisregistri süsteemist, mis lõputöö käigus asendatakse uuega ning integreeritakse olemasolevasse süsteemi. Pilootrakenduse pealt tehakse järeldused ja leitakse koos spetsialistide abiga kriteeriumid, mille järgi viiakse läbi otsustusprotsess teenusevõrgustiku [2] valimiseks.

Töö esimeses pooles kirjeldatakse lahti olemasolev süsteem ja pilootrakenduse eesmärk ning lahendus. Tehakse järeldused pilootrakenduse arendamise järel. Teises töö pooles leitakse riigi poolt juhitud mikroteenuste arhitektuurile sobiv teenusevõrgustik läbi kriteeriumite hindamise. Esmalt leitakse koostöös spetsialistidega sobivad kriteeriumid, mille järgi hinnata teenusevõrgustiku.

Kriteeriumite leidmiseks kasutatakse esialgu Architecture Tradeoff Analysis Method(ATAM) meetodist [3] osalist lahendust , et täpsustada arhitektuuri vajadus ja nõuded läbi stsenaariumite. Kriteeriumite põhjal hinnatakse iga teenusevõrgustiku varianti autori poolt [4] ja seejärel tehakse küsitlus Analytical Hierarchy Process(AHP) otsustusprotsessi [5] kasutades, et leida osakaalud kriteeriumitele.

Seejärel tehakse läbi otsustusprotsess tulemuste pealt, kus valitakse sobiv teenusevõrgustik. Leitud kriteeriume on võimalik kasutada sarnast protsessi läbides teistes riigi süsteemide projektides.

Valikusse on võetud 4 erinevat teenusevõrgustikku:

- Istio
- Consul
- Linkerd2
- Maesh

Eelduseks on, et Istio osutub kõige sobivamaks teenusevõrustikuks. Istio on kõige laiemalt tuntud ning on väga muutmisvõimeline. Istiot toetavad mitmed suured ettevõtted ning see on paindlik kasutusele võtmise poolest.

2 Taust ja eesmärgi püstitus

Selles peatükis kirjeldan magistritöö tausta ning püstitan eesmärgi. Seletan lahti probleemi seoses uuele arhitektuurile üle minekuga ning tutvustan võimalikke lahendusi ja toon välja metoodikad, mida kasutada võrdluseks.

2.1 Teenusevõrgustik

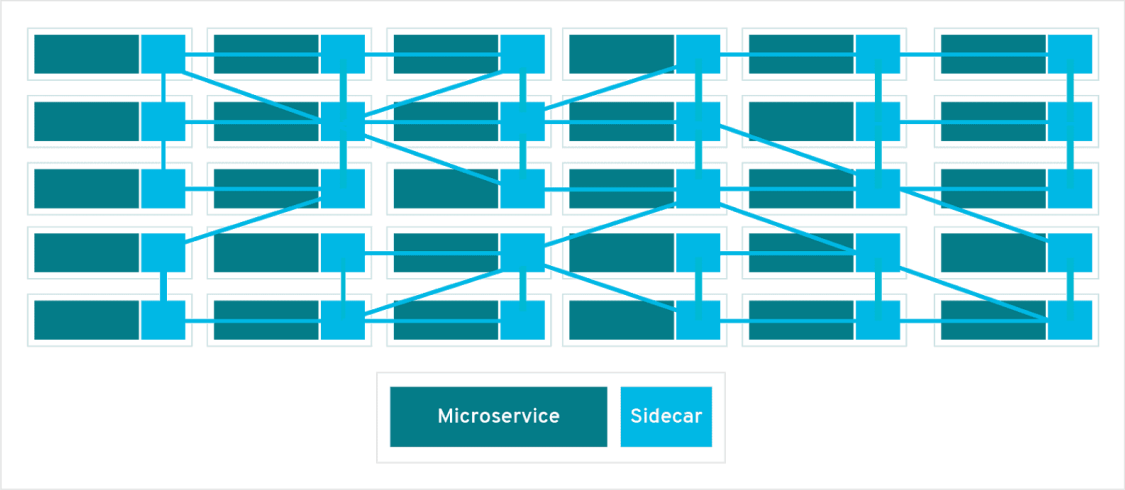
Teenusevõrgustik (*Service mesh*) tähendab tarkvara arhitektuuri kontekstis teenuste võrgustikku, mis liidab ühtse halduse alla ühe süsteemi erinevad osad. See arhitektuur on loodud spetsiaalselt konteineripõhiste mikroteenuste juhtimiseks. Teenusevõrgustiku eesmärgiks on käsitleda ja manageerida, tegeleda kompleksuse ja omavahelise suhtluse lahendamisega paljude mikroteenustega arhitektuurides [6]. Võrgustik võimaldab piirata nende ressursside ligipääsetavust. Teenusevõrgustik võimaldab jälgida ja juhtida teenuseid, mis kuuluvad võrgustikku.

Teenusevõrgustik koosneb kahest kihist. Kontrollkiht (control plane) ja andmekiht (data plane). Andmekihiks nimetatakse seda osa süsteemist, mis on otseselt seotud iga süsteemi osaga ning mis tegeleb võrgustikus suhtluse vastuvõtmisega ja teenusele edasi suunamisega ja päringute kontrollimisega. Need on proksi komponendid, mida paigaldatakase kas otse iga teenuse kõrvale või laiemalt iga klastri node'i ehk masina peale, kus on mitmed teenused.

Külgkorvi proksi ühendub teiste endasugustega läbi teenuse avastamise ning nii luuakse kinnine võrgustik, kuhu lastakse ainult selline võrgu liiklus, mis tuleb teistest proksidest või mis on eraldi lubatud. Nad elavad koos teenusega ning on nendele teenustele väravaks nende ees, kust käib läbi kõik liiklus, mis läheb teenusesse edasi. Need proksid on targad ning hoiavad pidevalt ennast kursis võrgus olevatest teistest proksidest ja nendega seotud teenustest. Samuti jälgib proksi ka teenust, mille küljes ta ise on, kasutades selleks perioodilisi päringuid ehk tervisekontrolle mingil kindlal aadressil, mis on teenuse poolt implementeeritud.

Kontroll kiht ühendab omavahel kõik osad ja proksid ning ühendab need üheks süsteemiks, kust on võimalik teha erinevaid tegevusi. Kontrollkiht koosneb mitmest

erinevast tööriistast, mille alla kuulub teenuste avastamine, logimine, koormuse jaotamine, monitoorimine, paigaldamine. Läbi kontrollpaneeli saavad proksid teada kuidas omavahel ühenduda. Kontrollkiht võimaldab teha sinise/rohelise paigaldusi, kuna ta saab juhtida võrguliiklust prokside pihta ning valikuliselt suunata mingit liiklust kindla teenuse pihta. Sinise/rohelise teenuse paigalduse muster on kahe erineva versiooni teenusest koos toodangu keskkonnas hoides ning liikluse jupihaaval uuele versioonile suunates. Samuti võimaldab see kiht juhtida süsteemi konfiguratsiooni ning muuta süsteemi päringute liikumise omadusi nagu uuestiproovimised ja äralõikamised ja koormuse suunamised.



Joonis 1 Külgkorvi meetodi vaade

2.2 Istio

Istio on Kubernetes [7] põhine teenuste võrgustiku lahendus, mis on algselt Lyfti poolt loodud ning mida on paljud suured tehnoloogiaettevõtted otsustanud kasutama ja toetama hakata. Google, IBM ja Microsoft kasutavad seda oma Kubernetese toodetes. Istio on eraldanud andmekihi ja kontrollkihi, kasutades külgkorvi proksit, mis laetakse kubernetesesse iga teenuse kõrvale. Kontrollkiht koosneb samuti podidest, mis jooksevad klastris, mis võimaldab paremat vastupidavust tõrgete puhul. [8]

2.3 Linkerd2

Linkerd on vabavaraline teenusevõrgustik, mis on disainitud kasutamiseks erinevates raamistikeks ja konteineri haldustarkvaras. Sellest sai esialgne teenusevõrgustik kui selle looja Buoyant esmakordselt kasutas väljendit “*Service mesh*”. Linkerd on algselt

kirjutatud Scala’s ja disainitud kasutamiseks hostipõhiselt. Kriitika selle võrdlemisi suure mälukasutuse pärast viis arendajad Conduiti välja töötamiseni, mis on kerge sevice mesh spetsiifiliselt disainitud Kubernetese jaoks ja kirjutatud Rust ja Go keeles. See projekt ühendati Linkerdiga, mis publitseeriti uuesti Linkerd2-na 2018 juuli. [9]

2.4 Consul

Consul on täis-funktsionaalsusega teenuse halduse raamistik ja koos Connect v1.2-ga annab see Consulile teenuse avastamise funktsionaalsuse, mis teeb temast komplektse teenusevõrgustiku. Consul on osa HashiCorp’i tööriistadest infrastruktuuri manageerimisel. See sai alguse Nomadi peal jooksvate teenuste manageerimise lahendusest ja on arendatud toetama teisi konteinereid jooksutavaid süsteeme kaasa arvatud Kubernetes. [10]

2.5 Maesh

Maesh on teenusevõrgustik, mis on ehitatud kasutades Traefik proksit. [11] Traefik on mõeldud lihtsaks implementeerimiseks ja kasutamiseks. Maesh võimaldab võrgustiku sees oleva liikluse jälgimist ja juhtimist, mis asub kubernetese sees. See liiklus on sama oluline arvesse võtta ja jälgida kui ingress ja egress liiklus. Maesh lubab lihtsat, kuid täisfunktsionaalset teenusevõrgustiku funktsionaalsust. See on ehitatud toetades Service Mesh Interface (SMI) spetsifikatsiooni, mis võimaldab integreerimist olemasolevate lahendustega. Maesh on ka opt-in, mis tähendab, et olemasolevad teenused saavad muutumatult edasi liikuda kuni otsustatakse see võrgustikku liita. [12]

2.6 ATAM meetod

Arhitektuuri kompromisside põhine analüüsi meetodit kasutatakse arhitektuuri hindamiseks ja võrdlemiseks kvalitatiivsete omaduste eesmärkide põhjal. ATAM meetod koosneb esialgselt üheksast sammust:

1. ATAM meetodi presenteerimine.
2. Äriliste eesmärkide presenteerimine.
3. Arhitektuuri presenteerimine.

4. Arhitektuuriliste võimaluste identifitseerimine.
5. Kvalitatiivsete atribuutide puu genereerimine.
6. Arhitektuuriliste võimaluste analüüsimine.
7. Ajurünnakul stsenaariumite leidmine ja prioritiseerimine.
8. Arhitektuuriliste võimaluste analüüsimine uuesti.
9. Tulemuste esitamine.

Originaalselt on see meetod mõeldud läbi viidama töötubade raames ning läbi otsese suhtluse. Tänapäevaste vahenditega on võimalik sama protsessi imiteerida ka läbi vastavate keskkondade, ja mitte otse füüsiliselt kokku tulles. See võimaldab spetsialistidel, kes sellest osa võtavad panna rohkem aega protsessi sisendit andes ja vähem aega panna logistilisele poolele, et leida aeg ja koht, kus kokku saada, et töötubasid läbi viia. Selles lõputöös modifitseeritakse seda meetodit veel, integreerides sellesse Saaty meetodi hierarhilise analüüsi protsessi, mis aitab lisaks võtta kokku spetsialistide leitud stsenaariumid ning anda neile kaalud, mille järgi on võimalik täpsemat valikut teha arhitektuuri valimisel. [3]

2.7 Saaty meetod

Saaty meetod on analüütilise hierarhia protsess ehk AHP(Analytical Hierarchy Process), kus kriteeriumide paaride võrdluse kaudu saadakse teada nende kriteeriumite kaalud ehk tähtsused. [5]

Seda kasutatakse ühe võimalusena subjektiivselt valitud kriteeriume hinnata ja leida objektiivne hierarhia kriteeriumitele. Selleks võetakse valitud kriteeriumid ning antakse need ekspertidele võrdlemiseks paarikaupa. Eksperdid peavad valima paarist olulisema ning hindama kindla skaala järgi nende erinevuse olulisust. Selle tulemusena saab järjestada kriteeriume nende omavaheliste suhete järgi.

Saaty meetodi protsessi osad on järgnevad:

- Vastavalt probleemile ära määratleda millist eesmärki meetodi kasutamisega täidetakse

- Kirjeldada otsustamise hierarhia, mille tipus on eesmärk, keskel otsustamiseks kriteeriumid ja lõpus valikuvariandid

- Koostada võrdlusmaatriks, kus on paarid, kus igat elementi võrreldakse tema all oleva elemendiga ja tema suhtes.

- Võetakse leitud hierarhia ning kaalutakse erinevate variantide omadusi selle järgi. Igale kriteeriumile liidetakse kõik temale omistatud kaalud ning arvutatakse selle järgi tema üldine prioriteet. Protsessi korratakse kuni kõik alumise taseme variandid on saanud omale kaalu.

Tabel 1 AHP meetodi hindamisskaala [4]

Tähtsuse osakaal	Definitsioon	Selgitus
1	Võrdse tähtsusega	Mõlemad omadused annavad tulemusse samapalju väärtust
2	Nõrgalt tähtsam	
3	Mõõdukalt tähtsam	Kogemused ja teadmised eelistavad nõrgalt üht omadust
4	Mõõdukamalt tähtsam	
5	Tugevalt tähtsam	Kogemused ja teadmised eelistavad tugevalt üht omadust
6	Tugevamalt tähtsam	
7	Väga tugevalt tähtsam	Kogemused ja teadmised eelistavad väga tugevalt üht omadust
8	Eriti tugevalt tähtsam	
9	Äärmiselt palju tähtsam	Ühte omadust väärtustatakse teisest absoluutselt tähtsamaks
Pöördväärtused	Kui ühele omadusele omistatakse ülevaltpoolt väärtus, siis teine võrreldav omadus saab väärtuseks valitud väärtuse pöördväärtuse	Loogiline järeldus

1.1 - 1.9	Omadused on väga võrdse tähtsusega	Tähtsust on keeruline määrada, kuid väike erinevus näitab siiski ühe omaduse suuremat tähtsust.

2.8 Eesmärgi püstitus

Ehitisregister liigub uuele mikroteenuste põhisele arhitektuurile ning selle jaoks on vaja leida sobivaimad vahendid, mis toetavad seda arhitektuuri valikut. Lõputöö raames luuakse pilootrakendus, mis hakkab asendama olemasoleva registri ühte osa ning see pilootrakendus paigaldatakse uuele arhitektuurile. Eesmärk on teha peale pilootrakenduse paigaldamist keskkonda järeldused ning valida sobiv arhitektuuriline süsteem. Uue ehitisregistri teenuste haldamiseks on plaan kasutusele võtta teenusevõrgustik. Sobiva lahenduse leidmiseks kasutame esmalt ATAM meetodit , millega leitakse täpsed kriteeriumid, mille põhjal hinnatakse erinevaid teenusevõrgustikke. Kriteeriumitele leitakse kaalud kasutades AHP meetodit ja hinnangute ning kaalude kokku arvutamisel lõpuks leitakse sobiv lahendus.

3 Olemasolev rakendus ja piloot

Selles peatükis kirjeldatakse olemasoleva Ehitisregistri süsteemi seisu ning autori poolt loodud pilootrakendust, mis on ehitatud uuele arhitektuurile. Tehakse järeldused pilootrakenduse arendamisest. Leitakse järeldused pilootrakenduse loomise järgselt ning põhjendatakse edasise arhitektuuri täiendamise vajadusi.

3.1 Olemasolev Java ja .net rakendus

EHR süsteem koosneb peamisest veebirakendusest ning lisamoodulitest. Peamine veebirakendus on Java rakendus, mis on paigaldatud Apache serveri peale ning lisamoodulid nagu VRM ja kaardirakenduse server on paigaldatud omaette serveri masinate peale, mida kõike haldab majutusteenust pakkuv ettevõte. Praegune EHRi süsteem on kolmas iteratsioon Eesti ehitisi ja ehitisega seotud toiminguid haldavast süsteemist. Praeguse rakenduse eellaseks on .net rakendus, mis on üle 10 aasta vanune tarkvara. See rakendus on praegu ümber nimetatud arhiiviks ning võimaldab vaadata selle rakendusega väljastatud dokumente ning kuni uue süsteemi peale üleminekut ehitiste andmekoosseisu. Vana rakenduse integratsioon uuega on minimaalne ning arhiivis andmete vaatamiseks juhitakse kasutaja brauseris uuele vahelehele, kus avatakse vana rakendus küsitud lehel. Samuti on vana rakendus staatilises seisus ning seda pole võimalik uuendada, sest puudub lähtekood. See on põhjustanud seisu, kus vana rakendus jookseb windowsi serveri peal, millel puudub tugi. Vana .net arhiivi rakendus on see osa, mida autori poolt loodud pilootrakendus hakkab asendama ning see lubab vananenud ja haldamatu tarkvara kasutuselt maha võtta. Eesmärk on integreerida uus rakendus praegusesse Ehitisregistrisse ning pakkuda sujuvamat kasutuskogemust vanade andmete vaatamiseks.

3.2 Pilootrakendus

Majandus- ja Kommunikatsiooniminsteerium(MKM), kes haldab EHRi, eesmärk on Ehitisregistrit täielikult kaasajastada ning viia monoliit rakendus üle uuele teenusepõhisele pilves elavale arhitektuurile. Praegune Java ja Wicketi põhine monoliitne rakendus viiakse üle järk-järgult, kasutades Brownfieldi tarkvaraarenduse meetodit [13], kus vana ja uus süsteem peavad samal ajal suutma koos töötada. Planeeritud on uued

loodud süsteemi osad asendada järjest vanas rakenduses ning hoida vana süsteem nii kaua kasutusel, kuni on võimalik vana UI raam ära vahetada uuel arhitektuuril põhineva kasutajaliidese raamiga. Esimese osana on plaanis vahetada välja vana arhiivi osa, kus säilib andmebaasi osa koos andmemudeli ja andmetega ning uuesti luuakse rakendus koos kasutajaliidesega kasutajale sobivamal ja lihtsamal kujul kasutamiseks.

Pilootrakendus on mõeldud olema üks komponent tulevasest süsteemist ning see koosneb tehniliselt kahest osast - arhiivi API teenus ja UI teenus. Arhiivi API on tehtud Java Spring Boot [14] rakendusena, mis kasutab andmeteks olemasoleva vana arhiivi Oracle baasi andmemudelit ja andmeid. Rakenduses töötleb andmed sobivale kujule ja need edastatakse UI-le. Kasutajaliidese rakendus on Reacti raamistiku põhine Javascripti rakendus. Riigil on eesmärk luua ühtsemat kogemust erinevates riigi keskkondades. Selleks on Maksu- ja Tolliamet loonud oma keskkonna jaoks React komponentide library ning jaganud seda repositooriumi MKMi koodivaramus, kust saavad kõik riigi asutused neid komponente endale kasutusele võtta. Pilootprojekt ehitati kasutades neid komponente. Nende komponentide kasutamise eesmärk on luua riigi süsteemidele sarnasem välimus ja kasutuskogemus, et süsteemid ei oleks nii erinevad.

Keeruliseks tegi pilootrakenduse arenduse olemasoleva .net rakenduse dokumentatsiooni ja lähtekoodi puudus. Seetõttu oli ainuke viis funktsionaalsuse loomiseks uues rakenduses vaadata läbi vana rakenduse kasutajaliideses kuvatavad andmed ning seejärel otsida andmebaasist vastavad seosed nendele väljadele.

3.3 Uuele ahritektuurile paigaldamine

Pilootrakendus on paigaldatud Riigipilve poolt hallatavale pilvekeskkonnale, millel jookseb Kubernetesel põhinev Rancheri haldussüsteem. Rancheri [15] abil paigaldatakse rakendus Riigipilve [16].

3.4 Pilootrakenduse paigaldamise järeldused

Pilootrakenduse eesmärk oli läbi katsetada uus platvorm ning läbi töötada esimesed protsessid ja paigaldus uuel arhitektuuril. Peale paigaldamist oli võimalik üle vaadata ja teha järeldused arhitektuuri suhtes ning vajalike muudatuse ja järgmiste sammude kohta.

Järgnevalt tuuakse välja ja selgitatakse lahti teemade kaupa järeldused, mis kogunesid autori poolt piloodi paigaldamisel ja selle järgselt ülejäänud süsteemi arenduse kohta.

3.4.1 Jälgimine

Tavapärane logimine ja teenuse tegevuste jälgimine toimub kõik teenuse sees ning logid on igale teenusele omased. Uuel arhitektuuril tekib juurde aga uue mõõtmena võrguliiklus, mis nüüd on rohkem osa süsteemist kui kunagi varem. Teenused ise ei ole võimelised endast väljapoole nii komplektselt jälgima, et see võimaldaks luua tervikpilti kogu süsteemis toimuvast tegevusest. Seetõttu tekib vajadus kõrgema komponendi järgi, mis näeb kogu liiklust ning suudab selle ühtselt kokku võtta ja arusaadavaks teha. Teenuse tegevust on ilma ühtse logimiseta võimalik jälgida ainult teenusest endast ning see ei näita kogu liiklust teenusega seoses, vaid ainult seda osa, mis toimub teenuse sees. Kui süsteem on suur ning päringuid toimub segamini mitme teenuse vahel, siis tekib vajadus ühtsema monitoorimise järgi, mis võimaldab lisaks teenuste logide agregeerimisele ühendada logidesse ka võrguliikluse ning siduda logimine omavahel, et saaks päringute kulgu jälgida läbivalt üle süsteemi. Samuti on võimalik implementeerida monitoorimine, et jälgida ka süsteemi seisukorda ja ressursikasutust üldiselt.

3.4.2 Turvalisus

Turvalisus on üks tähtsamaid aspekte, mida peavad kõik süsteemis olevad teenused jälgima ja mis on tihti ka üks keerukamaid, et see õigesti lahendada. Paljude teenustega süsteemil peavad kõik teenused implementeerima autentimise ja autoriseerimise iseseisvalt ning veenduma, et kõik töötab korrektselt. See nõuab iga teenuse arendamisel peale üldise turvalisuse aspektide implementeerimise ka ligipääsu kontrollide sisse ehitamist. Samuti tuleb iga teenuse puhul arendajal sellele aega panna ning iga kord uuesti implementeerimine toob endaga lahenduste killustatuse ja lahenduste kvaliteet ei ole kõikidel teenustel samal tasemel, vastavalt arendajate tasemele. Parem lahendus oleks vähemalt osaliselt turvalisust lahendada ühtselt ja väljaspool teenuseid kindla komponendiga. Seda on võimalik lahendada proksiga, mis paigaldatakse iga teenuse juurde ja on ühtne üle süsteemi. See proksi võtab vastu kogu teenusele suunatud liikluse ning suunab selle ise edasi teenusele vastavalt ligipääsudele ja kontrollidele. Proksit kasutades saaks olla kindel, et kõikidel teenustel on sarnane turvalisuse tase ning samasugune lahendus. Samuti võimaldab see ühtset seadistamise võimalust läbi ühe kindla liidese ning aja kokkuhoidu, hoides ära suure mahu arendust seoses turvalisusega

3.4.3 Teenuse koormuse jaotamine

Ehitisregistri uus süsteem paigaldatakse Riigipilve platvormile, millel jookseb Kubernetes. Kubernetes suudab näiteks ise koormust jagada vastavalt mingi teenuse instantside arvule. Mikroteenustega seoses juba enne mainitud juurde tekkinud võrgu liiklus on samuti üks osa süsteemist. See tähendab, et päringud ja nende haldamine ja kontrollimine muutub väga oluliseks. Seda olemasolev Kubernetese liides ei võimalda. Jällegi on võimalik teenusesse sisse kirjutada päringute uuesti proovimist ja teisi meetodeid, et muuta võrguliiklust kindlamaks. Teistpidi ei ole selles lahenduses ühtegi kontrollmehhanismi, mis jälgiks teenuste vastuvõtlikkust ning seisukorda vastavalt nendele suunatud päringute tulemusele. Kubernetes võimaldab küll tervisekontrolle implementeerida, kuid see jälgib rohkem üldist teenuse tervist ning ei anna otseselt aimu teenuse suutlikkusest sellel hetkel rohkem päringuid vastu võtta.

Selliste probleemide lahendamiseks peab jällegi kasutama välist komponenti, mis suudab näha suuremat pilti teenustest kui võrgustikust. Teenuse ette paigaldatud proksi suudab võtta kogu võrguga seotud tegevused enda kanda ning aidata muuta seeläbi süsteemi liiklust kindlamaks. Proksid on nähtavad läbi ühe liidese ning nad annavad edasi infot teenuse liikluse kohta, mis on väga hea näitaja teenuse seisukorra kohta. Selle info pealt on edasi võimalik teha otsuseid, kuidas süsteemi efektiivsemalt seadistada. Teine kasutegur prokside kasutamisest on konfiguratsiooni seadistamine, millega saab kiirelt süsteemiüleseid muudatusi sisse viia võrguliikluse käsitlemise konfiguratsioonis.

3.4.4 Ühtsed lahendused

Veel korra on mõistlik välja tuua ühe järeldusena edasivaatavalt süsteemi ühtsus. Mida rohkem suudetakse samasid funktsionaalsuseid erinevatel teenustel ühtlustada ja teenusest väljapoole tõsta, seda lihtsam on edaspidi nende haldus ja muutmine. Samuti kuna on teada, et arendajad, kes teenuseid arendavad on erinevad ning nende valitud keeled võivad erineda. See võib tekitada probleeme hetkedel, kui süsteemis toimuvad arhitektuurilised muutused, mis vajavad muutusi teenuste omavahelises suhtlemises.

3.5 Pilootrakenduse järelduse analüüs

Eelnevate järelduste põhjal on nähtav vajadus süsteemi juhtiva kihi järgi, mis haaraks enda alla kogu süsteemi ning suudaks olla kontrolliv ja juhtiv lahendus tervele süsteemile.

Sellist lahendust pakuvad teenusevõrgustikd, mis on mõeldud just mikroteenustel põhinevate süsteemide ühtseks haldamiseks. Teenusevõrgustik on loodud lahendama kõiki eelpool toodud probleeme ja võimaldab tänu prokside kasutamisele koos ühtsete liideste abil need lahendada. Teenusevõrgustike lahendusi on mitmeid ning õige võrgustiku valimiseks just sobivalt riigi süsteemi jaoks tuleb nende järelduste järgi hakata leidma täpsemalt kriteeriume, mille järgi hakata valima sobivat teenusevõrgustikku.

4 Kriteeriumide leidmine

Võttes aluseks eelmises peatükis tehtud järeldused ning lisaks üldised mikroarhitektuuri olulisemad omadused, leitakse neljandas peatükis kriteeriumid, mida kasutada teenusevõrgustike lahenduste hindamiseks. Kriteeriumid valitakse oluliste mikroteenuste arhitektuuri omaduste põhjal, võttes kasutusse osaliselt ATAM meetodit valiku täpsustamiseks. ATAM meetodiga on võimalik täpsustada omadusi juba stsenaariumite valikute tasemel, kasutades oma ala eksperte, et võrdlusesse satuksid sellele arhitektuurile vastavad olulised omadused.

4.1 Arhitektuuri valimise kriteeriumid

Kriteeriumite ja lõpuks ka lahenduse valimise kontekstiks on riigi poolt juhitud mikroteenuste arhitektuuril põhinev süsteem, mida on vaja hallata ning millel on alati mitu erinevat osapoolt, kellel on erinevad eesmärgid ja piirangud. Ühelt poolt on süsteemist tulu saav kliendi pool, kes soovib teha kõrgetasemelisi otsuseid ning juhtida süsteemi kulgu. Teiselt poolt on süsteemi huvigrupiks arendajad, kelle eesmärk on täita mingi ülesanne, kas uute arenduste või olemasoleva tarkvara hoolduse näol.

Esimese kriteeriumi valiku on võimalik teha olemasoleva tööde põhjal [17] ja juba uuritud kriteeriumite põhjal. Järgnevalt tuuakse ära esmane valik kriteeriumeid, mille põhjal hakata ATAM meetodiga täpsustama kriteeriume:

- Jõudlus
- Reliability
- Saadavus
- Turvalisus
- Muudetavus
- Porditavus
- Funktsionaalsus
- Variability
- Subsetability
- Kontseptuaalne terviklikkus

4.2 ATAM põhised stsenaariumid

Stsenaariumite leidmine läheb kokku eelnevalt selgitatud ATAM meetodi seitsmenda punktiga. Kaasates spetsialiste, leitakse nende jaoks tähtsaid stsenaariume, millega teenusevõrgustiku valimisel peaks arvestama. Selleks kasutati veebipõhist küsimustiku vormi, kus spetsialistid said lisada enda jaoks olulisi stsenaariume ning olemasolevaid stsenaariume võrrelda ja hinnata. Selle tulemusena tekkis nimekiri stsenaariumitest, millest siin tuuakse olulisemad, mis välja valiti:

Tabel 2 ATAM Stsenaariumid

Turvalisus	Võrgustikku lisatud teenustele pääseb väline liiklus ligi ainult kontrollitud viisil
	Teenus saab kasutada teist võrgustikus olevat teenust alles siis, kui talle on selleks õigused antud
	Väliste tokenite gateway autentimise mooduli võimalikkus
Jõudlus	Süsteemi osad püsivad kättesaadavad keeruliste toimingute ajal(teenusevõrgustik ei hakka takistama protsesse overheadi pärast)
	Võrgustikku teenuseid lisades kasvab ka võrgustiku haldamise koormus(iga teenusega lisandub teenusevõrgustiku lisad hakkavad koormama üldist süsteemi)
Saadavus	Süsteemi osad saavad korrektselt omavahel ühendust mahukate toimingute ajal
	Uue versiooni tõrgete puhul on võimalik suunata rakendus tagasi eelmisele versioonile
	Kontrollkihi tõrgete korral suudavad kõik süsteemi osad jätkata tööd
	Uue teenuse versiooni kasutusele võtmisel probleemide korral süsteem suudab taastuda
Muudetavus	Sobivama lahenduse leidmisel on võimalik võrgustiku osa asendada teisega
	Konfiguratsiooni muudatused kanduvad kiirelt üle kogu süsteemi
	Võrgustiku süsteemse osa võib vahetada teise tehnoloogiaga
	Süsteemi kesksete osade versioonide uuendamine ei tekita seisakuid
	Süsteem võimaldab kasutada erinevaid tööriistud ja tehnoloogiaid
	Süsteem koosneb osadest, mida on lihtne uuendada ja asendada

Teenuste haldus	Teenuste versioonide haldus toodangus
Arendamine	Teenust võib meshi lisada ja sealt välja võtta igal hetkel
	Arendaja kasutuskogemus, kasutuslihtsus
Administreerimine ja kasutuslihtsus	Süsteemi on võimalik viia üle teistele pilvemajutust pakkuvale keskkonnale (Azure [18], Kubernetes, AWS [19])
	Meshi ühte funktsionaalsust implementeerides ei tekita see hiljem suurt ajakulu teisele süsteemile ümber portimisel, sest põhifunktsionaalsused on sarnased või samad
Muudetavus ja osaline kasutus	Meshi on võimalik kasutusele võtta osade kaupa

4.3 Kriteeriumid

Esialgselt valitud kriteeriumite ja ATAM meetodi stsenaariumite põhjal tehti lõplik valik kriteeriumitest. Valituks saanud kriteeriumid selgitatakse lahti ning kirjeldatakse vastavas kontekstis, milleks on riigi poolt hallatava mikroteenuste võrgustik.

4.3.1 Turvalisus

Monoliitse rakenduse laiali lõhkumine pakub mitmeid erinevaid positiivseid külgi nagu agiilsus, skaleeruvus ja taaskasutatavus. Samas on mikroteenustel natuke erilisemad turvanõuded:

- Man-in-the-middle rünnaku [20]kaitseks peab kogu liiklus teenuste vahel olema krüpteeritud
- Paindlikku ligipääsu pakkumiseks peab olema võimalik kasutada teenuste vahelist vastastikust TLS-i (Transport Layer Security) [21] ja peeneteralist ligipääsude kontrolle
- Auditeerimiseks on vaja tööriistu, mis võimaldavad kirjeldada täpselt kes, kus ja mida tegi.

Teenusevõrgustikus peab olema tavaliselt kogu kommunikatsioon krüpteeritud. Ilma selle kasutamiseta on teenusevõrgustikus asuvad teenused avatud rünnakutele ja

manipuleerimisele. Kui autoriseerimist mitte kasutada, võivad teenused sattuda ohtu ning pahatahtlike kasutajate ohvriks.

Turvalisuse jaoks on oluline ka tööriistad, mida selle rakendamiseks kasutatakse. Lisaks nende olmeasolule võrgustikus on võimalus neid ise valida või vahetada võrgustikus üks motivaator võrgustiku valimisel.

Riigi poolt hallataval süsteemil on tihtipeale mitu arendaja osapoolt. Seetõttu on oluline, et ligipääsude määramine ja nende kontroll oleks selgesti piiritletud ja arusaadav kõikidele osapooltele.

4.3.2 Saadavus

Süsteemi valmisolek ja skaleeruvus on kokku võetud saadavuse ehk valmisoleku alla. Paljustki oleneb süsteemi üldine saadavus ehitatud teenustest ning nende võimekusest horisontaalselt skaleeruda. Siiski on oluline ka teenusevõrgustiku käitumine tõrgete ajal või teenusevõrgustikku hallatavate süsteemide probleemide korral.

Võrgustiku suutlikkus juhtida teenuseid selle all ning selle liiklust ja ressursse, võib suuresti mõjutada süsteemi kättesaadavust ja valmisolekut. Teenusevõrgustikud kasutavad liikluse kontrollimiseks ja juhtimiseks erinevaid võrgu suunamise tehnikaid nagu Circuit Breaker(kaitselüliti), mis lõikab ühenduse teenusega, kui see enam ei vasta, mis omakorda põhjustab liikluse teistele teenuse instantsidele suunamise.

4.3.3 Muudetavus ja osaline kasutus

Võrgustikud ise pakuvad komplekseid lahendusi, mille kõige kasutusele võtmine ühekorraga võib tekitada väga suuri muudatusi ja esmast panust. Lisaks võivad süsteemis olla juba mingid asjad kasutusel, mida võrgustik pakub teisel kujul. Seetõttu on oluline, et võrgustiku pakkumine oleks paindlik oma funktsionaalsuse kasutamises ning lubaks teha valikuid nende kasutusele võtmisel.

Ehitisregistri kontekstis on süsteem niivõrd uus, et võrgustiku täielikult kasutusele võtmine ei oleks väga ajamahukas ja konflikti tekitav, kuid kõikide süsteemide kasutamise võtmisel peavad olema valideerimise perioodid ning kohandumise aeg ja kui seda on võimalik teha osade kaupa, siis see lihtsustab protsessi läbiviimist, andes nii üleminekut sujuvamalt läbi viia.

4.3.4 Jõudlus

Iga teenusevõgustik on oma olemuselt üks lisakiht süsteemis, mis kindlasti tekitab lisakoormust ja nõuab lisaressurssi selle kasutamiseks. Oluliseks on kindlasti süsteemi latentsuse suurenemise määr, mida paratamatult kõik proksid lisavad. Võrgustikud kasutavad erinevaid lahendusi proksideks, mis omakorda tõlgendub lisandunud latentsuse mahus.

Teistpidi muutub jõudlus oluliseks suurema liikluse puhul ja suurtes süsteemides. Teenuste arvu kasvades kasvab ka kästiletavate komponentide arv, mida võrgustiku kontrollkiht peab suutma protsessida ja ajakohasena hoida. Kogu võrgustiku seadistuste ajakohase hoidmise ja konfiguratsiooni jagamise töömaht muutub võrgustiku suurusega.

4.3.5 Administreerimine ja kasutuslihtsus

Võrgustiku halduse lihtsus võib oluliselt mõjutada, kuidas süsteemi kasutusmugavus välja kujuneb. Oluline on võrgustiku kasutamise ja seadistamise lihtsus, mis peab olema üheselt mõistetav, kuid samas peab võrgustik lubama ka vajadusel ligipääsu funktsionaalsustele, mida tavapäraselt võibolla ei vajata.

Võrgustiku administreerimine on tihtipeale süsteemihaldurite tegevus, kuid süsteemi arendamisel peavad ka arendajad suutma paigaldada keskkondi võrgustikega.

Dokumentatsiooni olemasolu ning kommuuni olemasolu annavad aimu lahenduse elujõulisusest.

4.3.6 Teenuste ja liikluse käsitlemine

Teenusevõrgustiku üks eesmärk on pakkuda haldajale kontrollpaneeli, kust on võimalik näha võrgustiku seisu ja ka mingil määral juhtida võrgustikku. Teenusevõrgustiku võimelisus ära kasutama 7 taseme prokside funktsionaalsusi annab haldajale suuremat vabadust juhtida võrgustikku ja selle teenuseid.

4.3.7 Kontseptuaalne terviklikkus

Riigi poolt arendatavad süsteemid peavad tihtipeale mitmeid aastaid töötama ilma suurte süsteemi muudatusteta ning nende suurem eesmärk on olla pigem töökindlad kui eesrindlikud uue tehnoloogia näol. Samuti liiguvad riigi süsteemid üha enam vabavaraliste lahenduste poole. Seega mõjutab otsustamist kindlasti valitud

teenusevõrgustiku tugi ja tuleviku suund. Hinnatakse tugevat toetust võrgustikule ning aktiivsele kommuunile ja võrgustiku omanike tuleviku vaateid.

4.3.8 Monitoorimine

Selleks, et telemeetriat, jälgi ja mõõdikuid üldse näha, on vaja need esmalt kokku koguda. Muidu peaks iga teenust eraldi jälgima ning selle logid kokku korjama manuaalselt. See on enamasti tehtud kontrollkihi teenuse poolt, mis kogub kokku logid andmekihi proksidelt. Kui teenuseid on vähe, siis võib otsustada seda ka mitte teha. Osa telemeetriat on võib-olla vajalik võrgustiku normaalseks funktsioneerimiseks.

5 Küsitlus läbiviimine

Selles peatükis viiakse läbi küsitlus, vastavalt neljandas peatükis valitud kriteeriumitele. Küsitlus viiakse läbi kasutades AHP meetodit, mille tulemusena saavad valitud kriteeriumid endale tähtsuse osakaalud. Küsitluse vastajateks on spetsialistid, kes tegelevad mikroteenuste ja teenustevõrgustikega igapäevaselt.

Küsitlus viiakse läbi veebikeskkonnas [22], mis võimaldab läbi viia AHP põhiseid küsitlusi. Küsitlus koosneb 28 küsimusest, kus igat kriteeriumi võrreldakse omavahel. Igas küsimuses on vaja valida paarist olulisem kriteerium ning hinnata selle erinevuse suurust skaalal 1-9, kus 1 tähendab võrdset tähtsust ning 9 tähendab ekstreemselt ülekaalukat tähtsust võrreldes teise valikuga.

Küstluste tulemustest eeldatakse, et turvalisus kujuneb kõige suurema osakaaluga, sest see on sellise arhitektuuri ja riigi süsteemi puhul keeruline ja kõrge tähtsusega. Madalaima osakaalu võib saavutada kontseputaalne terviklikkus, sest mikroteenuste arhitektuur peaks olema võimeline uuendusteks oma ülesehituse poolelt.

5.1 Küsitluse tulemused

Küsitluse tulemused agregeeris veebikeskkond ise ning selle tulemusena väljastati tulemus. Joonis 2 kujutab endast iga vastanu eraldi lõpptulemust. Tulemused on ridade kaupa, kus ühel real on ühe vastanu lõpptulemuse osakaalud.

Turvalisus	Saadavus	Muudetavus ja osaline kasutus	Jõudlus	Administreerimine ja kasutuslihtsus	Teenuste ja liikluse käsitlemine	Kontseptuaalne vaade ja tulevik	Monitoorimine	CR_{max}
25.9%	25.4%	10.7%	7.9%	6.0%	6.6%	5.1%	12.5%	1.7%
24.2%	19.7%	11.4%	3.4%	6.2%	5.6%	5.9%	23.6%	9.2%
22.0%	24.8%	8.5%	6.7%	4.3%	14.9%	4.2%	14.6%	6.9%
30.8%	23.6%	14.6%	5.8%	5.0%	4.4%	11.5%	4.2%	9.3%
15.4%	34.4%	2.6%	32.9%	4.8%	2.3%	2.1%	5.4%	15.8%
25.6%	15.9%	22.6%	4.2%	6.1%	6.9%	2.4%	16.3%	7.7%

Joonis 2 Vastanute osakaalud

Joonisel 3 on kujutatud vastanute tulemused kokku agregeeritult lõplike osakaaludega.

Decision Hierarchy

Level 0	Level 1	Glb Prio.
Service meshi kriteeriumid	Turvalisus 0.259	25.9%
	Saadavus 0.254	25.4%
	Muudetavus ja osaline kasutus 0.107	10.7%
	Jõudlus 0.079	7.9%
	Administreerimine ja kasutuslihtsus 0.060	6.0%
	Teenuste ja liikluse käsitlemine 0.066	6.6%
	Kontseptuaalne vaade ja tulevik 0.051	5.1%
	Monitoorimine 0.125	12.5%
		1.0

Joonis 3 AHP küsitluse tulemus

Tulemustest on näha, et turvalisus ja saadavus on ülejäänutest kriteeriumitest kõvasti üle. Kõige rohkem tähtsustatakse saadavust ning samaväärselt tähtis on turvalisus. Edasi tulevad üle kahe korra madala osakaaludega monitoorimine, muudetavus ja osaline

kasutus ning jõudlus. Kõige madalama osakaaluga on jäänud teenuste ja liikluste käsitlemine, administreerimine ja kasutuslihtsus ning kontseptuaalne vaade ja tulevik.

6 Teenusevõrgustike lahenduste võrdlus

Selles peatükis tuuakse välja erinevate teenusevõrgustike variantide omadused vastavalt eelnevalt leitud tähtsamatele kriteeriumitele. Igat varianti hinnatakse kriteeriumi vastavusele ning lõpuks hinnangute järgi valitakse välja sobiv teenusevõrgustik.

Hindamiseks kasutatakse sarnaselt eelnevale küsitlusele punktiskaalat 1-9. Hindamine ise toimub teistmoodi kui küsitluses. Teenusevõrgustikke hakatakse hindama eraldi igat kriteeriumi. See erineb eelnevas küsitluses kasutatud AHP meetodist, sest jääb ära kõikide variantide kõikide omaduste omavaheline võrdlus. Kriteeriumite koguse tõttu oleks AHP tavapärasel meetodil hindamise läbiviimine väga ajamahukas ning segane. AHP meetodi kujul läbi viimine võib tekitada täpsema tulemuse asemel hoopis kallutatud ja liiga ühtlase tulemuse, olles mõjutatud liiga sügavalt eelnevatest valikutest ja teiste võrgustike omadustest. Seetõttu tehakse kriteeriumitele vastavuse hindamine järgnevalt koos kirjeldusega ning lõpus kasutatakse küsitluses saadud kaalusid, et viia kokku hinnangud ja leida lõplik tulemus.

6.1 Istio võrdlus

Selles alapeatükis kirjeldatakse Istio võrgustikku vastavalt kriteeriumitele ja hinnatakse neid.

6.1.1 Turvalisus

Istio pakub turvalisuse implementeerimiseks komplekset lahendust, mis võimaldab lahendada nii süsteemisisese truvalisuse, kui ka väljastpoolt sisse tuleva kommunikatsiooni turvamise. Samuti on Istio võimeline pakkuma ka auditeerimise teenuseid juba tehtud toimingute kontrollimiseks. Selleks on Istio kasutusele võtnud järgnevad komponendid:

- Citadel - võtme ja sertifikaadi manageerimine
- Külgkorvi ja ääre proksid, et implementeerida kommunikatsioon ja autoriseerimine klientide ja serveri vahel
- Pilot - jagab autentimise strateegiaid ja pakub turvalist nimelahendust proksidele

- Mixer - auditite manageerimine

Teenusevõrgustik on võimeline ka vaikimisi antud komponentide asemel kasutama juba olemasoleva süsteemi lahendusi, kui need suudavad liidestuda Istioga. Olemasolevad turvalahendused on võimalik integreerida Istio meshi ning puudub vajadus täielikult ümber teha süsteemi turvalisuses ülesehitus.

Hinnang: 8

6.1.2 Saadavus

Istio on kõrgelt skaleeruv tänu proksidele, mis lubab Istiol instantse luua vastavalt vajadusele ning Istio komponendid on võimelised haldama korraga tuhandeid pode. Vastavalt instantside ja podide arvule kasvab ka Piloti ressursside nõudlus, kuid ka Pilot on horisontaalselt skaleeruv ning ei tekita probleeme kõike omavahel ühendades.

Hinnang: 7

6.1.3 Muudetavus ja osaline kasutus

Istio võimaldab kasutada erinevaid komponente ning aktiivselt on lisatud juurde liidestumise võimalusi erinevate komponentidega

Istio komponente on võimalik asendada teiste samaväärsetega, kui olemasolev variant ei paku sobivaid funktsionaalsuseid või kui on kasutusel juba mingi komponent, mille protsessid on Istio võrgustiku jaoks sobivad.

Hinnang: 8

6.1.4 Jõudlus

Istio kasutatav Envoy proksi on Java põhine proksi, mis töötab küll kiiresti, kuid vajab märkimisväärselt rohkem mälu võrreldes teiste lahendustega. Kasutamist see ei mõjuta. Istio lisab igale päringule keskmiselt 6 ms latentsust. Skaleeruvust pidurdab vajadus päringuid suunata Mixer teenusele ning uuendused proksidele liiguvad läbi Pilot'i, mis on lisatakistus.

Hinnang: 5

6.1.5 Administreerimine ja kasutuslihtsus

Istio üles seadmine on dokumenteeritud väga detailselt ning pakub näiteid ja selgitusi läbivalt. Istio on mõeldud kasutamiseks väga paindlikult ning seetõttu on avatud väga palju funktsionaalsust, mida tavapäraselt vaja ei lähe. Funktsionaalsuste vahel otsimine ja seadistuste rohkus nõuab kindlasti teatud õppimiskõverat, kuid pakub selle eest väga paindlikku seadistamisvõimalust.

Istio funktsionaalsuse kasutamiseks on vaja seadistada mitu erinevat süsteemi, et süsteemi funktsionaalsust üldse terviklikuna kasutama hakata. Alates versioonist 1.4 on Istiol avalikustatud istioctl, mis lisab käsurealiidese võimalikkuse ning operaatoritele võimaluse jälgida võrgustikku läbi käsurea. Istio paigaldamiseks on võimalik kasutada Helmi ja samuti ka istioctl-i.

Hinnang: 4

6.1.6 Teenuste ja liikluse käsitlemine

Istio pakub võrguliikluse juhtimiseks mitmeid erinevaid funktsionaalsuseid: päringute suunamine versioonide vahel, sujuv liikluse suunamine, kaitselülitid, ajalimiitide seadistus.

Hinnang: 8

6.1.7 Kontseptuaalne terviklikkus

Istiol on tugev kommuun ning see on praegu enim tuntud võrgustik. Istio on vabavaraline, kuid selle suurteks toetajateks on suured ettevõtted nagu Google ja Red Hat Developers, kes on ka juhtival kohal Istio edasistes arengutes. Siiski on Istiol väga tugev arendajate kommuun, kes võtavad aktiivselt osa võrgustiku arendusest ja selle edasi viimisest. Istio on huvitatud võrgustiku edasi arendamisest ning kasutajate poolt nõutud vajalike funktsionaalsuste lisamisest.

Hinnang: 7

6.1.8 Monitoorimine

Istio genereerib detailse telemeetria kõikide teenuste jaoks võrgustikus. Telemeetria pakub jälgitavust teenusele, mis võimaldab operaatoritel tõrkeid avastada, hallata ja

optimiseerida nende süsteemi . Seda kõike ilma lisakohustusteta arendajatele. Läbi Istio näevad operaatorid kuidas teenused omavahel ja Istio kontrollkihiga suhtlevad.

Istio genereerib järgnevat telemeetriat, et pakkuda jälgitavust:

- Mõõdikud - Istio genereerib teenuse piires mõõdikuid, võttes aluseks latentsuse, liikluse, veateated ja küllastumise. Istio pakub ka detailseid mõõdikuid kontrollpaneelil, kus on genereeritud vaikimisi graafikud nende mõõdikute vaatamiseks.

- Hajutatud jäljed - Istio genereerib iga teenuse jaoks hajutatud jäljed, mis pakuvad operaatoritele detailse ülevaate päringute teekonnast ning teenuste omavahelistest sõltuvustest võrgustiku sees.

- Juurdepääsu logid -Istio on võimeline genereerima iga päringu kohta logi nii sissetuleva poole pealt kui ka väljaliikuvate vastuste pealt. See võimaldab operaatoritel auditeerida vajadusel iga teenuseinstantsi eraldi.

Hinnang: 7

6.2 Consul võrdlus

Selles alapeatükis kirjeldatakse Consul võrgustikku vastavalt kriteeriumitele ja hinnatakse neid.

6.2.1 Turvalisus

Consul kasutab turvalisuse ja TLS krüpteeritud suhtluse tagamiseks Client komponenti, mis erinevalt Istio andmekihi proksile asub igas klastris node'i tasemel. Võrgustikku lisatud teenused saavad oma turvalisuse läbi selle node küljes oleva Clienti. Seetõttu puudub Consulil 7 taseme TLS ja autoriseerimine. Seda on alati võimalik lisada ning Consul pakub selleks ühendamise võimalust erinevatele 7 taseme proksidele, et lisada vajalikku funktsionaalsust.

Hinnang: 4

6.2.2 Saadavus

Consul on jaotatud serveriteks ja nende all olevateks agentideks. Serveriteks nimetatakse ühe klastri node'i ja optimaalseks peetakse 3 kuni 5 instantsi. Serverite instantsid valivad omavahel master serveri, kelle läbi liiguvad kõik päringud. Masteri kadumise korral valitakse uus master. Kogu serveri sisene teenuste avastamine toimub läbi Gossip protocol'i, mis tähendab info jagamist läbi agentide, mitte otse ühe kindla teenuse avastaja vastu. See toimub samal viisil nagu kuulujuttude levimine - punktist punkti. Iga punkt saadab omakorda kõigile enda lähedal olevale punktile edasi info.

Hinnang: 7

6.2.3 Muudetavus ja osaline kasutus

Consul võimaldab kasutada ja asendada erinevaid komponente ning ühendada neid consuli võrgustikuga. Consulil on küll olemas proksi, kuid nad soovitavad kasutada vastavalt vajadusele sobivat ise valitud proksit ning ühendada see Consuli andmekihiga. See küll vähendab jõudlust, kuid lihtsustab üldist süsteemi.

Hinnang: 5

6.2.4 Jõudlus

Kuna Consulil pole proksisid iga teenuse kõrval siis on tal eelis jõudluse poolest, sest puudub samasugune koormuse kasv koos teenuste arvu suurenemisega nagu teistel lahendustel. Samuti ei lisandu päringutele ekstra latentsus tänu prokside puudumisele.

Hinnang: 8

6.2.5 Administreerimine ja kasutuslihtsus

Consul koosneb ühest komponendist, mis paigaldatakase iga nodei peale klastris ning sisaldab endas nii kontrollkihti kui ka andmekihti. Kõik teenused hoiavad ennast ajakohasena vastu Consul Clienti.

Hinnang: 4

6.2.6 Teenuste ja liikluse käsitlemine

Consulil on olemas 7nda kihi funktsionaalsused nagu urli põhine suunamine, liikluse suuna muutmine, koormuse jaotus ja monitoorimine. Puudu on 7ndal tasemel

identifitseerimine ja autoriseerimine, mis on olemas 4ndal kihil. Neid on võimalik lisada teenuse külgkorvi proksiga.

Hinnang: 5

6.2.7 Kontseptuaalne terviklikkus

Consul on eraettevõtte lahendus, kes pakub Consulit vabavaralise tootena ning samuti pakub ta tasulist tuge. Consulil on olemas kommuun, kus toimub aktiivne arutelu ja probleemide lahendamine. Consul on oma teenusevõrgustikku pakkunud juba aastast 2014 ning see on praktiline ja kasutatav tööriist, kuid kindlasti mitte terve lahendus vaid üks osa süsteemist.

Hinnang: 4

6.2.8 Monitoorimine

Consulis pole jälgimine koheselt saadaval. Kuna monitoorimine toimub teenuse tasemel, siis selleks on vaja Consulile lisada teenuste külge proksi ning seadistada see infot saatma valitud sihtkohta Consulis. See annab küll valikuvõimaluse valida omale sobivad tööriistad, kuid selle üles seadmine on lisatöö.

Hinnang: 2

6.3 Linkerd2 võrdlus

Selles alapeatükis kirjeldatakse Linkerd2 võrgustikku vastavalt kriteeriumitele ja hinnatakse neid.

6.3.1 Turvalisus

Linkerd kasutab kontrollkihil sarnaselt Istio’le külgkorvi proksit, mis automaatselt käivitab teenuste vahel mõlemapoolse TLS krüpteerimise. Ühendust prokside vahel aitab luua kontrollkiht. Erandina pole Linkerd2-l oma Ingressi ning toetub selleks täielikult kasutaja valikule. Selline valik on tehtud lihtsuse mõttes, et hoida võrgustikku väikse jalajäljega.

Hinnang: 7

6.3.2 Saadavus

Linkerd2 kasutab tavapärast teenuse avastamise lahendust ning lisaks sellega ka koormue jagamist, liikluse jagamist ja erinevad paigalduse variandid, et tagada saadavus. Erinevus on see, et see ei toetu Kubernetese teenuse avastamisele, vaid kasutab enda oma. Kui kontrollkiht kaob ära, siis kõik proksid jätkavad olemasoleva seadistuse järgi tööd. Uute teenuste lisandudes suudab süsteem ka DNSi järgi nendega ühenduda, aga uue proksi lisamisel ei saa see võrgustikku enne lisatud kui kontrollkiht uuesti kättesaadavaks muutub.

Hinnang: 4

6.3.3 Muudetavus ja osaline kasutus

Linkerd2 põhineb tugevalt nende enda loodud proksile, mis tähendab ka selle proksi kasutamist. Ühe komponendina peab Linkerd2-ga kindlasti kasutusele võtma eraldi Ingressi lahenduse, sest selle on nad lihtsuse mõttes oma komplektist välja jätnud. Samas on Linkerd väheste võimalustega kompoentide vahetamiseks ning mitmete asjade üle puudub otsene kontroll.

Hinnang: 5

6.3.4 Jõudlus

Linkerd2 on küll teine versioon Linkerd võrgustikust, kuid Linkerd2 on täiesti uus arhitektuur ning on ehitatud üles eelmisest versioonist õpitu pealt. Linkerd2 külgkorvi proksi on kirjutatud Go ja Rust keeles, mis annavad eelise ressursi kasutuse poolest ning on ka kiiruse poolest väga head, lisades päringutele minimaalselt latentsust.

Hinnang: 7

6.3.5 Administreerimine ja kasutuslihtsus

Linkerdi paigaldamiseks on vaja kasutada nende enda loodud CLI-d. Samas annab see hiljem palju käsurea võimalusi ja teeb need lihtsalt kättesaadavaks. See võimaldab teenust võrgutikku lisada, saada meetrikat teenuse kohta ning lisada klastrisse kontrollkihi komponente. Proksi seadistamise teenuse piires on võimalik annotatsioonidega üle kirjutada esialgsed seadistused.

Hinnang: 7

6.3.6 Teenuste ja liikluse kästlemine

Linkerd pakub väga mitmeid taseme 7 proksi funktionaalsusi nagu uuestiproovimised, automaatne mTLS(mutual Transport layer Security), liikluse jagamine ja suunamine jt.

Hinnang: 7

6.3.7 Kontseptuaalne terviklikkus

Linkerd2 on täiesti vabavaraline projekt ning seda arendatakse avalikult. Linkerd jälgib SMI protokolli suunda. Linkerd kuulub samas ühe ettevõtte alla ning selle suund on oluliselt mõjutatud ettevõtte huvidest.

Hinnang: 6

6.3.8 Monitoorimine

Linkerd2 pakub kohe "karbist välja" monitoorimist, mis töötab juba süsteemi üles pannes. Linkerd2 ei ole küll võimeline nägema teenuse sisest suhtlust, kuid annab teenuse välise tegevuse kohta infot nii läbi CLI, oma enda dashboardi kui ka Grafana eelseadistatud graafikutena. Monitooring aga on rohkem mõeldud lühiajaliseks jälgimiseks kui pikaajaliste trendide jälgimiseks ja logide kogumiseks. Selleks peab eraldi konfigureerima võrgustikku.

Hinnang: 8

6.4 Maesh võrdlus

Selles alapeatükis kirjeldatakse Maeshi võrgustikku vastavalt kriteeriumitele ja hinnatakse neid.

6.4.1 Turvalisus

Maeshis kasutatakse turvalisuse tagamiseks proksit iga node piires. Proksi kontrollib ligipääsu ainult node piires ning kõik teenused, mis on lisatud teenusevõrgustikku, saavad ligi teistele võrgustikus olevatele teenustele. See tähendab, et puudub eraldi autoriseerimise seadistamise võimalikkus ning sellega seoses puudub ka võimalus kaitsta igat teenust eraldi ning selle peab eraldi vajadusel lisama.

Hinnang: 3

6.4.2 Saadavus

Maesh pakub saadavuse konfigureerimist läbi Traefik proksi ning suudab juhtida liiklust tervetele teenustele. Nodei põhine proksi võimaldab väga suurt päringute kiirust ja läbilaskevõimet tänu väiksemale võrgustiku kihile.

Hinnang: 8

6.4.3 Muudetavus ja osaline kasutus

Maesh ise on väga õhuke ning lihtne, kuid on konkreetselt ehitatud Traefik proksi peale ning toetub sellele suuresti. Seetõttu ei ole võimalik Maeshi võrgustikus proksit asendada ning mitmeid funktsionaalsuseid, mis võrgustikus üldse puuduvad on vaja kasutajatel niikuinii ise juurde lisada

Hinnang: 3

6.4.4 Jõudlus

Kuna puudub iga teenuse küljes olevad proksid, püsib Maeshi protsessi vajadused väga madalad. Traefiku proksi on klastris node põhine proksi, mis haarab enda alla kõik selles node’s olevad teenused

Hinnang: 7

6.4.5 Administreerimine ja kasutuslihtsus

Maesh jälgib SMI protokolli ning on loodud olema võimalikult läbipaistev ja lihtne. Enamus Maeshi funktsionaalsust põhineb Traefiku proksil ja võrgustik on minimaalne, pakkudes lisaks protokolli funktsionaalsuseid. Maeshi üles panek on väga lihtne ning see koosneb ainult mõnest komponendist.

Hinnang: 8

6.4.6 Teenuste ja liikluse käsitlemine

Maesh võimaldab väga sujuvalt ja lihtsalt käivitada erinevaid funktsionaalsusi teenustel, kasutades selleks yaml konfiguratsiooni failide annotatsiooni välju. Seetõttu on väga lihtne esialgselt kasutusele võtta ainult esmased põhifunktsionaalsused nagu jälgimine.

Samuti on ka teenuste võrgustikku lisamisega, mis toimub lihtsalt läbi kasutatava urli muutmise. On olemas tase 7 funktsionaalsused nagu uuestiproovimised, kaitselülitid, voolu piirajad jne.

Hinnang: 7

6.4.7 Kontseptuaalne terviklikkus

Maesh on ühe ettevõte toode, mis on ehitatud nende enda proksi peale. See on 2019 avaldatud projekt ning seetõttu vähe kasutusel ja pole veel ennast tõestanud. Sellele on kindlasti palju toetajaid tänu paljude poolt kasutatavale Traefik proksile, mille peale Maesh on tervenisti ehitatud. Traefik ise on vabavaraline proksi.

Hinnang: 4

6.4.8 Monitoorimine

Võrgustikus kasutatav proksi Traefik pakub enamuse Maeshi funktsionaalsusest, kaasa arvatud monitoorimist. Traefik toetab viie erineva monitoorimise tarkvara: Jaeger, Zipkin, Datadog, Instana, Haystack. Traefik ise kasutab standardit OpenTracing päringute monitoorimisel.

Hinnang: 7

6.5 Võrgustike hindamise tulemused

Järgnevalt tuuakse välja hinnangute tulemused tabelitena. Selleks võeti hinnangud ning need pandi võrdlusmaatriksisse, kus selgitati välja vahekorrad erinevate valikute vahel kriteeriumi põhiselt. Seejärel võeti võrdlusmaatriksi tulemus ning järgmises tabelis võetakse selle kriteeriumi kohta võrdlusmaatriksi iga väli ja jagatakse läbi maatriksi summa reaga. See tulemus pannakse samatmoodi tabelisse ning rea lõppu arvutatakse kokku rea keskmine, mis annab iga valiku selle kriteeriumi prioriteedi. Erinevate valikute prioriteetide summa ühe kriteeriumi raames annab kokku terviku. Prioriteete saab hiljem edasi kasutada kriteeriumite osakaaludega läbi arvutamisel.

Tabel 3 Turvalisuse võrdlus

	Istio	**Consul**	**Linkerd2**	**Maesh**
Istio	1	2	1,143	2,667
Consul	0,5	1	0,571	1,333
Linkerd2	0,875	1,75	1	2,333
Maesh	0,375	0,75	0,429	1
Summa	2,75	5,5	3,143	7,333

Tabel 4 Truvalisuse võrdluse tulemus

	Istio	**Consul**	**Linkerd2**	**Maesh**	**Prioriteet**	**%**
Istio	0,364	0,364	0,364	0,364	0,364	36,364
Consul	0,182	0,182	0,182	0,182	0,182	18,182
Linkerd2	0,318	0,318	0,318	0,318	0,318	31,818
Maesh	0,136	0,136	0,136	0,136	0,136	13,636
				Summa	1	100

Tabel 5 Saadavuse võrdlus

	Istio	**Consul**	**Linkerd2**	**Maesh**
Istio	1	1	1,75	0,875
Consul	1	1	1,75	0,875
Linkerd2	0,571	0,571	1	0,5
Maesh	1,143	1,143	2	1
Summa	3,714	3,714	6,5	3,25

Tabel 6 Saadavuse võrdluse tulemus

	Istio	**Consul**	**Linkerd2**	**Maesh**	**Prioriteet**	**%**
Istio	0,269	0,269	0,269	0,269	0,269	26,923
Consul	0,269	0,269	0,269	0,269	0,269	26,923
Linkerd2	0,154	0,154	0,154	0,154	0,154	15,385
Maesh	0,308	0,308	0,308	0,308	0,308	30,769
				Summa	1	100

Tabel 7 Muudetavuse ja osalise kasutuse võrdlus

	Istio	**Consul**	**Linkerd2**	**Maesh**
Istio	1	1,4	1,4	2,333
Consul	0,714	1	1	1,667
Linkerd2	0,714	1	1	1,667
Maesh	0,429	0,6	0,6	1
Summa	2,857	4	4	6,667

Tabel 8 Muudetavuse ja osalise kasutuse võrdluse tulemus

	Istio	**Consul**	**Linkerd2**	**Maesh**	**Prioriteet**	**%**
Istio	0,35	0,35	0,35	0,35	0,35	35
Consul	0,25	0,25	0,25	0,25	0,25	25
Linkerd2	0,25	0,25	0,25	0,25	0,25	25
Maesh	0,15	0,15	0,15	0,15	0,15	15
				Summa	1	100

Tabel 9 Jõudluse võrdlus

	Istio	**Consul**	**Linkerd2**	**Maesh**
Istio	1	0,625	0,714	0,714
Consul	1,6	1	1,143	1,143
Linkerd2	1,4	0,875	1	1
Maesh	1,4	0,875	1	1
Summa	5,4	3,375	3,857	3,857

Tabel 10 Jõudluse võrdluse tulemus

	Istio	**Consul**	**Linkerd2**	**Maesh**	**Prioriteet**	**%**
Istio	0,185	0,185	0,185	0,185	0,185	18,519
Consul	0,296	0,296	0,296	0,296	0,296	29,630
Linkerd2	0,259	0,259	0,259	0,259	0,259	25,926
Maesh	0,259	0,259	0,259	0,259	0,259	25,926
				Summa	1	100

Tabel 11 Administreerimise ja kasutuslihtsuse võrdlus

	Istio	**Consul**	**Linkerd2**	**Maesh**
Istio	1	0,5	0,286	0,25
Consul	2	1	0,571	0,5
Linkerd2	3,5	1,75	1	0,875
Maesh	4	2	1,143	1
Summa	10,5	5,25	3	2,625

Tabel 12 Administreerimise ja kasutuslihtuse võrdluse tulemus

	Istio	**Consul**	**Linkerd2**	**Maesh**	**Prioriteet**	**%**
Istio	0,095	0,095	0,095	0,095	0,095	9,524
Consul	0,190	0,190	0,190	0,190	0,190	19,048
Linkerd2	0,333	0,333	0,333	0,333	0,333	33,333
Maesh	0,381	0,381	0,381	0,381	0,381	38,095
				Summa	1	100

Tabel 13 Teenuste ja liikluse käsitlemise võrdlus

	Istio	**Consul**	**Linkerd2**	**Maesh**
Istio	1	1,6	1,143	1,143
Consul	0,625	1	0,714	0,714
Linkerd2	0,875	1,4	1	
Maesh	0,875	1,4	1	1
Summa	3,375	5,4	3,857	2,857

Tabel 14 Teenuste ja liikluse käsitlemise võrdluse tulemus

	Istio	**Consul**	**Linkerd2**	**Maesh**	**Prioriteet**	**%**
Istio	0,296	0,296	0,296	0,4	0,322	32,222
Consul	0,185	0,185	0,185	0,25	0,201	20,139
Linkerd2	0,259	0,259	0,259	0	0,194	19,444
Maesh	0,259	0,259	0,259	0,35	0,282	28,194
				Summa	1	100

Tabel 15 Kontseptuaalse terviklikkuse võrdlus

	Istio	**Consul**	**Linkerd2**	**Maesh**
Istio	1	1,75	1,167	1,75
Consul	0,571	1	0,667	1
Linkerd2	0,857	1,5	1	1,5
Maesh	0,571	1	0,667	1
Summa	3	5,25	3,5	5,25

Tabel 16 Kontseptuaalse terviklikkuse võrdluse tulemus

	Istio	**Consul**	**Linkerd2**	**Maesh**	**Prioriteet**	**%**
Istio	0,333	0,333	0,333	0,333	0,333	33,333
Consul	0,190	0,190	0,190	0,190	0,190	19,048
Linkerd2	0,286	0,286	0,286	0,286	0,286	28,571
Maesh	0,190	0,190	0,190	0,190	0,190	19,048
				Summa	1	100

Tabel 17 Monitoorimise võrdlus

	Istio	**Consul**	**Linkerd2**	**Maesh**
Istio	1	3,5	0,875	1
Consul	0,286	1	0,25	0,286
Linkerd2	1,143	4	1	1,143
Maesh	1	3,5	0,875	1
Summa	3,429	12	3	3,429

Tabel 18 Monitoorimise võrdluse tulemus

	Istio	**Consul**	**Linkerd2**	**Maesh**	**Prioriteet**	**%**
Istio	0,292	0,292	0,292	0,292	0,292	29,167
Consul	0,083	0,083	0,083	0,083	0,083	8,333
Linkerd2	0,333	0,333	0,333	0,333	0,333	33,333
Maesh	0,292	0,292	0,292	0,292	0,292	29,167
				Summa	1	100

6.6 Osakaaludega arvestamine tulemustes

Osakaalude kasutamiseks pean võtma leitud hinnangute tulemused ning kõikide valikute tulemused läbi arvutama valiku kriteeriumi osakaaluga. Järgnevas tabelis tuuakse välja hinnangute tulemused ja osakaaludega läbi arvutamine.

Tabel 19 Osakaaludega läbi arvutamine

Kriteerium		Hinnang	Osakaal	Lõpptulemus
Turvalisus	Istio	0,364	0,259	0,094
	Consul	0,182	0,259	0,047
	Linkerd2	0,318	0,259	0,082
	Maesh	0,136	0,259	0,035
Saadavus	Istio	0,269	0,254	0,068
	Consul	0,269	0,254	0,068
	Linkerd2	0,154	0,254	0,039
	Maesh	0,308	0,254	0,078
Muudetvus ja osaline kasutus	Istio	0,35	0,107	0,037
	Consul	0,25	0,107	0,027
	Linkerd2	0,25	0,107	0,027
	Maesh	0,15	0,107	0,016
Jõudlus	Istio	0,185	0,079	0,015
	Consul	0,296	0,079	0,023
	Linkerd2	0,259	0,079	0,020

	Maesh	0,259	0,079	0,020
Administreerimine ja kasutuslihtsus	Istio	0,095	0,06	0,006
	Consul	0,190	0,06	0,011
	Linkerd2	0,333	0,06	0,020
	Maesh	0,381	0,06	0,023
Teenuste ja liikluse käsitlemine	Istio	0,322	0,066	0,021
	Consul	0,201	0,066	0,013
	Linkerd2	0,194	0,066	0,013
	Maesh	0,282	0,066	0,019
Kontseptuaalne terviklikkus	Istio	0,333	0,051	0,017
	Consul	0,190	0,051	0,010
	Linkerd2	0,286	0,051	0,015
	Maesh	0,190	0,051	0,010
Monitoorimine	Istio	0,292	0,125	0,036
	Consul	0,083	0,125	0,010
	Linkerd2	0,333	0,125	0,042
	Maesh	0,292	0,125	0,036

Neid tulemusi kokku võttes, on võimalik kokku võtta lõplik tulemus, mis tuuakse välja järgnevas tabelis.

Tabel 20 Valikute lõpptulemused

									Kokku
Istio	0,094	0,068	0,037	0,015	0,006	0,021	0,017	0,036	0,295
Consul	0,047	0,068	0,027	0,023	0,011	0,013	0,010	0,010	0,210
Linkerd2	0,082	0,039	0,027	0,020	0,020	0,013	0,015	0,042	0,258
Maesh	0,035	0,078	0,016	0,020	0,023	0,019	0,010	0,036	0,238

Tulemus näitab selgelt, et Istio on valikutest saanud kõige parema tulemuse. See läheb kokku esialgse ootusega, et Istio on valikutest kõige paindlikum ja sobitub kõige paremini riigi poolt juhitud süsteemi, võttes arvesse leitud kriteeriume. Järgmine valikutest on Linkerd2, mis näitab, et kohe algusest külgkorviga lahendus on piisavalt parem lahendus praeguse probleemi lahendamiseks ning sobib riigi poolt juhitava süsteemis kasutamiseks.

Tulemused on tugevalt mõjutatud hinnangutest, mida selle projekti raames valikutele anti. Kui valikuid annaks keegi teine või teise projekti raames, siis võivad tulemused tulla erinevad ning isegi nii palju, et Istio asemel on mõni teine valikuvariant parema tulemusega.

7 Kokkuvõte

Lõputöö eesmärk oli alustada Ehitisregistri üle viimist uuele arhitektuurilisele platvormile, luues esmalt pilootrakenduse, mis paigaldatakse uuele arhitektuurile ja integreeritakse olemasolevasse süsteemi. Seejärel oli eesmärk teha järeldused uue arhitektuuri kohta ning kasutada järeldusi aluseks, et leida kriteeriumid uue arhitektuuri võrgustiku valimise jaoks. Lõpuks hinnata erinevaid valikuid teenusevõrgustikest vastavalt leitud kriteeriumitele ning leida sobiv valik, mida edaspidi hakata implementeerima uuel arhitektuuril.

Esmalt arendas autor pilootprojekti ning paigaldas selle Riigipilvel asuvale Kubernetes keskkonnale ning integreeris uue rakenduse olemasolevasse süsteemi.

Pilootrakenduse järeldusest selgus vajadus uuel arhitektuurilisel platvormil võtta kasutusele ühtne haldav süsteem, mida pakuvad teenusevõrgustikud, mis koondavad enda alla kõik süsteemi teenused ning lubab osa võrguga seonduvat funktsionaalsust tõsta väljapoole teenustest, kasutades selleks korduvkasutatavaid komponente.

Kriteeriumite valimisel kasutati olemasolevat informatsiooni ning kasutati seda ATAM meetodiga, et leida stsenaariume, mille järgi täpsustati kriteeriumite nimekirja. Lõpuks valiti välja 8 kriteeriumit, mis kõige paremini võtsid kokku teenusevõrgustiku tähtsamad omadused.

Leitud kriteeriume kasutati välja valitud teenusevõrgustike lahenduste variantide hindamiseks. Kõik kriteeriumid hinnati autori poolt iga valiku kohta. Sellist varianti hindamiseks kasutati konkreetsuse ja lihtsuse eesmärgil ning lisaks ka aja kokkuhoidmise eesmärgil. Kriteeriumite enda hindamisel kasutatavat AHP meetodit ei olnud mõistlik kasutata valikute hindamiseks, sest selle küsitluse läbiviimine oleks olnud vastajate jaoks liiga mahukas ja segadusttekitav. Kriteeriumite järgi hinnangud pandi võrdlusmaatriksisse ning iga kriteeriumi järgi leiti lahenduste prioriteedid.

AHP meetodiga leiti kriteeriumitele osakaalud, mida kasutati lahenduste pealt välja arvutatud prioriteetide peal ning arvutati välja lõpptulemused. Parima tulemuse sai ka esmaselt välja pakutud Istio võrgustik, mis oli teistest üle selles kontekstis tänu oma paindlikkusele ja võimaluste rohkusele.

Hindamisel saadud lõpptulemus ei olnud kindlasti eelnevalt kindel teadmine ning protsessi ja hindamise tulemusena on näha, et ka teised variandid on võrreldavad ning praegu parema tulemuse saanud Istio võrgustik ei pruugi teistel tingimustel ja teises projektis sugugi olla parim valik.

Autori poolt loodud pilootrakendus on viidud toodangusse ning on kasutuses leheküljel www.ehr.ee. Tavakasutajale on seal näha see ehitise andmetes alammenüü „Ehitise andmed seisuga 30.09.2014“ vaates, kus kuvatakse andmeid läbi uue rakenduse. Ülejäänud uue rakenduse funktsionaalsusest on menetlejate ja kohalike omavalitsuste töötajatele kasutamiseks. Kui arneduste edasi minnakse, siis võib selle vaate asukoht muutuda.

Välja valitud kriteeriume on võimalik kasutada edaspidi ka teiste sarnaste süsteemide hindamisel ning kasutada otsustamiseks sama kooslust AHP meetodist ja võrreldavate valikute hindamisest.

8 Kasutatud kirjandus

[1] I. Nadareishvili, Microservice architecture aligning principles practices, and culture, 2016.

[2] S. Floyd, „What Is a Service Mesh?," Nginx, 3 04 2018. [Võrgumaterjal]. Available: https://www.nginx.com/blog/what-is-a-service-mesh/. [Kasutatud 13 10 2019].

[3] P. Wallin, „Making Decisions in Integration of Automotive Software and Electronics: A Method Based on ATAM and AHP," 11 07 2007. [Võrgumaterjal]. Available: https://ieeexplore.ieee.org/abstract/document/4228592. [Kasutatud 23 10 2019].

[4] S. Somma, „Sisulahenduse valimise metoodika," 2017. [Võrgumaterjal]. Available: https://digikogu.taltech.ee/et/Item/e812009d-fa25-42d4-b03a-6c1d1fa71d24. [Kasutatud 20 12 2019].

[5] T. Saaty, „Decision making with the analytic hierarchy process – International journal of services sciences, 1(1), 83-98.," [Võrgumaterjal]. Available: http://www.rafikulislam.com/uploads/resourses/197245512559a37aadea6d.pdf. [Kasutatud 20 10 2019].

[6] A. E. M. a. U. Z. Guiding, „Architectural Decision Making on Service Mesh Based Microservice Architectures," [Võrgumaterjal]. Available: http://eprints.cs.univie.ac.at/6073/1/paper.pdf. [Kasutatud 10 12 2019].

[7] Kubernetes, „Kubernetes docs," [Võrgumaterjal]. Available: https://kubernetes.io/docs/home/. [Kasutatud 17 10 2019].

[8] Istio, „Docs," [Võrgumaterjal]. Available: https://istio.io/docs/. [Kasutatud 12 12 2019].

[9] Linkerd2, „Linkerd2 overview," [Võrgumaterjal]. Available: https://linkerd.io/2/overview/. [Kasutatud 07 11 2019].

[10] Consul, „Consul docs," [Võrgumaterjal]. Available: https://www.consul.io/docs/index.html. [Kasutatud 07 11 2019].

[11] Maesh, „Maesh," [Võrgumaterjal]. Available: https://containo.us/maesh/. [Kasutatud 12 12 2019].

[12] Maesh, „Containous Maesh," [Võrgumaterjal]. Available: https://github.com/containous/maesh. [Kasutatud 12 12 2019].

[13] K. J. Richard Hopkins, Eating the IT Elephant: Moving from Greenfield Development to Brownfield, 2008.

[14] Spring, „Spring Boot," [Võrgumaterjal]. Available: https://spring.io/projects/spring-boot. [Kasutatud 25 10 2019].

[15] Rancher, „Rancher docs," [Võrgumaterjal]. Available: https://rancher.com/docs/. [Kasutatud 26 10 2019].

[16] Riigipilv, „Riigipilv," [Võrgumaterjal]. Available: https://riigipilv.ee/. [Kasutatud 27 09 2019].

[17] B. Boehm, „Evaluating a Software Architecture,“ 02 03 2017. [Võrgumaterjal]. Available: http://catalogue.pearsoned.co.uk/samplechapter/020170482X.pdf. [Kasutatud 12 10 2019].

[18] Microsoft Azure, „Azure,“ [Võrgumaterjal]. Available: https://azure.microsoft.com/en-us/. [Kasutatud 17 09 2019].

[19] Amazon AWS, „Amazon EC2,“ [Võrgumaterjal]. Available: https://aws.amazon.com/ec2/. [Kasutatud 17 10 2019].

[20] K. Adams, „Detecting and preventing man-in-the-middle attacks on an encrypted connection“. United States Patent US10171250B2, 30 09 2013.

[21] A. M. D. S. Colin Boyd, „Transport Layer Security Protocol,“ %1 *Protocols for Authentication and Key Establishment*, 2019, pp. 241-242.

[22] K. D. Goepel, „AHP Online System -AHP - OS,“ [Võrgumaterjal]. Available: https://bpmsg.com/ahp/. [Kasutatud 04 01 2020].