

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Harry Vipper 194091IACB

**ENTRINGO PARKIMIS- JA
LÄBIPÄÄSULAHENDUSE TARKVARA
HALDUSE AUTOMATISEERIMINE**

Bakalaureusetöö

Juhendaja: Edmund Laugasson
Magistrikraad

Kaasjuhendaja: Joan Markus Varik
Bakalaureusekraad

Tallinn 2022

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: Harry Vipper

15.05.2022

Annotatsioon

Käesoleva diplomitöö eesmärgiks oli vähendada Hansab Group'i poolt loodud Entringo parkimis- ja läbipääsulahenduse tarkvara paigaldamiseks, uuendamiseks ja seadistamiseks kuluvat aega, leides nende tegevuste teostamiseks automatiseeritud lahendus.

Bakalaureusetöös kirjeldas autor lugejale Entringo parkimis- ja läbipääsulahendust ning tarkvara keskhaldussüsteeme. Vastava ülesande teostamist ja sobivust kirjeldati pikemalt analüüsi osas. Tarkvara keskhaldussüsteemide sobivuse analüüsi tulemusena otsustas autor tarkvara keskhaldussüsteemi kaasamise vastu, luues ise Ansible arhitektuurist inspireeritud haldustööriista. Autor kirjeldas selle lahenduste loomiseks kasutatud tehnoloogiaid ja lähenemisi ning selle funktsionaalsust.

Diplomitöö tulemusena valmis Entringo haldustööriist, mis kiirendas protsesse Entringo tarkvara peamiste komponentide (SBO, LAS ja HAL) uuendamiseks, seadistamiseks ja paigaldamiseks, millest järeldatuna sai diplomitöös püstitatud ülesanne edukalt lahendatud.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 36 leheküljel, 7 peatükki, 7 joonist, 1 tabel.

Abstract

Automating Entringo Parking and Access Control System Software Management

Antecedently to this thesis the Entringo parking and access control system software management was lacking automation, and thus software management was arduously undertaken by hand. These manual processes suffered heavily from human error and were time consuming. In conjunction with Hansab Group the author set out to combat this legacy approach to software management.

The aim of this thesis was to create an automated solution for the management of primary Entringo software components (SBO, LAS and HAL) on Ubuntu long term support operating systems, congruent to the Entringo software architecture.

To achieve the intended goal, the author first analysed software configuration management tools, as a result of which, none were identified as suitable. The author instead embraced a grassroots approach to develop a new solution most appropriate to Entringo, which he dubbed the 'Entringo Management Tool'. This developed solution derives architectural inspiration from configuration management tools akin to Ansible, implementing a hierarchical but agentless solution. Subsequently the methods and technologies pertaining to this solution were discussed. The thesis also briefly explained the prerequisites to these topics and the Entringo system as a whole.

In testing, the created tool proved it was capable of performing the required tasks for software management of Entringo parking and access control systems and thus was adopted by Hansab Group for software management on client's systems.

The thesis is in Estonian and contains 36 pages of text, 7 chapters, 7 figures, 1 table.

Lühendite ja mõistete sõnastik

Entringo	Parkimis- ja läbipääsulahendus arendatud Eestis ettevõtte Hansab Group poolt [1].
SBO	Entringo tarkvara komponent, lühend rakenduse Inglise keelsest nimetusest – <i>System Back Office</i> (süsteemi tsentraalne haldusserver). Kasutatud ka seadme puhul, kuhu antud rakendus on paigaldatud tunnustamiseks [1].
LAS	Entringo tarkvara komponent, lühend rakenduse Inglise keelsest nimetusest – <i>Local Area Server</i> (lokaalne ala server). Kasutatud ka seadme puhul, kuhu antud rakendus on paigaldatud tunnustamiseks [1].
HAL	Entringo tarkvara komponent, lühend rakenduse Inglise keelsest nimetusest – <i>Hardware Layer</i> (riistvarakiht). Kasutatud ka seadme puhul, kuhu antud rakendus on paigaldatud tunnustamiseks [1].
Rajaterminal	Entringo süsteemi füüsiline komponent, mis on paigaldatud tõkkepuu ette kuvamaks ekraanile teateid [1].
RPC	<i>Remote Procedure Call</i> on viis teises seadmes protseduuri käivitamiseks.
YAML	<i>YAML Ain't Markup Language</i> on inimloetav andmete serialiseerimise keel, kasutatud enamasti rakenduste seadistusfailides [2].
HTTPS	<i>Hypertext Transfer Protocol Secure</i> ehk turvaline hüperteksti edastusprotokoll on kasutusel krüpteeritud andmete edastamiseks arvutivõrkudes.
Ubuntu	Kõige populaarsem Linuxi distributsioon, arendatud Canonical Ltd. poolt, mis rakendab suures osas vabavara.
PostgreSQL	Relatsiooniliste andmebaaside haldamise tarkvara, kasutatud osade Entringo tarkvaraliste komponentide (SBO ja LAS) poolt [1].
Elasticsearch-OSS	<i>Elasticsearch - Open Source Software</i> on sõne otsingumootor, kasutatud Entringo tarkvaraliste komponendi SBO poolt [1].
IP-aadress	<i>Internet Protocol address</i> ehk internetiaadress on võrku ühendatud seadmele määratud identifitseerimisnumber.
IPv4	<i>Internet Protocol version 4</i> ehk internetiprotokolli neljas versioon on standard, mis määrab IP-aadresside esitamise formaadi.

SSH	<i>Secure Shell</i> ehk turvakest on krüptograafiline võrguprotokoll turvaliseks andmete edastuseks.
SCP	<i>Secure Copy</i> on programm, mis on mõeldud failide kopeerimiseks erinevate seadmete vahel üle SSH protokoll.
SFTP	<i>SSH File Transfer Protocol</i> ehk turvaline failiedastusprotokoll on programm, mis on mõeldud failide kopeerimiseks erinevate seadmete vahel üle SSH protokoll.
<i>Systemd daemon</i>	Uute Ubuntu operatsioonisüsteemide osa, mis võimaldab hallata seadmes töötavaid teenuseid.
DevOps	<i>Development and Operations</i> on tarkvara arendamise lähenemine või selle kultuur, mille käigus seotakse arendus- ja haldusmetoodikad [3].
Node.js	JavaScript'i käitussüsteem [4].
npm	Node.js'iga koos levitatud JavaScripti paketi haldur [5].

Sisukord

1 Sissejuhatus	11
2 Taust	12
2.1 Entringo	12
2.1.1 <i>System Back Office</i> – Süsteemi tsentraalne haldusserver	12
2.1.2 <i>Local Area Server</i> – Lokaalne ala server	13
2.1.3 <i>Hardware Layer</i> – Riistvarakiht	13
2.1.4 Maksejaam ja teised komponendid	13
2.1.5 Entringo lahenduse arhitektuuri joonis	14
2.2 Hetkeolukorra analüüs	15
2.2.1 Entringo tarkvara paigaldus	15
2.2.2 Entringo tarkvara uuendamine	16
2.2.3 Entringo tarkvara seadistamine	16
2.3 Nõuded lahendusele	17
2.4 Ülesande püstitus	18
2.5 Tarkvara keskhaldussüsteemid	18
2.5.1 Lükkepõhised	18
2.5.2 Tõmbepõhised	19
3 Arhitektuuri analüüs	20
3.1 Tarkvara keskhaldussüsteemide analüüs	20
3.1.1 Lükkepõhised	20
3.1.2 Tõmbepõhised	21
3.1.3 Peamine probleem tarkvara keskhaldussüsteemidega	21
3.2 Lahenduse arhitektuuri valik	22
4 Tehnoloogiad	26
4.1 Tarkvaratehnika	26
4.2 Testkeskkond	27
4.3 Käskluste edastamise meetod	28
4.4 Programmeerimiskeel	30
4.5 Kasutatud vabavaralised teegid	31

4.6 Lahenduse teisaldatavus	32
5 Lahenduse funktsionaalsus	34
5.1 Kasutajaliides.....	34
5.2 Entringo tarkvara uuendamine.....	35
5.3 Entringo tarkvara seadistamine	37
5.4 Entringo tarkvara paigaldamine.....	39
6 Lahenduse analüüs.....	42
6.1 Testimise tulemused	42
6.2 Turvalisus	44
6.3 Tuleviku võimalused	45
Kokkuvõte	46
Kasutatud kirjandus	47
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	49
Lisa 2 – Näide kasutatud Here document'ist üle SSH käskluste edastamiseks.....	50
Lisa 3 – Haldustööriista lähtekood	51

Jooniste loetelu

Joonis 1. Entringo süsteemi võimalik ülesehitus [1].	14
Joonis 2. Entringo haldustööriista arhitektuur.	25
Joonis 3. Lahenduse testimiseks loodud virtuaalmasinad VMware vSphere 7 keskkonnas.	27
Joonis 4. Haldustööriista peamenüü.	35
Joonis 5. Näidis valikud haldustööriista kaudu uue süsteemi lisamisel.	35
Joonis 6. Entringo tarkvara uuendamise automatiseeritud protsess.	36
Joonis 7. Protsess Entringo tarkvara seadistusfailide laadimiseks lokaalsesse seadmesse.	38

Tabelite loetelu

Tabel 1. Tõmbe ja lükke meetodite võrdlus lahenduse arhitektuuri valikuks.	22
---	----

1 Sissejuhatus

Tänapäeva ühiskond on muutumas aina mugavamaks, auto omanike arv üha kasvab ja vajadus uute mugavate lahenduste järele, mis hõlbustavad auto omamisega kaasnevad protsesse sh parkimine on igati teretulnud. Entringo parkimis- ja läbipääsulahendus on loodud just neid trende silmas pidades, eesmärgiga muuta auto parkimine juhtidele ja parklaoperaatoritele võimalikult mugavaks. Selle saavutamiseks on loodud piletivaba parkimis- ja läbipääsulahendus, mis kasutab numbrituvastuskaameraid, et vähendada parklasse sisenemiseks ja sealt väljumiseks kuluvat aega. Kuna tegemist on uudse lahendusega, mis on alles agiilses arenduses, et täiendada olemasolevaid süsteemi võimekusi ning ka neid juurde lisada, antakse tihti välja uusi süsteemiversioone. Olles keskendunud peaasjalikult kliendi mugavustele, on Entringo süsteemi automatiseeritud paigaldamise, uuendamise ja seadistamise võimekus jäänud tahaplaanile ning aina kasvavate süsteemide arvu tõttu on see muutumas probleemseks.

Käesoleva diplomitöö eesmärgiks on luua automatiseeritud lahendus Hansab Group'i poolt loodud Entringo parkimis- ja läbipääsulahenduse tarkvara paigaldamiseks, uuendamiseks ja seadistamiseks Ubuntu pikaajalise toega operatsioonisüsteemidel, mis oleks Entringo tarkvaralise arhitektuuriga kooskõlas ja küberturvaline.

Bakalaureusetöö koosneb kolmest osast: ülesande taust, analüüs ja lahenduse tehniline teostus. Esimeses osas kirjeldab autor lugejale Entringo parkimis- ja läbipääsulahendust ning tarkvara keskhaldussüsteeme. Teises osas analüüsitakse antud tarkvara keskhaldussüsteemide sobivust ülesande teostamiseks, mille tulemusena otsustab autor tarkvara keskhaldussüsteemi kaasamise vastu ning valib loodavale lahendusele sobiva arhitektuuri. Kolmandas osas kirjeldab autor lahenduse loomiseks kasutatud tehnoloogiaid, lähenemisi ja funktsionaalsust ning analüüsib loodud lahenduse vastavust sellele seatud nõuetele.

2 Taust

Entringo parkimis- ja läbipääsulahendus on loodud eesmärgiga muuta auto parkimine juhtidele ja parklaoperaatoritele võimalikult mugavaks. Selleks on arendatud piletivaba parkimis- ja läbipääsulahendus, mis kasutab numbrituvastuskaameraid, et vähendada juhtidel parklasse sisenemiseks ja sealt väljumiseks kuluvat aega. Traditsiooniline parkimispilet on asendatud auto numbrimärgiga, mida tuvastatakse parklasse sisenemisel ja sealt väljumisel numbrituvastuskaameratega [1].

2.1 Entringo

Järgnevas alampeatükis tutvustab autor lugejale, Entringo parkimis- ja läbipääsulahenduse tarkvara kolmetasemelist arhitektuuri hierarhiat. Entringo süsteemid koosnevad vähemalt kolmest erinevast tarkvara kihist, mis võivad osaliselt või täielikult olla paigaldatud samale seadmele. Need kolm tarkvaralist komponenti on tsentraalne haldusserver, lokaalne ala server ja riistvarakiht, ehk vastavalt: SBO, LAS ja HAL. Entringo tarkvaraliste komponentide poolt täidetav ülesanne jaguneb järgnevalt [1].

2.1.1 *System Back Office* – Süsteemi tsentraalne haldusserver

Entringo süsteemi tööd koordineerib tsentraalne haldusserver, mille peamine ülesanne on kõigi operaatori parklate töö juhtimine. Server on operaatori põhine ning annab süsteemioperaatorile läbi veebiliidese võimaluse oma parklaid hallata. Veebiliidest saab operaator näiteks jälgida, oma parklates parkivate autode arvu, nende parkimisaega ja ka palju muud. SBO on liidestatud kõigi parklaoperaatori parklatega, kuhu on paigaldatud lokaalne ala server. Koos SBO rakendusega on paigaldatud ka andmebaasid, kuhu kõikide ala serverite andmed varundatakse. Parklate juhtimine toimub läbi RPC käskluste, mis on sertifikaadi alusel valideeritud [1].

2.1.2 Local Area Server – Lokaalne ala server

Ühe parkla tööd juhib lokaalne ala server, mis on paigaldatud parkla sisevõrku, tõstmaks süsteemi kiirust. See server juhib kõiki parkla sisse- ja väljasõite, määrates ära näiteks parklasse lubatud autode arvu, nende numbrimärgid, parkimishinna ning ühtlasi ka juhtidele kuvatud teated. Sarnaselt tsentraalsele haldusserverile on ka lokaalse ala serveriga koos paigaldatud andmebaas, mis talletab kõiki selle parklaga seotud andmeid. Parklasse kuuluvate sisse- ja väljasõitude juhtimine toimub läbi RPC käskluste, mis on sertifikaadi alusel valideeritud [1].

2.1.3 Hardware Layer – Riistvarakiht

Süsteemi riistvarakiht juhib ühte sisse- või väljasõitu. Selleks koordineerib see auto numbrimärgi tuvastus kaamera ja tõkkepuu tööd. Enamasti on HAL'i rakendus paigaldatud äriklassi arvutile rajaterminalis, mille külge ühendatud ekraani kasutatakse juhile teadete edastamiseks, nende hulka kuuluvad suunised peatumiseks või sõidujätkamiseks, numbrimärgituvastused, aga ka operaatore poolt määratud erisõnumid, näiteks parklakorra, vms kohta. Leidub paigaldatud süsteeme, kus ekraan puudub, mis juhul vastutab HAL ainult tõkkepuu ja kaamera juhtimise eest. Tuvastades auto jõudmise tõkkepuu ette, alustatakse kaameras numbrimärgituvastuse protsess, mille tulemus edastatakse päringuga LAS'i kontrollimaks, kas autol on lubatud parklasse siseneda või sealt väljuda. LAS'ist saadud positiivse vastuse korral kuvatakse juhile teade ja avatakse tõkkepuu, negatiivse vastuse puhul kuvatakse juhile teade ja põhjus, miks tõkkepuud ei avatud. Antud diplomitöös käsitletakse riistvarakihi all ainult Entringo HAL rakendust, mitte selle poolt hallatavaid kaameraid ega tõkkepuid [1].

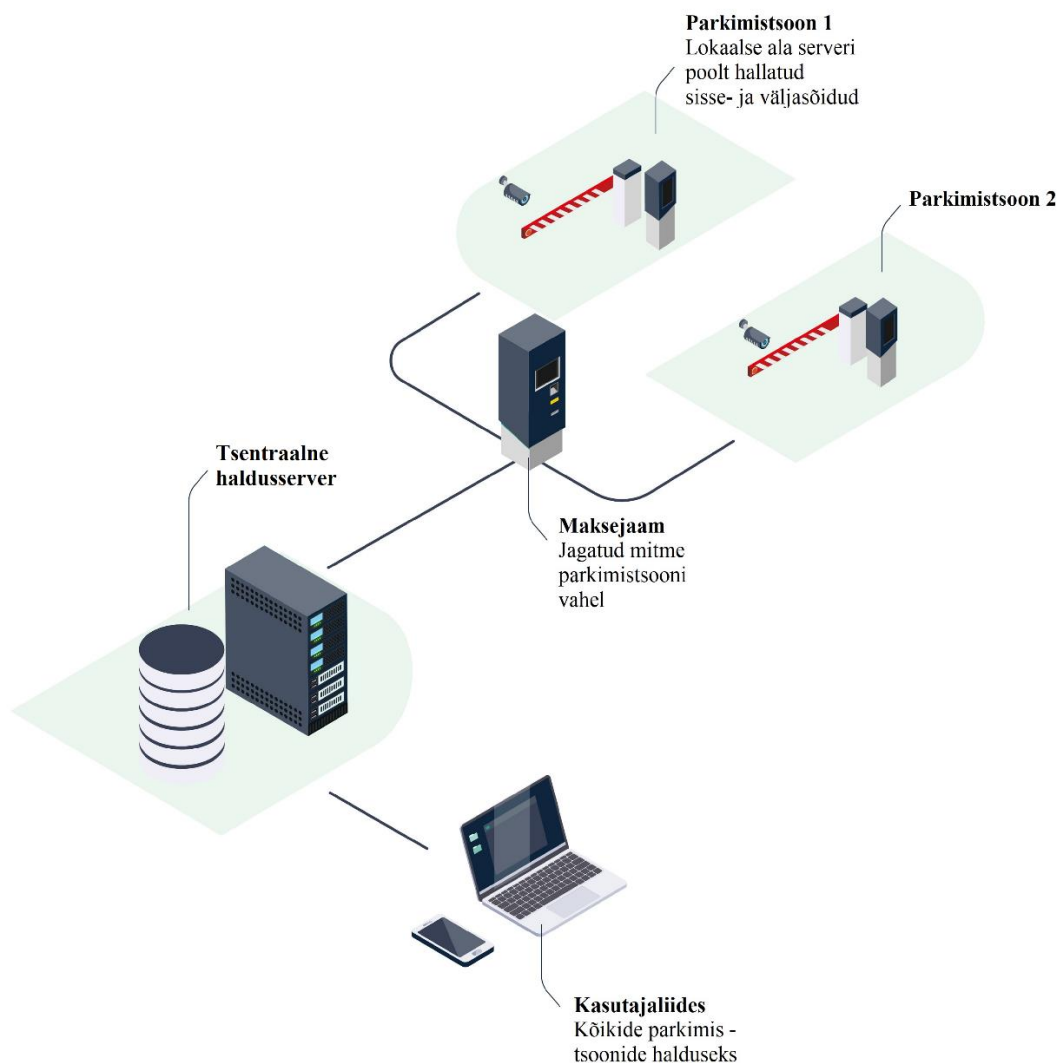
2.1.4 Maksejaam ja teised komponendid

Entringo süsteemide osaks, võib sõltuvalt parklatüübist ja parklaoperaatori soovist olla ka teisi komponente. Kõige levinum neist on maksejaam, kus parklakliendid saavad oma parkimise eest tasuda, kas kaardimaksega või sularahas. Erinevalt teistest Entringo süsteemi osadest töötab maksejaam Windowsi operatsioonisüsteemil. Paljudes riikides, k.a Läti ja Leedu, kuhu on Entringo süsteeme paigaldatud on sularaha käsitlevate süsteemide tarkvara vaja koostöös maksuametiga sertifitseerida. Seetõttu on maksejaama tarkvara uue versiooni välja laskmine harvem kui süsteemi põhikomponentide puhul.

Maksuametid seavad kohati piiranguid sularaha käsitlevate süsteemide tarkvaralistele uuendustele ning nende automatiseerimisele. Neid põhjuseid ja ettevõttepoolseid soove arvestades, otsustas autor maksejaama kui süsteemi osa, diplomitöö käigus loodava lahenduse skoobist välja jätta. Lisaks maksejaamale, võivad Entringo süsteemi koosseisu kuuluda näiteks infotablood või parkla klientidele mõeldud parkimise broneerimise lahendused ja muud integratsioonid. Kuna need on enamasti parklaoperaatori spetsiifilised lisad, mis tavapärase Entringo süsteemi kooslusesse ei kuulu, otsustas autor ka need diplomitöö käigus loodava lahenduse skoobist välja jätta [1].

2.1.5 Entringo lahenduse arhitektuuri joonis

Joonisel 1 on väljatoodud Entringo süsteemi võimalik ülesehitus.



Joonis 1. Entringo süsteemi võimalik ülesehitus [1].

Joonisel kuvatud tsentraalne haldusserver võib olla paigaldatud lokaalselt ühte parkimistsooni või kõigist parkimistsoonidest eemale, näiteks pilve. Parkimistsoonid üks ja kaks asuvad joonisel illustreeritud olukorra puhul lähestikku, näiteks võib tegemist olla ühe firma kontori töötajate ja klientide vahel eraldatud parklatega. Sõltuvalt olukorrast võivad need ka üksteisest eemal asuda. Spetsiifilisem seletus Entringo lahenduse arhitektuurist on lisatud diplomitöö hilisemates osades, kus see on asjakohane [1].

2.2 Hetkeolukorra analüüs

Entringo lahendus on hetkeseisuga üpris uudne ning agiilses ehk välearenduses, mis tõttu antakse keskmiselt iga kahe nädala tagant välja uus tarkvara versioon peamistele süsteemi komponentidele (SBO, LAS ja HAL). Seda enam tuleks uued tarkvara versioonid kliendi süsteemidele paigaldada. Selleks aga puudub hetkel automatiseeritud lahendus ning tulevikku planeeritud laiapõhjalise versiooni ja seadistuse halduse lahendus, pole veel arenduse ajakavva mahtunud, samuti puudub Entringo tarkvara esmapaigalduseks automatiseeritud lahendus. Automatiseeritud süsteemide puudumise tõttu tuleb kõik uuendused, seadistused ja paigaldused teostada manuaalselt. See on aga väga ajakulukas ja vead on kerged tekkima. Nende hulka kuuluvad näiteks kirjavead seadistusfailides või ebakorrektsed failiõigused. Piirates kuluva aja ainult töö skoobis olevatele komponentidele on keskmiselt ühe süsteemi puhul arvestatud järgmise ajakuluga: kaks tundi süsteemi uuenduseks, neli tundi esmapaigalduseks, mis hõlmab endas potentsiaalsete probleemide lahenduseks kuluvat aega ja ettevõtte kasumimarginaali. Kuna müügiprojektsioonide kohaselt on paigaldusse minevate süsteemide arv tulevikus kasvamas, tekib siinkohal probleem, nimelt süsteemide arvu kasvamisest, kasvab ka lineaarselt nende uuendustele ja esmapaigaldustele kuluv aeg. Praeguseks on kolme Balti riiki paigaldatud kokku umbes kümme Entringo süsteemi, mis on kõik varieeruvate suurustega. Kuna süsteemi tarkvara uuendused on vägagi aja- ja töömahukad, siis ei jõua need uuendused süsteemi arvule järele. Järgnev osa seletab lühidalt Entringo tarkvara paigaldust, uuendamist ja seadistust [1], [6].

2.2.1 Entringo tarkvara paigaldus

Enamasti algab Entringo tarkvara esmapaigaldus masinale tõmmise laadimisega, kus on tarkvara poolt kasutatavad lisarakendused, nagu näiteks andmebaas juba paigaldatud ja

osaliselt seadistatud. Diplomitöös käsitletud Entringo tarkvara peamised komponendid (SBO, LAS ja HAL) töötavad kõik Linuxi operatsioonisüsteemidel, enamasti Ubuntu 20.04 pikaajalise toega distributsioonil. Ekraani kasutavad HAL'id rakendavad operatsioonisüsteemiks graafilisekasutajaliidesega versiooni, teised komponendid ainult käsureaga versiooni. Peale operatsioonisüsteemi paigaldust, seadistatakse masinale korrektne IP-aadress ja teostatakse muudatused juba paigaldatud rakenduste seadetes. Viimasena paigaldatakse vastav Entringo rakendus, selle konfiguratsioonifailid ja sertifikaadid RPC autoriseerimiseks. Entringo rakenduste tarkvara töötab *systemd daemon*'ina, hakates tööle kohe peale masina käivitumist. Tarkvara esmapaigaldus võib toimuda ka kliendi poolt ülesseatud keskkonda, olgu selleks siis virtuaalmasin pilves või kliendi taristul, kohati ka füüsiline arvuti kliendi võrgus. Sellisel juhul tuleb teostada lisarakenduste paigaldus ja seadistus, mis muidu oleks operatsioonisüsteemi tõmmisega juba teostatud. Kõik see toimub automatiseerimise puudumise tõttu, manuaalselt üle turvakesta, suures osas käsk-käsu haaval ning kohati skriptidega, mis osa paigalduseks vajalikke seadistusi korruga ära teevad. Automatiseeritud lahenduse puudumisel, olid skriptid kasutusel ainult paigaldatud andmebaasi seadistamiseks ja *systemctl* teenuse loomiseks [6].

2.2.2 Entringo tarkvara uuendamine

Entringo tarkvara uuendamiseks, tuleb igasse masinasse tõsta vastavad rakenduse failide kaustad. Neis kaustades on käivitav rakenduse fail, mallid ja seadistus failid, mida rakendus loeb. Seejärel tuleb rakenduse vana versioon peatada ja muuta *systemd daemon*'it kasutamaks uusi faile või vanad failid uute vastu välja vahetada ning lõpuks uus Entringo rakendus käivitada [6].

2.2.3 Entringo tarkvara seadistamine

Entringo tarkvara seadistamine on selle uuendamisest ja paigaldamisest hõlpsam protsess, kuna sõltub oluliselt süsteemi füüsilisest paigaldusest ja töö skoobist välja jäetud komponentidest. Diplomitöö skoopi on kaasatud ainult Entringo tarkvara kolme komponendi (SBO, LAS ja HAL) ja nende lisarakenduste: PostgreSQL'i ja Elasticsearch-OSS'i seadistamine. Nii Entringo tarkvara rakendused, kui ka eelmainitud lisarakendused kasutavad seadistusfaile nende töö juhtimiseks. Neid faile tuleb vastavalt vajadusel muuta, saavutamaks soovitud süsteemi käitumine [6].

2.3 Nõuded lahendusele

Hansab Group'i poolt püstitatud nõuded antud diplomitöö käigus loodavale lahendusele on järgnevad:

1. lahendus peab kiirendama hetkel käsitsi teostatud protsessesse:
 - 1.1. tarkvara uuendamine;
 - 1.2. tarkvara seadistamine;
 - 1.3. tarkvara paigaldus;
2. lahendus peab töötama järgmiste Entringo tarkvara komponentidega:
 - 2.1. *System Back Office* – süsteemi tsentraalne haldusserver (SBO);
 - 2.2. *Local Area Server* – lokaalne ala server (LAS);
 - 2.3. *Hardware Layer* – riistvarakiht (HAL);
3. lahendus peab olema kooskõlas Entringo lahenduse arhitektuuriga;
4. lahenduses kasutatud andmeedastuskanalid peavad turvalisuseks olema krüpteeritud;
5. lahendus ei tohi põhjustada olulisi turvaauke Entringo süsteemides;
6. lahendus peab töötama Ubuntu pikaajalise toega operatsioonisüsteemiga 20.04;
7. lahendus peab kasutama internetiprotokolli neljandat versiooni (IPv4);
8. lahendust tuleb testida valideerimaks selle töökindlust, enne kliendi süsteemidel kasutamist;
9. lahendus ei tohi vajada lisa investeeringuid tarkvara litsentside või muu sarnase näol ja peaks kaasama vaid avatud lähtekoodiga osi;
10. lahenduse kood peaks olema kommenteeritud nii, et lahendusega kursis olev arendaja sellest aru saaks;
11. lahendust peab olema võimalik tulevikus edasi arendada;
12. lahendus peab esinenud vigadest selle kasutajat teavitama;
13. lahendus ei tohi vajada klientidele vastuvõetamatuid muudatusi võrgu- või kaughaldusseadetes;
14. lahenduse loomisel peaks keskenduma selle funktsionaalsusele, mitte kujundusele, luues esmalt funktsionaalsus Entringo tarkvara uuendamiseks.

2.4 Ülesande püstitus

Ettevõtte poolt püstitatud nõuetest tuletas töö autor järgneva ülesande püstituse:

1. luua automatiseeritud lahendus, mis suudab paigaldada, uuendada ja seadistada Entringo parkimis- ja läbipääsusüsteemi tarkvara Ubuntu pikaajalise toega operatsioonisüsteemil 20.04;
2. loodav lahendus peab töötama järgmiste Entringo tarkvaraliste komponentidega: SBO, LAS ja HAL;
3. loodav lahendus peab olema Entringo tarkvaralise arhitektuuriga kooskõlas ja küberturvaline.

2.5 Tarkvara keskhaldussüsteemid

Tarkvara keskhaldussüsteemid ehk tarkvara- ja konfiguratsioonihaldustööriistad on vahendid, mis võimaldavad hõlpsasti hallata suurt hulka seadmeid, võimaldades tsentraalselt määrata, mis tarkvara, peaks millisele seadmele paigaldatud olema ning millised seadistused sellel tarkvaral olema peaksid. Tarkvara- ja konfiguratsioonihaldustööriistad jagunevad oma tööpõhimõttelt kahte põhilisse kategooriasse: lükkepõhised süsteemid ja tõmbepõhised süsteemid. Diplomitöös on väljatoodud ainult keskhaldussüsteemide võimekused, mis on püstitatud ülesandele relevantsed. Järgnevalt toob autor esile mõne vabavaralise tarkvara keskhaldussüsteemi, mida hilisemates peatükkides analüüsitakse.

2.5.1 Lükkepõhised

Lükkepõhised tarkvara keskhaldussüsteemid nagu näiteks Ansible ja Puppet Bolt on ehitatud töötama ilma paigaldust vajavate agentideta ja kasutavad nende asemel ühte tsentraalset serverit, kust laetakse konfiguratsioon hallatavatele seadmetele. Samas kui Linux klientide puhul kasutatakse SSH protokollit ja avaliku võtme krüptograafiat, et saata klientidele käsked või skripte, kasutatakse Windowsi operatsioonisüsteemide puhul WinRM teenust saavutamaks samalaadset tulemust. Enamasti on sellised lahendused mõeldud eeldefineeritud seadmete grupi halduseks [7], [8].

Ansible on vabavaraline ja avatud lähtekoodiga tarkvara keskhaldussüsteem, loodud aastal 2012 ja aastast 2015 sponsoreeritud Red Hat korporatsiooni poolt. Ansible võimaldab seadistada Windows'i, Linux'i ja MacOS'i kasutavaid seadmeid. Enamus konfiguratsiooni ja automatiseerimise käsked on määratud ära YAML failides, kasutades süntaksi Python'ist, Shell'ist, Windows PowerShell'ist ja Ruby'ist. Ansible tarkvara on kirjutatud suuremas osas Python'is [7], [9].

Puppet Bolt on ülesehituselt sarnane Ansible'ga, pakkudes enamjaolt samu võimalusi ja toetades samu operatsioonisüsteeme ning kasutades seadmetele konfiguratsiooni laadimisel sarnaseid kanaleid. Olles loodud aastal 2017, viis aastat Ansible'st hiljem oli Puppet Bolt'i peamine eesmärk pakkuda Puppet'iga tuttavatele kasutajatele alternatiivi Ansible'le. Erinevalt Ansible'st on Puppet Bolt kirjutatud suuremas osas Ruby's [8], [10].

2.5.2 Tõmbepõhised

Tõmbepõhised tarkvara keskhaldussüsteemid nagu Puppet ja Chef lisavad enda töösse ka agentrakenduse, mis tuleb igasse hallatavasse seadmesse paigaldada, et saavutada ühendus klientseadmete ja tsentraalse serveri vahel. Paigaldatud agent, käib iga eeldefineeritud aja tagant serverilt konfiguratsiooni muudatuste kohta pärimas ning rakendab seejärel neid lokaalses seadmes [11], [12].

Puppet on üks laialdasema kasutusega tarkvara- ja konfiguratsioonihaldustööriistadest, olles populaarsetest alternatiividest oluliselt vanem, loodud juba aastal 2005. Puppet on võimeline haldama Windows'i, Linux'i ja MacOS'i kasutavaid seadmeid, kasutades selleks HTTPS'i suhtluses seadmele paigaldatud agendi ja tsentraalse serveri vahel. Agentide tarkvara seadistused on kirjutatud manifestidena, Puppet'i deklaratiivses keeles. Uuemad versioonid Puppet'i tarkvarast on kirjutatud peamiselt Ruby's ning toetavad manifestides Ruby domeenipõhist keelt [13], [11].

Chef on sarnaselt Puppet'ile laialdase kasutusega tarkvara keskhaldussüsteem, millel on palju sarnasusi Puppet'i uuemate versioonidega. Näiteks, kasutavad mõlemad samu protokolle agentide ja serveri vahelises suhtluses ning suudavad hallata enamjaolt samu operatsioonisüsteeme. Chef kasutab ka Ruby domeenipõhist keelt konfiguratsiooni failide loomisel, kuid manifestide asemel on Chef'i agentide konfiguratsiooni failid nimetatud retseptideks. Chef'i tarkvara on loodud aastal 2008 [12].

3 Arhitektuuri analüüs

Antud peatükis analüüsib autor, erinevate tarkvara keskhaldussüsteemide arhitektuurilist sobivust ülesande lahendamiseks ning toob esile nende peamise puudujäägi antud kontekstis. Tulenevalt sellest puudujäägist, analüüsib autor ülesande lahendamiseks loodava lahenduse arhitektuurilisi võimalusi. Diplomitöös teostatud analüüsi skooopi on kaasatud ainult töö käigus loodav lahendus ning seda otseselt mõjutavad Entringo lahenduse aspektid. Üldisemad Entringo lahenduse osad on jäetud analüüsi skoobist välja.

3.1 Tarkvara keskhaldussüsteemide analüüs

Tarkvara keskhaldussüsteemi kasutusele võtmine lihtsustaks Entringo tarkvarahaldust, kuna võimaldaks seda kergemini paigaldada, seadistada ja uuendada. Lisaks aitaks see standardiseerida Entringo süsteemide paigaldust ja seadistust, elimineerides varieeruvuse süsteemis, mis tekib manuaalseprotsessi tagajärjel. Neid aspekte kaalutledes, võiks tarkvara keskhaldussüsteem üldpildis olla sobilik vahend püstitatud ülesande täitmiseks.

3.1.1 Lükkepõhised

Lükkepõhiste tarkvara keskhaldussüsteemide peamine eelis, agendi kasutatavate tööriistade ees, on agendi paigalduse puudumine. See vähendaks oluliselt süsteemi esmaseks seadistamiseks kuluvat aega, mille kasutegur avalduks juba ülesseatud süsteemide puhul, kuhu tagantjärele agendid paigaldada tuleks. Antud tüüpi tööriistade, peamiseks probleemiks on vajadus ühendada iga Linuxit kasutatav klient seadmega üle SSH, mis tõttu tuleks Entringo süsteemi kasutatavatel operaatoritele, luua iga seadme jaoks eraldi võrgusuunamine. See oleks operaatoritele tülikas, eriti kui kasutusel on paljude seadmetega süsteemid. Lisaks oleks see küberturvalisuse vaatenurgast halb lahendus, kuna paljastaks suure hulga seadmeid välisvõrku ja seda tuleks võimalusel vältida. Suures pildis oleks võimalik püstitatud ülesannet lahendada lükke tüüpi tarkvara- ja konfiguratsioonihaldustööriistadega, nagu näiteks Ansible ja Puppet Bolt [7], [8].

3.1.2 Tõmbepõhised

Püstitatud probleemi lahendamisel on agenti kasutatavate tõmbepõhiste tarkvara keskhaldussüsteemidel, lükkepõhiste süsteemide ees eeliseks, eelmises punktis väljatoodud võrgusuunamiste vajaduse puudumine. Võrguseadistuse muutmise vajaduse puudumine, vähendab nõudeid parkla operaatoritele ning nende vältimine on ka küberturvalisuse huvides. Selle asemel, mõjutab nende kasutusele võtmist agentide paigalduse ja seadistamise vajadus, mis tõstab oluliselt esmaseks seadistuseks kuluvat aega. Agent tõmbe tüüpi tööriistadel on ka tuleviku arendustesse kaasamisel probleeme, kuna võrreldes lükketüüpi tööriistadega, tuleks agendid lahendusse kaasata ning see tõstaks oluliselt protsessi keerukust. Kokkuvõtvalt on sarnaselt lükke tüüpi süsteemidele võimalik püstitatud ülesannet lahendada siiski ka agent tõmbe tüüpi tarkvara- ja konfiguratsioonihaldustööriistadega, nagu näiteks Puppet ja Chef [11], [12].

3.1.3 Peamine probleem tarkvara keskhaldussüsteemidega

Kõikidel analüüsi kaasatud lükke- ja tõmbepõhistel tarkvara keskhaldussüsteemidel on püstitatud ülesande lahendamiseks üks oluline ja jagatud probleem. Selleks on nende peamine tööpõhimõte ja kasutusvaldkond, milleks on paljudele seadmetele standardiseeritud konfiguratsiooni laadimine [14]. Arvestades, et loodav lahendus peab suutma töötada kolme eri tüüpi Entringo tarkvara komponendiga (SBO, LAS, HAL), mis kõik kasutavad erinevat tarkvara, tuleks luua kolm eraldatud protsessi. See ei oleks suureks takistuseks, kuid arvestades, et erinevatele operaatoritele paigaldatud süsteemid ja kohati samasse parklasse paigaldatud samalaadsed komponendid, kasutavad erinevat konfiguratsiooni, siis kasvaks vajaminevate protsesside arv oluliselt ning nende süstemaatiliseks halduseks tuleks inkorporeerida kõrgetasemeline programmeerimiskeel. See seaks, koostöös keskhaldussüsteemi arhitektuurist tulenevate probleemidega, küsimuse alla tarkvara keskhaldussüsteemi kaasamise terviklahendusse. Kas kasutada kõrgetasemelist programmeerimiskeelt, lisaks tarkvara keskhaldussüsteemile või ainult kõrgetasemelist programmeerimiskeelt, töö autor tegi valiku just viimase kasuks, mis võimaldas rohkem vabadust lahenduse arhitektuurilises ja tehnoloogilistes valikutes, võimaldades paremini täita antud ülesannet.

3.2 Lahenduse arhitektuuri valik

Tuginedes eelmises alapeatükis väljatoodud analüüsi punktidele, otsustas autor vältida tarkvara keskhaldussüsteemi kasutusele võtmist ja luua uus lahendus, mis on kooskõlas Entringo parkimis- ja läbipääsulahenduste tarkvaraarhitektuuriga ning diplomitöös püstitatud ülesandega. Kaalutledes varem analüüsitud tarkvara keskhaldussüsteemide ülesehitusele on loodava lahenduse arhitektuuriks kaks peamist valikut: lükke ja tõmbe meetod. Luues nullist uut lahendust, ei ole nende kahe meetodi head ja vead samad, mis olid juba valmis süsteemide puhul, kuna autoril on oluliselt rohkem vabadust luua uus funktsionaalsus ja seeläbi vältida varasemalt analüüsitud probleeme.

Tõmbe meetodid, saab kasutada sarnaselt näiteks Puppet'ile, kus igale diplomitöö käigus käsitletud Entringo seadmele, tuleks paigaldada agent, mis suhtleks tsentraalse serveriga, käies sealt etteantud aja tagant muudatuste kohta pärimas. Agendi vaba ehk lükke meetodil töötava süsteemi kasutamisel, tuleks see kohandada Entringo arhitektuurile. Kui valmis lahendused nagu Ansible laevad seadistused otse igasse seadmesse, siis tuleks võrgusuunamiste vähendamiseks kasutada proksiserverit ehk vahendavat komponenti, suhtlemaks välisvõrgust varjatud süsteemi komponentidega. Selleks sobiks hästi Entringo süsteemi SBO, mida saaks kasutada kõikidele LAS'idele käskluste edastamiseks ning LAS'e omakorda proksina HAL'idele käskluste edastamiseks. Järgnevas tabelis 1 võrdleb autor loodava lahenduse erinevate arhitektuuride sobivust ülesande lahendamiseks.

Tabel 1. Tõmbe ja lükke meetodite võrdlus lahenduse arhitektuuri valikuks.

Valdkond	Tõmbe meetod	Lükke meetod
Esmaseadistus	Lihtne. Vajalik vaid süsteemi komponentide defineerimine ja nende hierarhia määramine.	Lihtne. Vajalik vaid süsteemi komponentide IP-aadresside ja struktuuri defineerimine.
Agendi paigaldus	Vajalik. Operatsioonisüsteemi tõmmisesse agendi paigaldamine teeks paigalduse mugavaks, teistel juhtudel tuleks aga agent paigaldada käsitsi.	Vajadus puudub.
Ülalpidamise keerukus	Lihtne. Juba kord paigaldatud agendid suudavad laadida erinevaid seadistusi ja uuendusi tsentraalsest serverist.	Lihtne. Defineeritud süsteemi struktuuri puhul on uute seadete ja uuenduste laadimine lihtne.

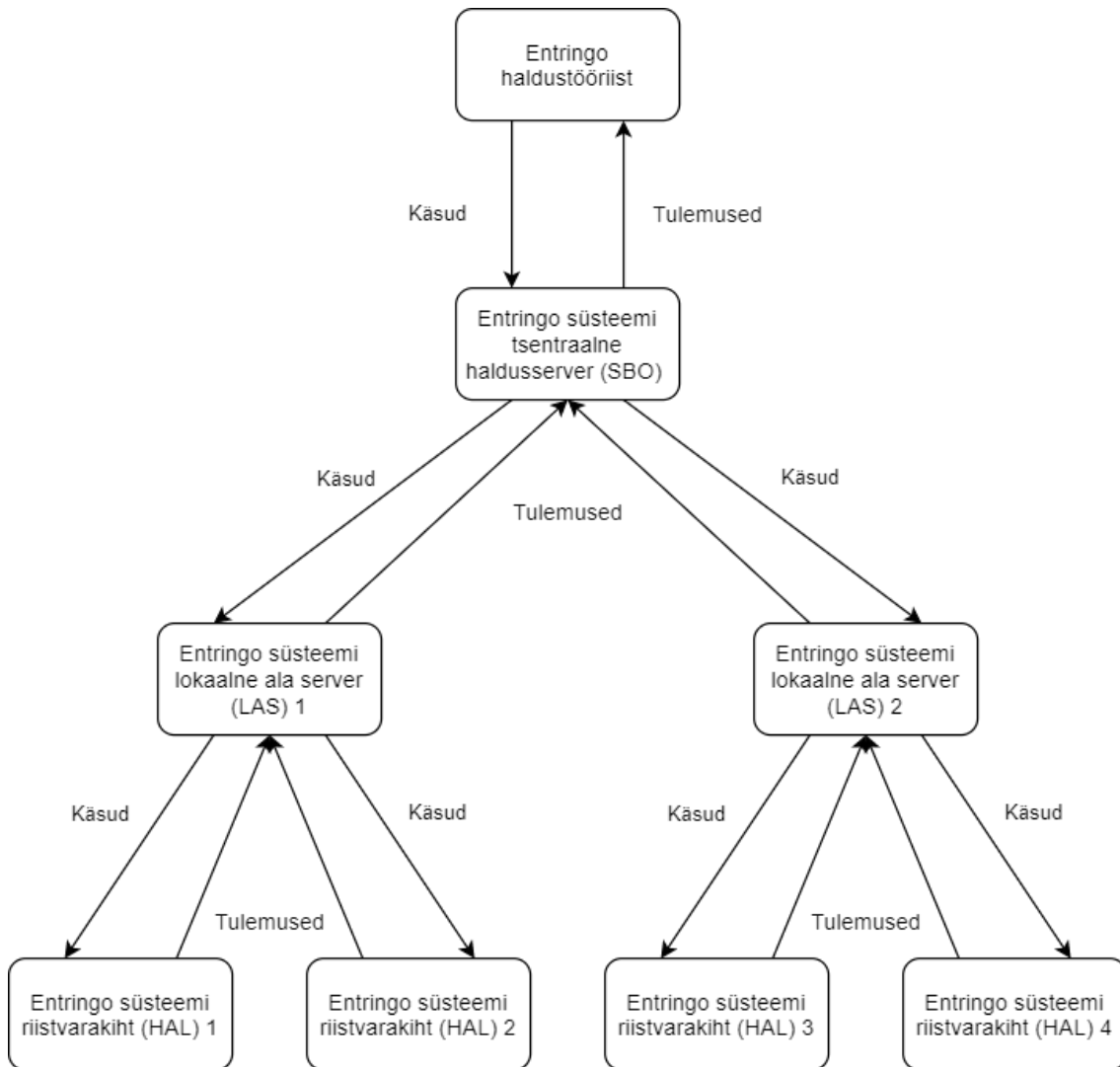
Valdkond	Tõmbe meetod	Lükke meetod
Võrgu nõuded klientidele	Väiksemad. Vajaks ainult pääsu välisvõrku, kustkaudu ühenduda tsentraalse serveriga.	Suuremad. Lahenduse tööks vajalik ühendus tsentraalsest serverist SBO'le. Mis on Entringo süsteemide kaughalduseks juba varasemalt nõutud.
Turvalisuse säilitamine	Keeruline. Tuleks luua uus või kasutusele võtta keerulisem protokoll välisvõrgus andmete ja käskluste edastuseks. Lisaks peaks ka agent olema turvaline ning selle paigaldus ettevõtte siseselt piiratud või autoriseeritud.	Lihtne. Kasutades SSH protokoll andmete ja käskude edastamiseks on andmeside hästi kaitstud.
Vigade tolerant	Madal. Kui paigaldatakse või uuendatakse agendi vigane versioon võib tekkida olukord, kus kõik agendid tuleb uuesti paigaldada.	Suurem. Vigase tsentraalse serveri puhul tuleb parandus teostada ainult selles serveris. Probleemiks kujuneb ka vigase seadistuse laadimine hallatavatele seadmetele.
Skaleeruvus	Skaleeruvam. Piiratud tsentraalse serveri võimekusega.	Vähem skaleeruv. Piiratud tsentraalse serveri võimekusega ja ka Entringo süsteemi komponentide võimsuse ja võrguühendusega, eriti määrav on SBO'd jooksvat server, kust toimub ühendus teistesse komponentidesse.
Ühildamine Entringo tarkvaraga	Parem. Võimalus integreerida halduse agent Entringo tarkvaraga. Tsentraalset serverit keeruline integreerida.	Kehv. Ainukest osa, tsentraalset serverit on keeruline Entringo tarkvarasse integreerida.
Seadistuse ja uuenduste ajastamine Entringo süsteemi komponentidel	Asünkroonne. Ajastus tuleks komponentide vahel koordineerida, tagamaks minimaalset katkestust Entringo süsteemis. Tsentraalse serveri poolel oleks selle realiseerimine keerulisem.	Sünkroonne. Lihtne hoida sünkroonset protsessi seadistuste ja uuenduste teostamiseks komponent haaval, õiges järjestuses, minimeerides katkestusi Entringo süsteemis.
Süsteemi arendamise keerukus	Keeruline. Tuleks luua kaks komponenti - tsentraalse serveri tarkvara ja sellega suhtlev agent, mis on paigaldatud igale käsitletud Entringo seadmele.	Oluliselt lihtsam. Vajab vaid tsentraalset serverit, kust käsklusi välja saata. Funktsionaalust võimalik implementeerida jupp haaval.

Selle tabeli põhjal ei saa kohe järeldusi tegema hakata, kuna mõlemal kaalutletud lahenduse arhitektuuril on eri valdkondades omad eelised ja nõrkused. Tõmbe meetodid kasutatav süsteem võimaldab realiseerida universaalsemat ja laiapõhjalisemat lahendust, mis suudaks pakkuda Entringo süsteemidele palju rohkem funktsionaalsust kui lükke meetodid kasutatav lahendus. Samas oleks sellise süsteemi arendamine tohutult ajakulukas ning diplomitöö käigus õnnestuks ajaliselt realiseerida sellest ainult väikene osa, mis ei oleks ilma edasiste arendusteta kuigi kasulik. Nendeks edasiarendusteks oleks vaja ettevõtte poolseid arenduskulutusi, mis on käesoleval aastal juba teistele protsessidele planeeritud.

Antud võimalusi koos ettevõttega kaalutledes, sai peamiseks määravaks teguriks arhitektuuri valikul, süsteemi arendamise keerukus, kus lükke meetodil töötaval süsteemil on oluline eelis tõmbe meetodid kasutatava süsteemi ees. Kulutades vähem aega süsteemi arhitektuuri realiseerimisele, saab võidetud aja ära kasutada süsteemile lisa funktsionaalsuse implementeerimiseks. Valides lükke meetodit kasutatava süsteemi on võimalik arendada süsteemi funktsionaalsust osade kaupa, mis annab võimaluse seda testkeskkonnas testida ja probleemide puudumise korral ka toodangu süsteemidel kasutada. Seega on ettevõttele kasulikum lükke meetodi põhjal töötav süsteem.

Lükke meetodi kasutusele võtmisel on oluline püsida Entringo tarkvara arhitektuuriga kooskõlas ja vältida lisa tarkvara paigaldamist vahendavale kihile. Seda vältides on võimalik elimineerida üks potentsiaalne murekoht süsteemi arhitektuuris. Lahenduse loomisel, peaks võimalusel kasutama juba olemasolevaid ühenduse meetodeid, hoides nii süsteemi kui ka selle nõudeid kompaktsetena. Süsteemi arhitektuur ei tohiks luua võimalust, Entringo tarkvara arhitektuurilistes kihtides üles liikuda, juhaks kui hierarhiliselt madalam süsteemikomponent näiteks HAL juhtub olema kompromiteerunud, ei tohi olla võimalik sellest edasi teda haldavasse LAS'i pääseda. Sama peab kehtima ka LAS'i ja SBO vahelise ühenduse puhul.

Järgnev joonis 2 näitab diplomitöö käigus loodava lahenduse arhitektuuri.



Joonis 2. Entringo haldustööriista arhitektuur.

Joonisel kuvatud Entringo süsteemi puhul on tegemist olukorraga, kus tsentraalne haldusserver haldab kahte lokaalset alaservert, mis mõlemad omakorda juhivad kahte riistvarakihti. Entringo süsteemi tsentraalne haldusserver on kasutusel haldustööriista poolt sisenemispunktina ligipääsemaks teistele seadmetele. Haldustööriista poolt välja saadetud käsklused, liiguvad mööda süsteemi hierarhilist struktuuri, tagastades tulemused tuldud teed tagasi haldustööriistani.

4 Tehnoloogiad

Tarkvara arenduse üks olulisemaid etappe on lahenduses kasutusele võetavate tarkvaratehnikate, tehnoloogiate ja arhitektuuri valik, kuna need seavad aluse edasistele tegevustele. Lahenduse arhitektuuri valik, koos selle analüüsiga on teostatud eelnevas peatükis. See jätab käesolevasse peatükki tarkvaratehnika ja tehnoloogiate valiku. Loodava lahenduse püstitatud ülesandega kooskõlas hoidmiseks, tuli autoril valida ka lahenduses kasutatavad tehnoloogiad ja tarkvaratehnika, mis ettevõtte poolsete nõuete ja piirangutega kooskõlas oleksid.

4.1 Tarkvaratehnika

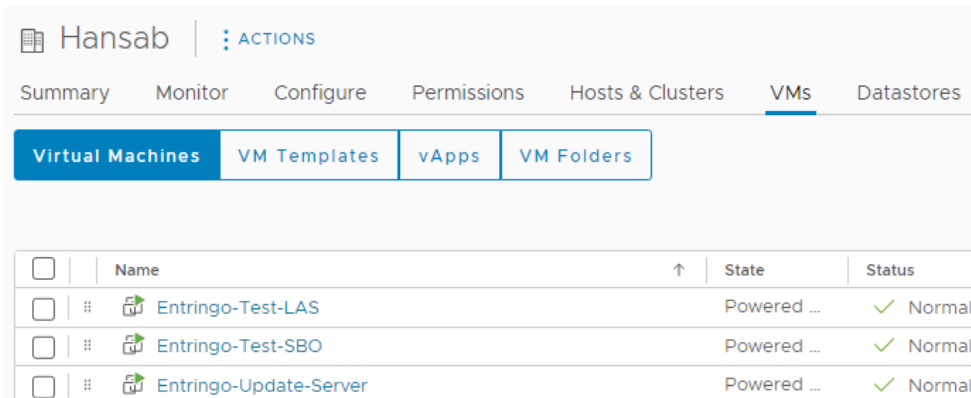
Tänapäeval on laialt levinud agiilne ehk välearendus ning sellega kaasnevad tehnoloogiad ja lähenemised, mis on kasutusel ka Entringo süsteemi tarkvara arendusel [15]. Üks välearendusest väljaarenenud meetoditest on DevOps, mis hõlmab endas rohkemat, kui vaid lihtsat tarkvaraarendust, kaasates ka tarkvaraoperatsioonid. Sarnaselt välearendusele, kasutab DevOps lähenemine lühemaid arendustsükleid kui traditsioonilised tarkvaraarendus meetodid, keskendudes kasutuskõlblike komponentide loomisele. DevOps'i kaasatud tarkvaraoperatsioonide hulka kuuluvad tihti tarkvara paigalduse automatiseerimine, mis lihtsustab paljude tarkvara versioonide paigaldamist [3].

Ettevõtte poolt püstitatud nõuetes on kirjeldatud vajadust luua esmalt Entringo tarkvara uuendamise komponent, mida saaks igapäevaselt kasutusele võtta. See vajadus elimineerib, kasutatavatest meetodikatest traditsioonilised meetodid ja soosib välearendus meetodite rakendamist. Arvestades seatud nõuetega ülesandele, valis autor kasutatavaks arendusmeetodiks DevOps'i. Autor arendas lahendust funktsionaalsete osade kaupa, hoides oma tegevuste käigus silmad lahti potentsiaalsetele lahenduse täiendustele.

Arendustegevuse käigus hoiustas autor loodud lahenduse koodi ettevõttesiseses GitLab keskkonnas, mis pakub palju funktsionaalsust just DevOps lähenemisele [16]. Enamus pakutud funktsionaalsusest jäi töö käigus kasutamata, kuna selle implementeerimine poleks ajaliselt olnud mõistlik. Küll aga kasutas autor töös GitLab'i automatiseeritud tarkvarapaigalduse võimekust, luues GitLab'i *Job* ehk töö, mis GitLab'is oleva koodi automaatselt tsentraalsele haldustööriista serverile paigaldas. Selleks oli vajalik, üles seada ligipääs ettevõttesisesest GitLab'i serverist sellele serverile ning luua skript mis rakenduse paigaldaks [17]. Rakenduse paigalduse skript oli väga lihtne, koosnedes vaid kahe faili: 'main.js' ja 'package.json' kopeerimisest serverisse ja seal ühe käsu: 'npm install' jooksutamisest.

4.2 Testkeskkond

Erinevate tehnoloogiate ja loodava lahenduse testimiseks lõi autor süsteemi testimiskeskonna, kus sai erinevaid meetodeid katsetada teiste süsteemide tööd häirimata. Testkeskkond sai loodud ettevõtte poolt kasutusel olevasse VMware vSphere 7 keskkonda ja koosnes kolmest virtuaalmasinast: esiteks tsentraalne server, kuhu diplomitöö käigus loodav tarkvara paigaldada; teiseks test Entringo süsteemi SBO ja kolmandaks test Entringo süsteemi LAS. Virtuaalmasinate kasutamine testkeskkonnana, andis füüsiliste masinate kasutamise ees eelise, kuna nende varasemat olekut oli võimalik kiiresti taastada, kasutades selleks hetktõmmiseid. Nii oli võimalik lühikese aja jooksul korduvalt protsesse testida. Kõik loodud virtuaalmasinad kasutasid Ubuntu 20.04 pikaajalise toega operatsioonisüsteemi, et püsida ettevõtte poolsete piirangutega kooskõlas. Järgnev joonis 3 näitab testimiseks loodud virtuaalmasinad VMware vSphere 7 keskkonnas.



<input type="checkbox"/>	Name	↑	State	Status
<input type="checkbox"/>	Entringo-Test-LAS		Powered ...	✓ Normal
<input type="checkbox"/>	Entringo-Test-SBO		Powered ...	✓ Normal
<input type="checkbox"/>	Entringo-Update-Server		Powered ...	✓ Normal

Joonis 3. Lahenduse testimiseks loodud virtuaalmasinad VMware vSphere 7 keskkonnas.

Lisaks joonisel kuvatud testmasinatele, oli testimisprotsessis kasutusel ka füüsilised seadmed ühe sisse- või väljasõidu imiteerimiseks, koos neid juhtiva Entringo HAL rakendusega. Erinevalt teistest Entringo süsteemi komponentidest, oli testimiskeskonna HAL paigaldatud füüsilisele Lenovo M720 seeria arvutile, mis asus näidis rajaterminalis ettevõtte kontoris. Tulenevalt selle loomise keerukusest, otsustas autor automatiseeritud testimisprotsessi mitte kasutada.

4.3 Käskluste edastamise meetod

Kuna Entringo süsteemide halduseks on kasutusel kõikidele seadmetele paigaldatud SSH serverid, siis on viimaste protokollide rakendamine, seadmetele käskude saatmiseks, ideaalne valik, mille kasutamine väldib vajadust uuteks võrgureegliteks ning tagab andmeedastuse turvalisuse.

Üle SSH protokollide andmeedastuse turvalisuse tagamiseks, otsustas autor kasutada elliptiliste kõverate (*elliptic curve*) krüptograafia Ed25519 võtme paare, mis on võrreldes standardsete RSA võtmetega sama võtmepikkuse juures, oluliselt keerulisemad murda ja tulevikukindlamad. Kasutades Ed25519 krüptograafiat on võimalik samal turvalisuse tasemel kasutada lühemaid võtmeid [18], Linux'i utiliidi *ssh-keygen* standardset võtmepikkused on Ed25519 puhul 256 bitti võrreldes RSA 3072 bittiste võtmetega [19]. Selle kaudu on võimalik saavutada kiirem võtmevahetus SSH kanali loomisel [18]. Arvestades, et diplomitöö käigus loodav lahendus hakkab kasutama SSH ühendust kõigi tasandite vahel, tuleb iga ühenduse korral läbi teha võtmevahetus, siis kumulatiivselt vähendab see seadistustele, paigaldustele ja uuendustele kuluvat aega. Kiiruse vahe jääb arvatavasti kasutajale märkamatuks, kuna Entringo süsteemid on paigaldatud modernsele riistvarale. Avalike võtmete paigalduseks, võttis autor kasutusele *sshpass* käsurea utiliidi, mis võimaldab likvideerida interaktiivsuse võtmete paigaldusel, andes utiliidile ette failis asuva Entringo seadmete esmasel paigaldusel kasutatava parooli [20]. Peale esmast võtmete paigaldust kasutab lahendus neid võtmeid kõigiks ühendusteks.

Automatiseeritud protsessis kasutatud võtmed on nende rakendamise lihtsustamiseks ilma paroolita. See põhjustab turvariski, kus saavutades ligipääsu ühte seadmesse on võimalik sellest edasi liikuda selle poolt hallatud alamseadmetele. Minimeerimaks selle riski mõju on kasutatud võtmed hierarhiliselt ära jaotatud, kasutades igal Entringo

arhitektuurilisel tasemel erinevaid võtmeid. Ühendus haldustööriistast Entringo süsteemi haldusserverisse on teostatud ühe võtmega, sealt edasi lokaalsetesse alaserveritesse järgmise võtmega. Igast alaserverist, selle poolt hallatud riistvarakihtidesse ühendumisel on kasutatud jällegi erinevat võtit. See tagab, et saades ligipääs ühte Entringo süsteemi seadmesse on sellest võimalik ainult hierarhiliselt madalamale tasemele liikuda, mis oli ka varasemalt Entringo arhitektuurist tingituna võimalik. Tagades hierarhiliselt kõrgemate süsteemi kihtide turvalisuse, eriti haldustööriista tsentraalse serveri oma on võimalik alandada ka riski kõikidele madalamatele süsteemi arhitektuurilistele kihtidele. Haldustööriista tsentraalse serveri turvalisuse tagamiseks, piiras autor tulemüüriga sellele ligipääsu ainult ettevõtte sisevõrgust ja eeldefineeritud IP-aadresside vahemikest.

Olles valinud käskluste edastamise meetodi, tuli autoril valida ka kasutusele võetav keel, milles edastatavad käsklused ja skriptid töötama peaksid. See pidid olema toetatud kõikide Entringo süsteemi arvutite ja virtuaalmasinate poolt. Lähtudes ettevõtte poolsetest nõuetest püstitatud ülesandele, peab lahendus toetama Ubuntu pikaajalise toega distributsiooni. Nii Ubuntu graafilisekasutajaliidese, kui ka ainult käsureaga distributsioonid kasutavad tavaliselt GNU Bash keelt. GNU Bash on ka diplomitöö autorile kõige tuttavam keel ning seega otsustas autor jätkata selle kasutamisega.

Süsteemi juurutamisega jätkamiseks oli autoril tarvis leida parim viis üle SSH ühenduse Bash käskude jooksutamiseks sihtmasinas. Üle SSH edastatud käsklusena on võimalik kasutada käskluste plokki ehk mitut käsklust korraga, mis on SSH käskluse poolt interpreteeritud ühe suure käsuna. Meetodid selleks on operatsioonisüsteemi keelt spetsiifilised ning neid on erinevaid. Peamised Bash keelt kasutatavad meetodid käskluste ploki edastamiseks SSH käsule on: esiteks, ülakomade või jutumärkidega piiratud käskluste plokk, teiseks, faili salvestatud käskluste plokk ja kolmandaks Here document [21].

Testides testkeskkonnas leidis autor, et parimaks meetodiks üle SSH käskluste edastamiseks on Bash'is ja ka teistes keeltes kasutatav Here document (Heredoc). Esimene, eelnevalt väljatoodud meetoditest on parim mõne käskluse edastamiseks, kuid muutub kiiresti segaseks kui käskluste arv kasvab ning kasutades keskkonnamuutujaid või parameetreid tuleb kasutajal olla ettevaatlik, millises masinas nende interpreteerimine toimub. Teine meetod on keerulisemate käskluste puhul esimesest parem, kuid vajab faili loomist. Samas kui skripti faili kasutatakse mitmel korral on see isegi kasulik omadus.

Diplomitöö käigus loodav lahendus ei kasuta enamasti sama skripti mitu korda järjest, seega võib skripti failide haldus muutuda tülikaks. Võrreldes eelneva meetodiga, võimaldab Heredoc kasutajal saata käskluste ploki otse SSH käsku, vältimaks failide või ülakomade/jutumärkide kasutusel tekkivaid probleeme. Heredoc'i kaudu saadetud käsklused laetakse üle SSH ühenduse sihtmasinasse, kus neid hoitakse ajutiselt failis, mis asub */tmp/* kaustas ning kogu fail interpreteeritakse nagu selle sisu oleks skriptina sihtmasinas käivitatud. Heredoc muutub loodavas lahenduses kasutuseks eriti kasulikuks, kui vaadelda selle võimekust hierarhiliselt käsklusi edastada. Näide loodud lahenduses, kasutatud Heredoc'i kasutavast käsklusest fiktiivsesüsteemi versiooni kontrollimiseks on toodud välja töö lisas (Lisa 2) [22].

SSH protokoll kasutamine loob ka võimaluse failide liigutamiseks ühest seadmest teise, kasutades SCP või SFTP protokollide. Mõlemad neist on ehitatud SSH turvaprotokollile, tagades sedakaudu liigutatavate failide turvalisuse. Neil mõlemal on omad eelised ja kasutusvaldkonnad. SFTP on uuem ning võimekam protokoll, andes kasutajale võimaluse, teostada lisaks failide tõstmisele ka muid failihaldus operatsioone nagu näiteks kaustade loomine ja ühenduse katkemise korral selle taastamist. SCP on võrreldes SFTP'ga lihtsam protokoll, kuid ei võimalda interaktiivselt selle kasutajal faili haldus operatsioone teostada. Teisalt on aga SCP SFTP'st kiirem, eriti kõrge latentsusega võrkude puhul, kuna ei oota vastuvõtvalt seadmelt pidevalt kinnitusi pakettide vastuvõtmisest. Loodava lahenduse puhul ei ole osa SFTP poolt pakutud lisa funktsionaalsusest vajalik, kuna sama tulemus on võimalik saavutada ka üle SSH käske saates ja interaktiivset funktsionaalsust on keeruline automaatselt rakendada. Nendel põhjustel valis autor failide edastamiseks SCP protokoll [23].

4.4 Programmeerimiskeel

Töös valitud lükke meetodil töötava arhitektuuri keskpäigas on tsentraalne haldustööriist, kust käsklused kõigile sihtmasinatele edastatakse. Sihtmasinatele saadetavate käskluste koostamiseks, otsustas autor kaasata lahendusse lihtsama programmeerimiskeele, kui seda on Bash. Diplomitöö käigus loodavat lahendust oleks võimalik realiseerida täielikult Bash'is, kuid see ei annaks lahendusele eeliseid vaid teeks selle realiseerimise ajakulukamaks. Kaasatava keele valikul kaalus autor peamiselt tema enda oskuseid, aga ka programmeerimise keele võimekust sõne töötlusel ja selle vabavaralisi valmis

komponente mida saaks ülesande lahendusse kaasata. Autoril oli varasem kogemus peamiselt C++'i ja JavaScript'iga. Võrreldes C++'i ja JavaScript'i loodavas lahenduses kasutamiseks on selge eelis JavaScript'il, kuna nagu nimigi vihjab on hilisem mõeldud just skriptimiskeelena kasutamiseks.

Ettevõtte poolsetest piirangutest lähtudes on lahenduses kasutatavas Ubuntu pikaajalise toega 20.04 operatsioonisüsteemis juba paigaldatud skriptimiskeeled Python ja Perl. Nendest kahest on Python oluliselt populaarsem ja laialdasemalt kasutatud, olles aastal 2022 kõige populaarsem programmeerimiskeel [24]. Python'is on ka loodud varasemalt analüüsitud Ansible tarkvara keskhaldussüsteem. Python oleks hea valik püstitatud ülesande lahendusse kaasamiseks, kuid autoril puudus täielikult sellega varasem kogemus. Seetõttu kuluks Python'i õppimiseks aega, mida oleks võimalik säästa kasutades autorile juba tuttavat JavaScripti. Sellel põhjusel sai lahenduse realiseerimiseks valitud JavaScript.

Täpsemalt võttis autor lahenduses kasutusele Node.js'i, mis on JavaScripti käituskeskkond ja ehitatud V8 JavaScripti mootorile, mis on kasutusel ka Chromium'i kasutatavates veebibrauserites [4], [25]. Node.js võimaldab arendajatel luua skaleeruvaid võrgurakendusi, kasutades asünkroonseid sündmustel põhinevaid protsesse. Töö autoril oli ka varasem kogemus Node.js'i kasutamisel projektides, kus see ennast õigustas. Node.js'iga koos on levitatud paketi haldur npm, mis on suurim vabavaraliste JavaScript'i moodulite kogumik [5].

4.5 Kasutatud vabavaralised teegid

Vabavaraliste tarkvara teekide kaasamine tarkvara arendusse, võimaldab arendajal vältida juba loodud lahenduste uuesti leiutamist, säästes seda kaudu arendusele kuluvat aega. Selle eesmärgiga võttis ka autor diplomitöö käigus loodud lahenduses kasutusele kolm vabavaralist teeki npm'ist. Nendeks on:

1. Inquirer.js [26];
2. node-randomstring [27];
3. ShellJS [28].

Esimene neist võimaldab hõlpsasti luua käsurea kasutajaliidest. `Inquirer.js`'i funktsionaalsusesse kuuluvad näiteks võimalused küsida rakenduse kasutajalt küsimusi ja valideerida nende vastuseid. Oma ülesehituselt kasutab see JavaScript'i *promise*'id, ehk lubadusi, mis võimaldavad asünkroonseid operatsioone järjestada ja sünkroonselt kasutajale kuvada. Teine kasutatud moodulitest on mõeldud juhusliku sõne loomiseks. See sai kasutusele võetud paroolide loomise eesmärgiga uuele Entringo süsteemile tarkvara esmapaigaldusel. Juhusliku sõne loomisel on kasutusel entroopia ehk operatsioonisüsteemi poolt kogutud juhuslikkus. `Node-randomstring` kasutab Linux'i operatsioonisüsteemidel, entroopiana operatsioonisüsteemi tuuma poolt loodud spetsiaalseid faile `/dev/random` ja `/dev/urandom`, mis tagavad loodud paroolide juhuslikkuse, elimineerides korduvalt kasutatud mustrid, mis omakorda võivad paroolide turvalisust kompromiteerida. Viimane kasutatud teekidest on Unix'i operatsioonisüsteemide käskude käivitamiseks läbi `Node.js`'i rakenduse. `ShellJS`'i kasutus lahenduses elimineerib vajaduse, ajutiste failide loomiseks ja käivitamiseks ning võimaldab käivitatud käskluste väljundit interaktiivselt suunata käsureale, kus kasutaja saab sellele anda lisa sisendeid. See loob rakendusele kaitsekihi juhtudeks, kus peaks muidu tekkima lahendusele ootamatu olukord, mille tagajärjel rakendus tagastaks vea või hanguks.

4.6 Lahenduse teisaldatavus

Üks olulisemaid valitud süsteemiarhitektuuri nõrkuseid on vajadus võrguühendusele käsklusteid väljasaatva haldusserveri ja käsitleva Entringo süsteemi SBO vahel. Diplomitöö käigus loodud lahendus on mõeldud peamiselt kasutuseks paigaldatuna ühele serverile, kust on võrgu ligipääsud kõikidesse Entringo süsteemidesse loodud. Kuid võib ka esineda olukord, kus see võimalik või mõistlik ei ole, näiteks Entringo süsteemi tarkvara esmapaigaldus olukordades, kus võrgu infrastruktuur veel paigas ei ole. Selliste olukordadega tegelemiseks otsustas autor, arhitektuurilise nõrkuse mõju vähendamiseks, luua võimalikult lihtsalt teisaldatav lahendus. Teisaldatavuse tõstmiseks lõi autor skripti, millega on võimalik loodud haldustööriista hõlpsasti paigaldada ka teistele Ubuntu 20.04 pikaajalise toega operatsioonisüsteemi kasutavatele seadmetele. Vältimaks korduvalt tehtud tööd, tuli autoril luua ka võimalus teisel seadmel tehtud toimingud, nagu näiteks uue süsteemi lisamine ja paigaldus teisaldada tagasi tsentraalsel serveril asuvasse

andmekogusse. Need võimekused olid osa põhjustest, miks autor otsustas loodud lahendusse andmebaasi kui valmis komponenti mitte kaasata.

Andmebaasi asemel rakendas autor lahenduses failide ja kaustade struktuuri, mida on kasutajal lihtne vajaduse korral käsitsi manipuleerida. Kaustade struktuur jälgib osaliselt Entringo tarkvara hierarhilist struktuuri, kaasates sinna ka lisa hierarhilisi tasemeid, nagu riik, kus Entringo süsteem asub ja selle omanik. Kaustade struktuur hoiustab Entringo süsteemi seadistusfaile ja loodud lahenduse Entringo süsteemi põhiseadetefaili, kus on määratud ära Entringo süsteemi komponentide arv, nimetused ning IP-aadressid ja pordid SSH ligipääsuks. Selline struktuur võimaldab realiseerida Entringo tarkvara seadistuse funktsionaalsust nii, nagu on kirjeldatud peatükis **5.1.3**. Struktuuri teisaldamiseks ühest seadmest, kus on paigaldatud haldustööriist teise, tuleb kasutajal tõsta rakenduse poolt loodud kaustade struktuur käsitsi ühest seadmest teise. Sellisel andmebaasi mitte kaasaval lahendusel on muidugi omad probleemid ja riskid. Nende hulka kuulub näiteks kõrgendatud oht andmelaostuseks, kuna puuduvad enamuste andmebaaside poolt kasutatavad kaitsemeetodid selle ohu langetamiseks. Samas ei ole selle riski mõju loodud lahenduse puhul kuigi suur, kuna kasutatav struktuur on lihtne ning kasutajal on võimalik tekkinud probleem käsitsi kõrvaldada. Lisaks on haldustööriista tsentraalse serveri andmed pidevalt varundatud, kasutades ettevõtte poolt kasutusel olevat Veeam Backup & Replication lahendust.

5 Lahenduse funktsionaalsus

Otsustades tarkvara keskhaldussüsteemi kasutamise vastu, tuli autoril diplomitöö käigus arendada tarkvara, mis suudaks püstitatud ülesande lahendada. Töös püstitatud ülesanne jaguneb kolmeks põhiliseks osaks koos etteantud nõuetega, mis nendele kehtivad. Põhilised püstitatud ülesande osad on loodava lahenduse võimekus paigaldada, seadistada ja uuendada Entringo parkimis- ja läbipääsulahenduse tarkvara. Autor otsustas koondada need ülesande osad ühte lahendusse, mis kõiki neid funktsionaalsuseid pakub ja nimetas diplomitöö käigus loodud lahenduse Entringo haldustööriistaks. Olles valinud loodava lahenduse arhitektuuri ja sellega kooskõlas olevad tehnoloogiad ning lähenemised, tuli autoril nende põhjal lahendus realiseerida. Käesolevas alampeatükis tutvustab autor lugejale diplomitöö käigus loodud lahendust ja selle funktsionaalsust lähemalt. Haldustööriista lähtekood on leitav GitHub'ist kasutades lisatud juhendit (Lisa 3).

Üks ettevõttepoolsetest nõuetest lahendusele oli, lahenduse valminud osade rakendamine nii pea kui need olid kasutuskõlblikud. Seda jälgides lõi autor lahenduse osade kaupa. Esmalt keskendus autor süsteemi kasutajaliidesele ja uuenduste teostamise võimekusele, kuna see oli lahenduse loomise ajal kõige tihedamini vajatav funktsionaalsus. Uuendusi teostava komponendi valmides suunas autor tähelepanu paigaldus ja seadistus komponentidele.

5.1 Kasutajaliides

Lähtudes nõudest implementeerida esmalt süsteemi funktsionaalsus, kui kasutajaliidese kujundus, võttis autor loodavas lahenduses kasutusele käsureal töötava kasutajaliidese, vähendades sedakaudu kasutajaliidese arendusele kuluvat aega. Entringo haldustööriista kasutajaliidese implementeerimiseks, kasutas autor eelmainitud npm teeki Inquirer.js, püstitades selle abil kasutajale valikuid ja küsimusi. Küsimustele, mis võimaldavad kasutajalt vabas vormis teksti vastust on lisatud sisendi kontroll, vältimaks lubamatuid tähemärke ja vales formaadis vastuseid. Lahendus on loodud töötama peamiselt

ühendudes üle SSH või muu kaughaldus kanali spetsiifilisse serverisse, kuhu tarkvara on paigaldatud. Täpne kasutajaliides võib sõltuda ühendumiseks kasutatud SSH kliendist või programmist. Haldustööriista käivitamisel, käsurealt ilma lisa argumentideta, kuvatakse kasutajale peamenüü, kust kasutaja saab valida soovitud tegevuse. Järgnevad joonised 4 ja 5 illustreerivad loodud lahenduse peamenüüd (Joonis 4) ja näidis vastustega vastatud küsimusi (Joonis 5), joonised on tehtud, kasutades Putty SSH klienti.

```
What would you like to do? (Use arrow keys)
□ Create new system
  Deploy new system
  Update existing system
  Save configuration files from an existing system
  Deploy saved configuration files to an existing system
  Check version of an existing system
  Connect to a device over SSH
(Move up and down to reveal more choices)
```

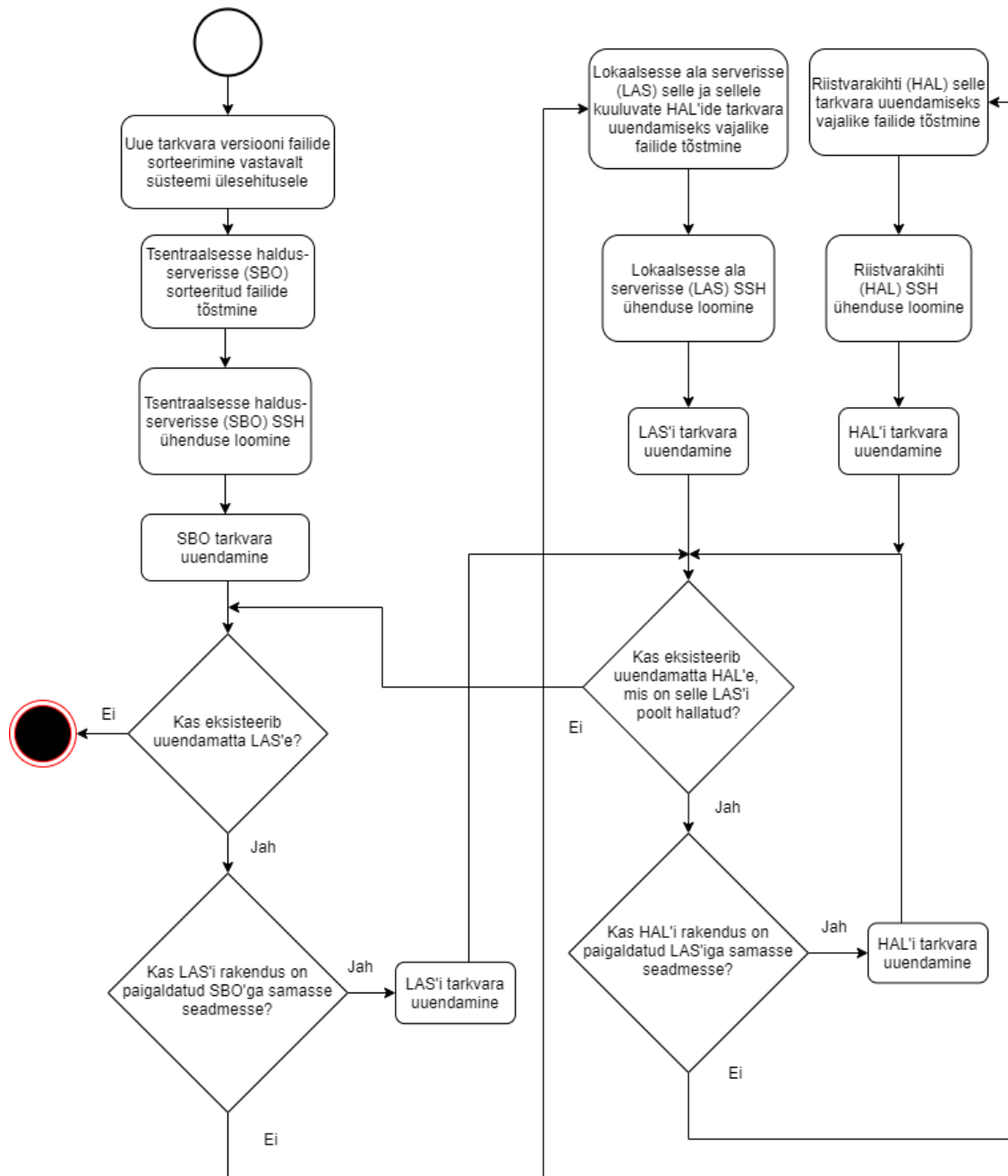
Joonis 4. Haldustööriista peamenüü.

```
Select country the system will be located in, or add a new country to the list: EE
Select the owner of the system, or add a new owner to the list: AS Hansab
Input the new Back Office name: Example Back Office
Input the Back Office IP: 192.168.1.1
Input the timezone in which the new system will be located: Europe
>> Please enter a valid timezone, for example: "Europe/Tallinn"
```

Joonis 5. Näidis valikud haldustööriista kaudu uue süsteemi lisamisel.

5.2 Entringo tarkvara uuendamine

Alustades lahenduse realiseerimist Entringo tarkvara uuendamise funktsionaalsuse implementeerimisega, tuli autoril arvestada ka lahenduse korduvkasutusega Entringo tarkvara seadistuse ja paigalduse komponentide implementeerimises. Lahenduse koodi loetavuse ja loogilise ülesehituse säilitamiseks, tuli osade komponentide korduvkasutus hüljata. Varasemalt kirjeldatud Entringo tarkvara uuendamise protsess on oma põhimõttelt lihtne. Peamine tegevus tarkvara uuendamiseks on vana tarkvara versiooni failide vahetamine uute vastu. Automatiseeritud protsessi käigus, luuakse uuendatava Entringo süsteemi struktuuri baasil sõne, mis koosneb uuenduseks vajalikest käsklustest. Sõne valmimisel käivitatakse see Bash käsklusena. Tarkvara uuenduse protsessi saab kasutaja käivitada läbi haldustööriista peamenüü, valides seejärel uuendamisele kuuluva Entringo süsteemi. Joonis 6 kirjeldab automatiseeritud Entringo tarkvara uuenduse protsessi.



Joonis 6. Entringo tarkvara uuendamise automatiseeritud protsess.

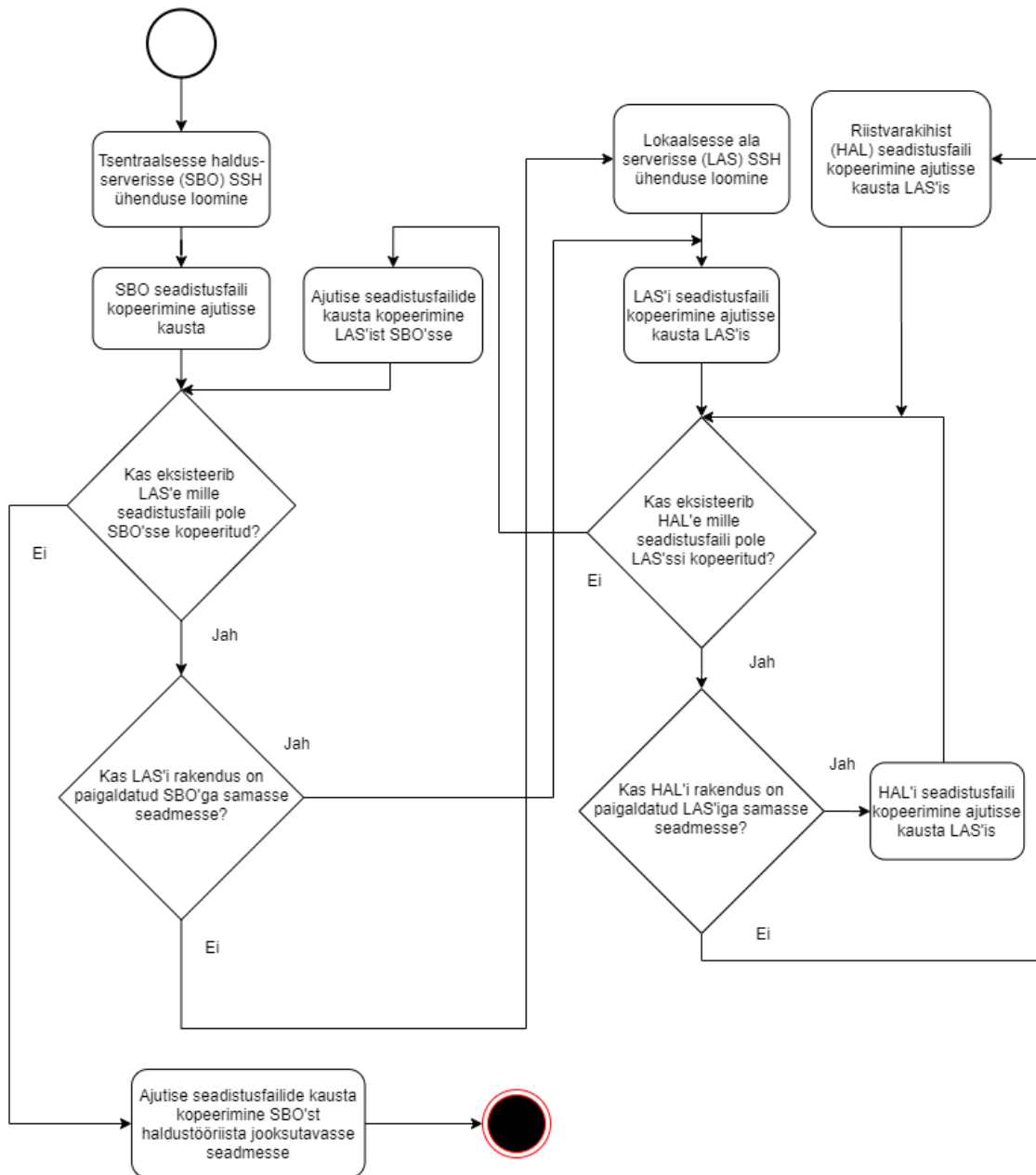
Koos Entringo süsteemi uuendamise funktsionaalsusega, lõi autor võimaluse, kontrollida hetkel süsteemile paigaldatud tarkvara versiooni. See töötab sarnaselt uuenduse protsessile, kuid failide tõstmine selle puhul vajalik ei ole ning tarkvara uuendamise asemel kontrollitakse hoopis selle versiooni. Tuvastatud Entringo tarkvara versiooni, kuvatakse tarkvaralise komponendi kaupa kasutajale olukordadeks, kus kõik süsteemi komponendid ei ole samal versioonil, mis võimaldab kontrollida versiooni uuenduse õnnestumist. Uuenduse käigus tekkivad vead kuvatakse sellegipoolest kasutaja käsureale, võimaldamaks probleeme kohe tuvastada ning lahendada.

5.3 Entringo tarkvara seadistamine

Töös käsitletud Entringo tarkvara seadistamist on keeruline ühe automatiseeritud protsessina realiseerida, kuna see hõlmab lisaks Entringo tarkvaralistele komponentidele ka lisarakenduste: PostgreSQL'i ja Elasticsearch-OSS'i seadistamist. Nende rakenduste seadistamine sai haldustööriistas kokku viidud tarkvara paigalduse protsessiga, kuna seadistused tuleb teostada enne esimest Entringo tarkvara käivitust ja hilisemad muudatused toimuvad ainult erandolukordades. Seega saab reeglina hilisemaid tegevusi piirata Entringo rakenduste seadistusfailide muudatustele. Kõik Entringo rakendused kasutavad seadistuste lugemiseks faili 'config.json', mis peab asuma rakendusega samas kaustas. Seega antud faili ülesehitus ja selles määratud väärtused on rakenduse spetsiifilised ehk iga rakendus kasutab erinevat faili. Loodud haldustööriist pakub Entringo seadistusfailidega tegutsemiseks järgnevaid võimalusi:

1. süsteemile hetkel paigaldatud seadistusfailide laadimine lokaalsesse masinasse;
2. süsteemile uute seadistusfailide laadimine;
3. süsteemi komponendiga üle SSH ühenduse loomine.

Neist esimesed kaks on loodud lihtsustamaks Entringo tarkvara seadistusfailide haldust. Laadides paigaldatud seadistusfailide koopiad lokaalsesse masinasse, kust haldustööriist käivitati, saab kasutaja neid hõlpsamini muuta, kasutades selleks võimekamaid graafilisi rakendusi, mida pole käsureal võimalik rakendada. Peale soovitud muudatuste teostamist, saab kasutaja muudetud failid Entringo süsteemile laadida, kus need koheselt rakendatakse. Joonisel 7 on illustreeritud automaatset Entringo tarkvara seadistusfailide laadimist lokaalsesse seadmesse. Protsessi käigus koondatakse parkla kaupa, kõikide sellesse parklasse kuuluvate Entringo seadmete (HAL ja LAS) seadistusfailid kokku parklat haldavasse LAS'i. Sealt edasi laetakse kõikide Entringo põhiliste tarkvaraliste komponentide (SBO, LAS ja HAL) seadistusfailid kokku seadmesse, kuhu on paigaldatud SBO rakendus ning seejärel laetakse need seadmesse kust haldustööriist käivitati.



Joonis 7. Protsess Entringo tarkvara seadistusfailide laadimiseks lokaalsesse seadmesse.

Uute seadistusfailide laadimine Entringo süsteemi seadmetele on ülesehitatud nende lokaalsesse masinasse laadimise vastandina. Selle käigus sorteeritakse seadistusfailid hierarhiliselt ära ja laetakse seadmest, kus haldustööriist käivitati esmalt Entringo süsteemi haldavasse serverisse, kuhu on paigaldatud SBO. Sealt edasi laetakse LAS'i ja HAL'ide seadistusfailid parklat haldavasse seadmesse ehk sinna, kuhu on paigaldatud LAS. LAS'ist tõstetakse HAL'ide seadistusfailid edasi HAL'i jooksutavatesse seadmetesse.

Viimane võimekus ehk süsteemi komponendiga üle SSH ühendamine, sai loodud haldustööriista arenduse käigus, võimaldamaks testkeskkonnas katsetades, kontrollida sihtmasinas haldustööriista poolt tehtud muudatusi. Arutades ettevõttega selle funktsionaalsuse rakendamist teistes olukordades, osutus see väga kasulikuks ka igapäevastes süsteemihaldus toimingutes. Nimelt, võimaldab see haldustööriista kasutajal kiiresti ühenduda soovitud seadmega üle turvakesta. Selle funktsionaalsuse puududes, tuleb kasutajal kõigepealt ühenduda seadmesse, kuhu on paigaldatud Entringo SBO ja sealt alles järgmistesse seadmetesse. Sõltuvalt süsteemi ülesehitusest, võib ühenduse protsess seadmesse, kuhu on paigaldatud Entringo HAL rakendus, vajada kuni kahte lisa SSH ühendust, kõigepealt seadmesse kuhu on paigaldatud süsteemi SBO, sealt edasi seadmesse, kuhu on paigaldatud seda HAL'i haldav LAS ja lõpuks seadmesse, kuhu sooviti ühenduda. Kasutades loodud haldustööriista SSH ühenduse loomiseks soovitud seadmesse, tuleb kasutajal kõigest valida, mis seadmesse ta ühenduda soovib ning seejärel teostatakse kõik vahepealsed SSH ühendused automaatselt. Selline meetod säästab kasutaja aega ja vähendab käsitsi ühenduste loomisega võrreldes, vigade tekkimise tõenäosust.

5.4 Entringo tarkvara paigaldamine

Entringo haldustööriistas on tarkvara paigaldamise protsess jagatud kaheks komponendiks: uue süsteemi lisamine ja uuele süsteemile tarkvara paigaldamine. Antud eraldus on tehtud, võimaldamaks olukordasid, kus uue süsteemi haldustööriista lisamine ja uue süsteemi paigaldus ei toimu ühel ajal. Uue süsteemi lisamine koosneb samuti kahest osast. Esimeses, küsitakse kasutaja sisendit uue süsteemi loomiseks ning teises, luuakse selle sisendi põhjal ettenähtud süsteemi struktuur. Uue süsteemi sisestamiseks küsitakse kasutajalt järjest küsimusi näiteks komponentide arv, seadmete IP-aadressid ja seadmete nimed süsteemi ülesehituse ja seadistuste kohta. Pärast kõikidele küsimustele vastamist liigutakse edasi teisele osale, kus teostatakse süsteemi struktuuri loomiseks järgnevad tegevused:

1. süsteemipõhise kaustade struktuuri loomine;
2. süsteemi seadistuse loomine;
3. süsteemi ja haldustööriista seadistusfailide kirjutamine;
4. süsteemi sertifikaatide loomine.

Lahendus võimaldab kasutajal, kas valida või soovitud valiku puudumisel, sisestada Entringo süsteemi paigalduseks riiki nt Eesti, Läti või Leedu ja ühtlasi ka süsteemi omaniku. Neid kasutab haldustööriist süsteemi otsingu lihtsustamiseks. Pärast riigi ja omaniku määravate kaustade loomist või valikut, loob haldustööriist kasutaja sisendi põhjal, Entringo tarkvaralise arhitektuuriga kooskõlas kaustade struktuuri, kuhu hiljem tarkvaralise komponendi põhilised seadistusfailid kirjutatakse. Enne nende failide kirjutamist, tuleb luua nende sisu, kasutades selleks seadistusfailide malle, mis sisaldavad seadistusfaili struktuuri ja vaikeväärtuseid. Kasutaja sisendit enamuste seadistusfaili väljade väärtusteks ei küsita, vaid rakendatakse automaatselt vaikeväärtuseid. Viimaste rakendamine on reeglina kasulik, kuna valdav enamus diplomitöö kirjutamise hetkel paigaldatud Entringo süsteeme, kasutavad samuti neid vaikeväärtuseid. Peale uue süsteemi loomist on kasutajal võimalik rakendatud vaikeväärtused, vajaduse põhiselt ära muuta. Viimane uue süsteemi loomise osa on Entringo tarkvara poolt kasutusel olevate sertifikaatide loomine. Selleks kasutab haldustööriist *openssl* käsurea rakendust [29]. Peale uue süsteemi lisamise protsessi lõppu, küsitakse kasutajalt, kas ta soovib kohe süsteemile ka Entringo tarkvara paigaldada.

Haldustööriist eeldab, et seadmetele kuhu Entringo tarkvara paigaldatakse on juba paigaldatud Ubuntu 20.04 operatsioonisüsteemid, määratud IP-aadressid, käivitatud SSH teenused ja tehtud mõned teised vajalikud seadistused. Automatiseeritud Entringo tarkvara paigalduse protsess jaguneb järgnevatiks osadeks:

1. SSH võtmepaaride loomine ja paigaldus;
2. paroolide loomine ja vahetamine;
3. kaustade loomine ja õiguste seadistamine;
4. sertifikaatide paigaldus;
5. PostgreSQL'i ja Elasticsearch-OSS'i paigaldus ja seadistus;
6. Entringo rakenduste ja nende seadistusfailide paigaldus.

Olles käskluste edastamise kriitiline lüli, tuleb edasiste tegevuste võimaldamiseks esmalt luua ja seejärel, paigaldada Entringo süsteemi seadmetele vajalikud SSH võtmepaarid. Selle automatiseerimiseks on kasutusel *sshpas* ja *yes* utiliidid, mis võimaldavad eemaldada võtmete paigaldamiseks vajaliku kasutaja sekkumise. *Yes* on kasutatud Ubuntu operatsioonisüsteemi standardne osa, kuid *sshpas* utiliit tuleb enne käsitsi paigaldada käsuga 'sudo apt-get install sshpas -y' [20]. Kui käskude edastamise funktsionaalsus on

Entringo süsteemi seadmetel võimaldatud, luuakse seadmetele uued paroolid. Seadmetele loodud paroolid on tähtnumbrilised ja 32 tähemärki pikad. Loodud paroolid jagunevad kolmeks, vastavalt Entringo tarkvara komponentide järgi. Süsteemi seadmele paigaldatakse parool, vastavalt Entringo arhitektuuri hierarhias kõige kõrgemal asuval tarkvaralisele komponendile, mis on vaja sellele seadmele hiljem paigaldada. Paigalduse käigus salvestatakse need paroolid faili, mis asub seadmes, kus haldustööriist käivitati. Peale paigaldust tuleks tööriista kasutajal see fail kustutada ja selles olevad paroolid lisada ettevõtte sisesesse parooli haldustarkvarasse Team Password Manager, kus ligipääs nendele on võimaldatud ainult teatud kasutajatele. Autor kaalutles koos ettevõttega ka paroolide asendamise võtmepaaridega, kuid otsustas selle vastu. Kuna kasutades paroolihaldustarkvara on paroolide jagamine mitme kasutaja vahel, oluliselt lihtsam kui võtme failide majandamine. Ühe süsteemi paroolide lekke korral on lihtsam olukord kontrolli alla saada, vahetades süsteemil kasutatud paroolid ära, võrreldes lekkinud privaatvõtmega, mis võib olla paigaldatud kõikidesse süsteemidesse ja tuleks lekke korral sealt eemalda.

Paigalduse protsess jätkub, Entringo süsteemi failide ja rakenduste jaoks, kaustade loomisega sihtseadmetes ning nende kaustade õiguste piiramisega. Loodud kaustadesse paigaldatakse esimesena, Entringo rakenduste poolt kasutatavad sertifikaadid. Need jaotatakse seadmetesse laiali tagades, et seadmes oleksid paigaldatud ainult sertifikaadid, mida sellese seadmesse paigaldatud Entringo rakendus või selle rakenduse poolt hallatud teised Entringo rakendused vajavad. Pärast sertifikaate, paigaldatakse Entringo süsteemi tsentraalsele haldusserverile ja lokaalsetele ala serveritele, Entringo rakenduste tööks vajalikud lisarakendused. Tsentraalsele haldusserverile paigaldatakse nii PostgreSQL, kui ka Elasticsearch-OSS, lokaalsetele ala serveritele ainult PostgreSQL. Nende rakenduste paigaldusel kontrollitakse rakenduste varasemat olemasolu, vältimaks olemasolevate andmete laostust või üle kirjutamist. Olukordades, kus Entringo SBO ja LAS rakendused on paigaldatud samale seadmele, teostatakse topelt paigalduse asemel PostgreSQL'is lisaseadistused. Tehtavate PostgreSQL'i seadistuste alla kuuluvad ajatsoonide muudatused, andmebaasi tabelite loomine ja nende seadistus. Viimaseks paigaldatakse süsteemile Entringo rakendused, koos eelnevalt loodud seadistus failidega. See protsess on oma ülesehituselt väga sarnane varasemalt töös kirjeldatud Entringo tarkvara uuenduse protsessile, kuid tarkvarauuenduse teostamise asemel tehakse selle paigaldus.

6 Lahenduse analüüs

Diplomitöö käigus loodud lahenduse analüüsi olulisemaks aspektiks on püstitatud ülesande teostamine loodud lahendusega ja lahenduse kooskõla ettevõtte poolt püstitatud nõuetega. Selgitamaks välja diplomitöös loodud lahenduse kooskõla osas **2.3** püstitatud nõuetega ja nende põhjal osas **2.4** püstitatud ülesandega, testis autor lahenduse komponente kõigepealt testkeskkonnas ja seejärel võimaluse korral kliendi süsteemidel. Haldustööriista poolt toimingute teostamiseks kuluva aja mõõtmiseks, kasutas autor Linux'i käsku *time*. Kriteeriumide puhul, mida testida polnud võimalik on lisatud analüüs ja sellel põhinev autori arvamus.

6.1 Testimise tulemused

Testides loodud haldustööriista võimekust, teostada Entringo süsteemi uuendusi testkeskkonnas selgus, et loodud lahendus on võimeline edukalt tavaolukordades Entringo peamisi tarkvara komponente (SBO, LAS ja HAL) uuendama. Testkeskkonnas kulus uuenduse teostamiseks reeglina aega alla minuti, mis oli tingitud süsteemi väiksest suuruselt ning väga suurest andmeedastuskiirusest, mis on saavutatav samal serveril asuvate virtuaalmasinate vahel. Peale edukat testimist, võeti haldustööriist ettevõtte poolt kasutusse süsteemi uuenduste teostamiseks kliendi süsteemidel. Siinkohal tekkisid mõningad probleemid, millede peamiseks põhjuseks oli vanemate süsteemide ebastandardne paigaldus, kuna Entringo tarkvara paigaldamise tavad on ajaga muutunud. Haldustööriist ei arvestanud varasemate paigaldus tavadega, mistõttu tuli vanemad süsteemid uutele standarditele üle viia. Selle teostas autor mõne lihtsa skripti abiga. Kui tarkvara paigaldus sai vanemate ja uuemate süsteemide vahel standardiseeritud, suutis haldustööriist ka kliendi süsteeme edukalt uuendada. Ajakulu uuenduse teostamiseks oli sõltuvalt süsteemi suuruselt ja võrguühenduse kiirusest 1-15 minutit. Võrreldes originaalse, käsitsi uuenduse teostamiseks arvestatud ajakuluga (kaks tundi) on vahe drastiline. Automatiseeritud lahendus ei tee süsteemi uuenduste teostamisel ka vigu, mis

võiksid käsitsi teostamisel tekkida, tagades sedakaudu Entringo süsteemi töö taastumise pärast uuendust.

Haldustööriista Entringo tarkvara seadistuse võimekuse testimine oli võrdlemisi lihtne, kuna tarkvara uuenduse võimekuse testimisel selgunud probleemid, said selleks ajaks lahendatud. Testimine oli nii testkeskkonnas, kui ka kliendi süsteemidel edukas, võimaldades kasutajal lihtsamini hallata Entringo tarkvara seadistusfaile. Ajavõitu on siinkohal keeruline ennustada, kuid protsess on varasemast kindlasti lihtsam ja peaks sellepärast ka vähem ajakulukas olema. Koos automatiseeritud uuenduse funktsionaalsusega, võimaldab seadistuse võimekus, ettevõttel oma klientidele kiiremat ja kvaliteetsemat teenust pakkuda, hoides samas kokku ka ettevõttesiseselt sellele kuluvat aega. Lisaks ei nõua haldustööriista kasutamine selle kasutajalt nii palju teadmisi Entringo ja Linux'i süsteemide tööst, kui varasemalt käsitsi teostatud protsessid. Need eelised parandavad klientide kogemust Entringo süsteemidega ja vähendavad ühtlasi ka ettevõtte kulusid.

Entringo tarkvara paigaldamise funktsionaalsust oli diplomitöö kirjutamise ajaks, autoril võimalik testida vaid testkeskkonnas, kuna ühtegi uut süsteemi selle loomise ajal klientidele ei paigaldatud. Testkeskkonnas olid katsed edukad, mis tähendab, et lahendus oli suuteline teostama kõiki eelmainitud Entringo tarkvara paigalduseks vajalikke samme. Testkeskkonnas võttis tarkvarapaigaldus aega ligikaudu kolm minutit. Enamus sellest kulus lisarakenduste PostgreSQL'i ja Elasticsearch-OSS'i paigalduseks. Päris süsteemide puhul oleks paigalduseks kuluv aeg testkeskkonnaga võrreldes suurem, kuna enamasti koosnevad päris süsteemid rohkematest komponentidest kui kasutatud testkeskkond. Lisaks mõjutab paigalduseks kuluvat aega võrguühenduse kiirus. Suurem eelis on automatiseeritud lahenduse veakindluses, kuna just tarkvara paigaldamine osutus käsitsi teostatuna kõige probleemsemaks tegevuseks. Hõlmates endas teistest tegevustest rohkem osi, mis pidid kõik olema korrektselt teostatud enne, kui Entringo tarkvara edukalt käivitada sai. Seega võib järeldada, et loodud lahendus kiirendab varasemalt käsitsi teostatud protsesse ja kaasab peamisi Entringo tarkvara komponente (SBO, LAS ja HAL) ehk 1. , 2. ja 8. nõue lahendusele on täidetud.

Lisaks uuris töö käigus autor erinevaid võimalusi, loodavat lahendust Entringo lahenduse arhitektuuriga kooskõlas hoida ja selle kaudu vältida klientidele vastuvõetamatuid muudatusi võrgu või kaughaldus seadetes ning järeldab, et ka 3. ja 13. nõue lahendusele

on täidetud. Kuna kõik testkeskkonna masinad, ja ka kliendi süsteemide seadmed, kasutavad Ubuntu 20.04 pikaajalise toega operatsioonisüsteemi nii graafilisekasutajaliidese, kui ka ainult käsurea võimekusega ning töötavad IPv4 protokollil, võib järeldada, et lahendus täidab ka nõudeid 6 ja 7. Loodud lahenduses on kasutatud ainult vabavaralisi mooduleid, mis ei vaja lisa investeeringuid, seega 9. nõue lahendusele on täidetud. Lahenduse loomisel, lõi ja rakendas autor, esmalt tarkvara uuendusi teostava komponendi ning piirdus kasutajaliidese disainiga vaid käsurea rakendusele, sedakaudu täites nõude 14. Vigade tekkimise korral teavitab loodud lahendus nendest kasutajat, võimaldamaks probleemide tuvastamist. Sellest järeldab autor 12. nõude täidetust.

6.2 Turvalisus

Loodud lahenduse mõju Entringo süsteemide turvalisusele saab jagada kaheks aspektiks: esimene neist on lahenduse enda turvalisus. Seda aspekti analüüsis autor diplomitöö eelpooltoodud osades. Peamine tuvastatud haldustööriistast tingitud turvarisk on paroolita SSH võtmepaaride kasutamine, mille abil on võimalik pääseda ligi kõikidele Entringo süsteemide komponentidele. Vastava riski madaldamiseks kasutas autor Entringo lahenduse arhitektuuriga kooskõlas olevaid võtmeid, mis on eri hierarhilistel tasemetel erinevad. Siiski on ülimalt oluline, tagada turvalisus seadmes, kuhu haldustööriist paigaldatakse. Kasutades haldustööriista peamiselt ühel serveril, millele ligipääs on piiratud on võimalik maandada kasutatud SSH võtmetest põhjustatud riski veelgi. Haldustööriistas kasutatavate põhitehnoloogiate valikul ja rakendamisel lähtus autor Entringo tarkvara arhitektuurist ja sellega eos kaasnevatest turvariskidest ning kaalutles, lahenduses kasutatud arhitektuurist tulenevaid turvariske, millest järeldub, et loodud lahendus uusi olulisi turvariske ei põhjusta.

Teine turvalisuse aspekt on tänu automatiseeritud lahendusele, saavutatud turvaeelised käsitsi teostatud protsesside ees. Olulisim neist on Entringo süsteemide standardiseerimine ja parem ülevaade süsteemidel tehtud seadistustest. Automatiseeritud lahendus on võimeline kiiresti teostama ligipääsu piirangute seadistusi komponentidele nagu andmebaasid, mis käsitsi oleksid ajakulukad ja seetõttu, võidakse nendes süvenemise asemel, rakendada vähem piiravaid üldistatud seadistusi. Loodud lahendus parandab ka Entringo süsteemidel kasutatud paroolide turvalisust, kuna varasemalt olid

tihti kasutusel mugavad ja korduvad paroolid kõigil ühe süsteemi seadmetel. Selliste paroolide asemel loob ja kehtestab haldustööriist uue süsteemi paigaldusel turvalised paroolid, mis on hierarhiliselt jaotatud. Autor järeldab turvalisuse analüüsist, et nõuded 4 ja 5 lahenduse turvalisusele on täidetud.

6.3 Tuleviku võimalused

Tulevikus plaanib autor, diplomitöö käigus loodud haldustööriista ka edasi arendada, kuna selle funktsionaalsus osutus ettevõttele väga kasulikuks ning selles nähti tuleviku potentsiaali. Peamiseks planeeritud funktsionaalsuseks on automaatne uuenduste laadimine kõikidele süsteemidele, ilma kasutaja sekkumiseta. Seda plaanib autor realiseerida, kasutades diplomitöö käigus loodud uuenduse ja versiooni kontrolli funktsionaalsuseid. Selle võimaldamiseks seadis autor haldustööriista Entringo tarkvara uuendamise ja versiooni kontrolliprotsessi ülese vastavalt. Täisautomaatse uuenduse protsessi, saaks ka integreerida Entringo tarkvara versioonide väljastamise protseduuriga, kus peale uue versiooni testimist, saaks testija määrata, kas uus versioon on kasutuskõlblik. Kui testija annab versioonile kinnituse, laetakse see automaatselt haldustööriista serverisse ja käivitatakse järgmisel ööl uuendused kõikide Entringo süsteemide tarkvarale.

Olukordades, kus automaatne uuendus kasutaja teadmata ebaõnnestub, võib see oluliselt häirida Entringo süsteemi tööd, mis põhjustaks kliendi pahameelt ja langetaks arvamust Entringo süsteemidest üldiselt. Selline olukord omakorda kahjustaks ettevõtte ärilisi huvisid ja seega tuleks seda igas olukorras vältida. Selleks tuleks enne täisautomaatse uuenduste laadimise implementeerimist, tõsta uuendusprotsessi veakindlust, kuna siis ei ole kasutaja aktiivselt sellega seotud ehk protsess peab ise kõikides olukordades hakkama saama. Veakindluse tõstmiseks on vaja protsessile lisada erinevaid kontrollivaid ja valideerivaid osi, mis kindlustaksid Entringo süsteemi töötamise peale selle uuendust, või uuenduse ära jätmise olukordades, kus see võiks süsteemi tööd häirida.

Autori arvamusel on diplomitöö käigus loodud haldustööriist seatud üles tuleviku kindlal viisil ja selle kood ka piisavalt kommenteeritud. Seega ta järeldab, et nõuded 10 ja 11 on täidetud.

Kokkuvõte

Diplomitöö eesmärgiks oli lahendada Entringo piletivaba parkimis- ja läbipääsulahenduse automatiseeritud paigaldamise, uuendamise ja seadistamise võimekuse puudus, mis põhjustas üleliigset ajakulu ning oli veaaldis. Selleks püstitas autor järgneva ülesande: „Luu automatiseeritud lahendus Hansab Group'i poolt loodud Entringo parkimis- ja läbipääsulahenduse tarkvara paigaldamiseks, uuendamiseks ja seadistamiseks Ubuntu pikaajalise toega operatsioonisüsteemidel, mis oleks Entringo tarkvaralise arhitektuuriga kooskõlas ja turvaline“.

Käesolevas töös tutvustas autor lugejale ülesande tausta, sellele Hansab Group'i poolt seatud nõudeid ja tarkvara keskhaldussüsteeme. Ühtegi neist tarkvara keskhaldussüsteemidest autor analüüsi tulemusena ülesande lahendusse ei kaasanud. Selle asemel lõi autor Entringo haldustööriista, mille arhitektuur on inspireeritud Ansible'st, kuid adapteeritud Entringo lahenduse arhitektuurile. Autor tutvustas lugejale haldustööriista loomisel kasutatud tehnoloogiaid ja lähenemisi ning selle funktsionaalsust.

Haldustööriist võeti Hansab Group'i poolt kasutusele klientidele müüdüd Entringo süsteemide uuendamiseks ja seadistamiseks, kus see vähendas oluliselt nendele toimingutele kulunud aega. Uutele Entringo süsteemidele tarkvara paigalduseks võetakse haldustööriist kasutusele järgmise Entringo süsteemi ettevalmistusel.

Diplomitöö tulemusena loodud Entringo haldustööriist on võimeline uuendama, seadistama ja paigaldama Entringo tarkvara peamiseid komponente (SBO, LAS ja HAL), olles samal ajal kooskõlas Entringo süsteemi arhitektuuriga ja turvaline, seega on diplomitöös püstitatud ülesanne edukalt lahendatud ning lahendusele seatud nõuded autori arvamusel edukalt täidetud.

Kasutatud kirjandus

- [1] Hansab Group OÜ, „Entringo access control and parking system,“ 2019.
- [2] YAML Language Development Team, „YAML Ain’t Markup Language (YAML™) version 1.2,“ 1 Oktoober 2021. [Võrgumaterjal]. Available: <https://yaml.org/spec/1.2.2/>. [Kasutatud Mai 2022].
- [3] Amazon Web Services Korporatsioon, „Introduction to DevOps on AWS,“ Detsember 2014. [Võrgumaterjal]. Available: https://d0.awsstatic.com/whitepapers/AWS_DevOps.pdf. [Kasutatud Mai 2022].
- [4] OpenJS Foundation, „About | Node.js,“ [Võrgumaterjal]. Available: <https://nodejs.org/en/about/>. [Kasutatud Mai 2022].
- [5] npm, Inc, „npm About,“ [Võrgumaterjal]. Available: <https://www.npmjs.com/about>. [Kasutatud Mai 2022].
- [6] Hansab Group OÜ, „Environment Configuration Manual for Entringo,“ 2021.
- [7] M. DeHaan, „GitHub - ansible/ansible,“ [Võrgumaterjal]. Available: <https://github.com/ansible/ansible>. [Kasutatud Mai 2022].
- [8] Puppet, „GitHub - puppetlabs/bolt,“ [Võrgumaterjal]. Available: <https://github.com/puppetlabs/bolt>. [Kasutatud Mai 2022].
- [9] M. DeHaan, „The Origins of Ansible,“ 8 Detsember 2013. [Võrgumaterjal]. Available: <https://www.ansible.com/blog/2013/12/08/the-origins-of-ansible>. [Kasutatud Mai 2022].
- [10] Puppet Inc., „Welcome to Bolt,“ [Võrgumaterjal]. Available: <https://puppet.com/docs/bolt/latest/bolt.html>. [Kasutatud Mai 2022].
- [11] Puppet, „GitHub - puppetlabs/puppet,“ [Võrgumaterjal]. Available: <https://github.com/puppetlabs/puppet>. [Kasutatud Mai 2022].
- [12] Progress Software Corporation, „Chef Documentation,“ [Võrgumaterjal]. Available: <https://docs.chef.io/>. [Kasutatud Mai 2022].
- [13] Puppet Inc., „Puppet overview,“ [Võrgumaterjal]. Available: https://puppet.com/docs/puppet/7/puppet_overview.html. [Kasutatud Mai 2022].
- [14] UpGuard team, „What Is Configuration Management and Why Is It Important? | UpGuard,“ 11 Jaanuar 2022. [Võrgumaterjal]. Available: <https://www.upguard.com/blog/5-configuration-management-boss>. [Kasutatud Mai 2022].
- [15] Stack Overflow, „Stack Overflow Developer Survey 2018,“ [Võrgumaterjal]. Available: <https://insights.stackoverflow.com/survey/2018#development-practices>. [Kasutatud Mai 2022].
- [16] GitLab B. V., „GitLab - The One DevOps Platform,“ [Võrgumaterjal]. Available: <https://about.gitlab.com/>. [Kasutatud Mai 2022].
- [17] GitLab B. V., „CI/CD pipelines | GitLab,“ [Võrgumaterjal]. Available: <https://docs.gitlab.com/ee/ci/pipelines/index.html>. [Kasutatud Mai 2022].

- [18] N. D. T. L. P. S. B.-Y. Y. Daniel J. Bernstein, „High-speed high-security signatures,“ 26 September 2011. [Võrgumaterjal]. Available: <https://ed25519.cr.yp.to/ed25519-20110926.pdf>. [Kasutatud Mai 2022].
- [19] Free Software Foundation, Inc., „ssh-keygen(1) — Linux manual page,“ 11 August 2021. [Võrgumaterjal]. Available: <https://man7.org/linux/man-pages/man1/ssh-keygen.1.html>. [Kasutatud Mai 2022].
- [20] Free Software Foundation, Inc., „sshpass(1) - Linux man page,“ [Võrgumaterjal]. Available: <https://linux.die.net/man/1/sshpass>. [Kasutatud Mai 2022].
- [21] V. Gite, „How To Run / Execute Command Using SSH - nixCraft,“ 29 August 2021. [Võrgumaterjal]. Available: <https://www.cyberciti.biz/faq/unix-linux-execute-command-using-ssh/>. [Kasutatud Mai 2022].
- [22] M. Cooper, „Advanced Bash-Scripting Guide: Chapter 19. Here Documents,“ 5 Aprill 2014. [Võrgumaterjal]. Available: <https://tldp.org/LDP/abs/html/here-docs.html>. [Kasutatud Mai 2022].
- [23] M. Blake, „SCP Vs SFTP – Which is better for file transfer?,“ [Võrgumaterjal]. Available: <https://techprojournal.com/scp-vs-sftp/>. [Kasutatud Mai 2022].
- [24] TIOBE Software BV, „TIOBE Index for April 2022,“ Aprill 2022. [Võrgumaterjal]. Available: <https://www.tiobe.com/tiobe-index/>. [Kasutatud Mai 2022].
- [25] Google LLC, „V8 JavaScript Engine,“ [Võrgumaterjal]. Available: <https://v8.dev/>. [Kasutatud Mai 2022].
- [26] S. Boudrias, „GitHub - SBoudrias/Inquirer.js,“ [Võrgumaterjal]. Available: <https://github.com/SBoudrias/Inquirer.js>. [Kasutatud Mai 2022].
- [27] E. Klughammer, „GitHub - klughammer/node-randomstring,“ [Võrgumaterjal]. Available: <https://github.com/klughammer/node-randomstring>. [Kasutatud Mai 2022].
- [28] B. F. Nate Fischer, „GitHub - shelljs/shelljs,“ [Võrgumaterjal]. Available: <https://github.com/shelljs/shelljs>. [Kasutatud Mai 2022].
- [29] The OpenBSD Foundation, „openssl(1) - OpenBSD manual page server,“ [Võrgumaterjal]. Available: <https://man.openbsd.org/openssl.1>. [Kasutatud Mai 2022].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Harry Vipper

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Entringo parkimis- ja läbipääsulahenduse tarkvara halduse automatiseerimine“, mille juhendaja on Edmund Laugasson
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

15.05.2022

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Näide kasutatud Here document'ist üle SSH käskluste edastamiseks

```
#Ühendume üle SSH süsteemi tsentraalse haldusserveriga.  
ssh -i ~/.ssh/entringo-update.key -p 22 entringo@192.168.1.1 /bin/bash <<-SBO
```

```
#Väljastame tsentraalse haldusserveri versiooni numbri.  
echo -n 'Back Office '; /entringo/sbo/sbo -version
```

```
#Ühendume üle SSH lokaalse ala serveriga.  
ssh -i ~/.ssh/sbo.key -p 22 entringo@192.168.1.2 /bin/bash <<-LAS1
```

```
#Väljastame lokaalse ala serveri versiooni numbri.  
echo -n 'Area Server 1 '; /entringo/las/las -version
```

```
#Ühendume üle SSH riistvarakihti jooksutava seadmega.  
ssh -i ~/.ssh/las.key -p 22 entringo@192.168.1.3 /bin/bash <<-  
HAL1
```

```
#Väljastame riistvarakihi versiooni numbri.  
echo -n 'Access Gate 1 '; /entringo/hal/hal -version
```

```
HAL1
```

```
LAS1
```

```
SBO
```

Lisa 3 – Haldustööriista lähtekood

Diplomitöö käigus loodud lahenduse lähtekoodi piiratud versioon on laetud ülese GitHub'i varamusse: [harry-vipper/Entringo-Management-Tool](https://github.com/harry-vipper/Entringo-Management-Tool).

Haldus tööriista käivitamiseks on vajalik paigaldatud Node.js (v16.14.2). Pärast kõikide failide allalaadimist, tuleb nende asukohas jooksutada käsklus teekide paigaldamiseks: `'npm install'`. Alles siis saab käivitada haldustööriista käsuga: `'node ./management.js'`.