

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Informaatika instituut

IDK70LT

Tatjana Solovjova 142164IABM

**KULUEFEKTIIVSE
KONFIGURATSIOONIHALDUSE
ANDMEBAASIGA
AUTOMAATTUVASTUSSÜSTEEMI
LOOMINE ETTEVÕTTE ITIL
PROTSESSIDE TOETAMISEKS**

Magistritöö

Juhendaja: Jekaterina Tšukrejeva
Magistrikraad
Õppejõu assistent

Tallinn 2016

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: Tatjana Solovjova

09.05.2016

Annotatsioon

Antud töö eesmärgiks on kuluefektiivse automatiseeritud konfiguratsioonihalduse andmebaasi projekteerimine ja loomine. Eesmärgi saavutatakse läbi nõuete väljatöötamise, olemasolevate toodete analüüsi, nõuetele sobivate süsteemide valimise, ja nende täiustamise (programmeerimise).

Probleemiks on see, et suurtes ettevõtetes süsteemid arenevad väga kiiresti ja väga tihti ei jõuta õigeaegselt tehtud muudatuste kohta teha vastavat kirjet. Praegu olukord on selline, et ettevõtetes hoitakse infot konfiguratsioonielementide kohta isetehtud tabelites. Need tabelid on pärast mahuhalduse sisendiks. Kuna neid täidetakse käsitsi, tihti need ei vasta tegelikkusele. Hetkel turul leida ainult kommertstooteid, mis automaatselt koguvad andmed mõnede konfiguratsioonielementide kohta ja panevad need õigeaegselt andmebaasi, aga need on väga kallid ettevõtte jaoks.

Antud töö tulemusena analüüsi käigus sai valitud parim avatud lähtekoodiga konfiguratsioonihalduse andmebaasi lahendus, mida edaspidi automatiseeriti nii, et see automaatselt kogub info virtuaalmasinate ja selle elementide kohta.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 69 leheküljel, 5 peatükki, 10 joonist, 16 tabelit.

Abstract

Creating a cost efficient autodiscovery system with automatic CMDB integration to support ITIL workflows

The purpose of this thesis is planning and development of a cost efficient system with automatic configuration management database (CMDB) integration for a large Estonian company. This system is required to support IT Infrastructure Library (ITIL) workflows such as capacity, change, incident and problem management. The objective is achieved through development of system requirements and analysis of existing products which together result into selection of the system that meets all requirements but can also be further improved (programming).

The core problem at hand is that systems, which large corporate entities have, evolve so rapidly that personnel does not always manage to keep up-to-date track record of any changes. At the moment the setup that the corporation chosen as the subject of this thesis has is that whole information about systems and resources is stored in the tables, which are maintained by the personnel manually. The tables in turn are the input for capacity management, and since the tables are filled manually, information does not always correspond to reality which can result into incorrect decisions in the budgeting process. At the moment only commercial products can be found on the market; these products typically automatically collect information about configuration elements, but their cost is high and they are compatible only with other products offered by the same developer.

In this work by means of a comprehensive analysis process the best open source CMDB that meets the requirements established by the company has been chosen.

Two scripts were created as the backbone of this thesis; the scripts, after being launched, exported from server data about virtual machines and their components and then imported them to a CMDB that the author has chosen at the earlier stages of the work. One of the main advantages of the script that import data to CMDB is that it does not remove existing data upon execution, but checks the existence of the received data in the database and if the data has been modified then the script changes corresponding entries in the database or removes it if it no longer exists. The script for exporting was written on PERL, because

the chosen technology for virtualization has additional PERL SDK, through which a connection to API can be established. The script for importing was written on PHP, because the chosen CMDB has classes written on PHP which allow to read and add data to/from it.

The thesis is in Estonian and contains 69 pages of text, 5 chapters, 10 figures, 16 tables.

Lühendite ja mõistete sõnastik

ITIL

Information Technology Infrastructure Library

IT teenuste halduse parima praktika juhendite kogu. ITILi omanikuks on OGC ja ta koosneb seeriast raamatutest, mis annavad juhendeid, kuidas osutada kvaliteetseid IT teenuseid ning milliseid protsesse ja funktsioone on nende toetamiseks vaja. [1]

Infotehnoloogia taristu teek - briti päritoluga (avaldatud alates 1989) populaarne karkass ja meetodika IT-teenuste halduse protsessikeskseks organisatsiooni strateegiaga integreeritud halduseks; sisaldab häid tavasid, meespeasid, ülesandeid, protseduure jms; ta põhjal loodi briti standard BS15000 ja sellest tuletatud ISO/IEC 20000 (millega ta täielikult ei ühildu) [2]

Konfiguratsiooni- halduse andmebaas (KAB)

CMDB

Andmebaas, kus hoitakse konfiguratsiooni kirjeid läbi kogu nende elutsükli. Konfiguratsioonihalduse süsteem peab ülal ühte või mitut KABi ja iga KAB sisaldab KE-de atribuute ja KE-de vahelisi seoseid. [1]

Konfiguratsiooniel- ement (KE)

Configuration Item (CI)

Igasugune komponent, mida on IT teenuse osutamiseks vaja hallata. Iga KE kohta on info talletatud konfiguratsiooni kirjesse (mis paikneb konfiguratsiooni halduse süsteemis) ja KE-d haldab läbi kogu elutsükli konfiguratsioonihaldus, KE-d on muudatusehalduse kontrolli all. KE-deks on tavaliselt IT teenused, riistvara, tarkvara, ehitused, inimesed ja ametlik dokumentatsioon nagu protsesside dokumentatsioon ja TTLd. [1]

Teenustaseme lepe (TTL)

Service Level Agreement (SLA)

Lepe IT teenuseosutaja ja kliendi vahel. TTL kirjeldab IT teenuse, dokumenteerib teenustasemete sihid ja spetsifitseerib IT teenuseosutaja ja

kliendi vastutused. Üks TTL võib katta mitut IT teenust või mitut klienti.
[1]

Reaktsiooniaeg

Response Time

Ajavahemik, mis kulub mingi operatsiooni või transaktsiooni lõpetamiseks. Kasutatakse mahuhalduses IT infrastruktuuri sooritusvõime mõõtmiseks ja intsidendihalduses kõnede vastuvõtmise või kuni diagnoosi alustamiseks kulunud aja mõõtmiseks. [1]

Tarkvara püsikogu (DML)

Definitive Media Library (DML)

Üks või mitu asukohta kõigi kinnitatud tarkvara konfiguratsioonielementide versioonide turvaliseks hoidmiseks. DML võib sisaldada ka seotud KE-d nagu litsentsid ja dokumentatsioon. DML on loogiliselt üks kogu isegi siis, kui ta paikneb mitmes erinevas asukohas laiali. [1]

Kasutajaabi

Help Desk

Kontaktpunkt kasutajatele intsidentidest teada andmiseks. Kasutajaabi on tavaliselt kasutajatoest rohkem tehnilise kallakuga ja ei paku ühtainsat kontaktpunkti kõikvõimalikuks kommunikatsiooniks. Terminit kasutajaabi kasutatakse tihti ka kasutajatoe sünonüümina. [1]

Varahaldus

Asset Management

Varahaldus on protsess, mis vastutab finantsvarade väärtuse ja omanike jälgimise ning aruandluse eest läbi kogu varade elutsükli. Varahaldus on osa üldisest vara- ja konfiguratsioonihalduse protsessist. [1]

Kooste

Build

Tegevus, mis seisneb konfiguratsioonielementide kokkupanekus IT teenuse mingi osa loomiseks. Terminit kooste kasutatakse ka levitamiseks autoriseeritud reliisile viitamiseks. Näiteks serveri kooste või sülearvuti kooste. [1]

Ettevõtte ressurside

ERP (Enterprise Resource Planning)

Lai tegevuste ring, kus kasutatakse erinevaid rakendusprogramme ja mis aitab tootmis- või muul ettevõttel hallata olulisi ärikomponente nagu

planeerimine (ERP)	tooteplaneerimine, detailide sisseostmine, laoseisu haldamine, suhtlemine tarnijatega, kliendihaldus, tellimuste täitmise jälgimine jne. ERP võib sisaldada ka finants- ja inimressursside halduseks mõeldud tarkvara. Harilikult ERP süsteem kasutab või on ühendatud relatsioonandmebaasiga. [2]
Sõltumatute ketaste liiasmassiiv (RAID)	<i>RAID (Redundant Array of Independent Disks)</i> sõltumatute ketaste liiasmassiiv Varasema nimetusega "Redundant Array of Inexpensive Disks" ehk "odavate ketaste liiasmassiiv", rida ketassalvesti käideldavuse tõstmise meetodeid. [2]
Klaster	<i>Cluster</i> Personaalarvutite mälu tehnoloogias nimetatakse klasteriks failimälu loogilist ühikut kõvakettal. Klastreid haldab arvuti opsüsteem. [2]
Südamik	<i>Core</i> Põhimälu ehk muutmälu. See termin pärineb aegadest, kus põhimäludes kasutati mälu elementidena ferriitsüdamikke. Tänapäeval on sõna "südamik" muutunud arhaismiks peaaegu kõikjal peale IBM'i, kuid seda kasutab ka UNIX'i seltskond ja vanad häkkerid (või need, kes tahavad näida vanade häkkeritena). [2]
Muutmälu (RAM)	<i>RAM (Random-Access Memory)</i> Arvuti keskne mälu seade, kuhu saab andmeid kirjutada ja kust saab neid lugeda. Suvapöördus (random access) tähendab seda, et igal mälu pesal on oma aadress ning nii lugemiseks kui kirjutamiseks on võimalik pöörduda suvalise aadressi poole. Enamik muutmälusid pole säilmälud, s.t. toite väljalülitamisel mälu olevad andmed hävivad [2]
kesktöölusseade, keskprotsessor (CPU)	<i>CPU (Central Processing Unit)</i> on arvuti aju. Personaalarvutite puhul mahub see ära ühte kiipi ehk mikroskeemi ja seda nimetatakse mikroprotsessoriks. Iga protsessori kaks põhikomponenti on aritmeetika-loogikaplokk (ALU), mis teostab aritmeetilisi ja loogikatehteid, ning juhtplokk, mis võtab mälu kätkest ja täidab neid ise või vajaduse korral pöördub täitmiseks ALU poole [2]

**IP-protokolli
versioon 6
(IPv6)**

IPv6 (Internet Protocol version 6)

Tugevaim pretendent asendamaks juba alates 1981.a. kasutusel olevat IP-protokolli IPv4. IPv6 peamiseks eesmärgiks on lahendada IP-aadresside defitsiidi probleem ning sellel on 8-rühmalised 128-bitised aadressid ja tugevam andmeturve [2].

Mahuhaldus

Capacity Management

Protsess, mis vastutab selle eest, et IT teenuste ja IT infrastruktuuri mahud oleksid piisavad teenustaseme sihtide saavutamiseks tulusal ja õigeaegsel viisil. Mahuhaldus arvestab kõigi IT teenuste osutamiseks vajalike ressurssidega ja esitab lühi-, kesk- ja pikaajalised ärinõuete plaanid [1].

Sisukord

1 Sissejuhatus	14
1.1 Taust ja probleem	14
1.2 Ülesande püstitus	15
1.3 Metoodika.....	15
1.4 Ülevaade tööst	15
2 Olemasolevate lahenduste ja probleemide analüüs	16
2.1 Probleemi kirjeldus.....	16
2.2 Süsteemile (tootele) esitatavad nõuded	17
2.3 Turul olemasolevad süsteemid ja praktikad	19
2.4 Olemasolevate KAB-ide analüüs.....	21
2.5 KAB-ide analüüsi tulemused.....	27
2.6 Olemasoleva olukorra eelised ja puudused	28
2.7 Täpsem ülesande püstitus	29
3 Alternatiivid ja lahenduse valik.....	31
3.1 Olemasolevate lahenduste alternatiivid	31
3.2 Alternatiivide võrdlemine.....	33
3.3 Alternatiivide analüüs ja parima alternatiivi valik	42
4 Lahenduse realiseerimine ja hindamine	48
4.1 Realisatsiooni projekt	48
4.2 Realisatsiooni teostamine	48
4.2.1 Realisatsiooni arhitektuur	48
4.2.2 Testkeskonna loomine	51
4.2.3 Arendamine	52
4.3 Testimine ja tulemuste analüüs.....	59
4.3.1 Testimine	59
4.3.2 Tulemuste analüüs	61
4.4 Pakkumised edasiseks arendamiseks.....	63
5 Kokkuvõtte	64
Summary.....	66

Kasutatud kirjandus	68
Lisa 1 – Virtualiseerimistehnoloogiate võrdlemise kriteeriumid ja nende kirjeldused..	70
Lisa 2 – VMware vCenter Server Appliance 6.0 installimine.....	76
Lisa 3 – Testid Seleniumis	77

Jooniste loetelu

Joonis 1. Kasutusjuhtude diagramm funktsionaalsetele ja mittefunktsionaalsetele nõuetele.....	17
Joonis 2. Seosed ettevõtte äriprotsesside ja IT taristu vahel.....	21
Joonis 3. Kõige populaarsemad avatud lähtekoodiga KAB-ide tarkvarad ettevõtte jaoks sourceforge.net portaali kohaselt.....	22
Joonis 4. Virtualiseerimise olemus.....	32
Joonis 5. Näide PHP klassidest, mis on eelnevalt API's defineeritud.	49
Joonis 6. KAB-i ja vSphere andmevoog.	49
Joonis 7. Protsessi tegevusdiagramm.	50
Joonis 8. i-doit'is API sisselülitamine.....	52
Joonis 9. JSON'is VM-te info.....	55
Joonis 10. Funktsionaalsete testide läbiviimine Selenium tarkvaraga.	61

Tabelite loetelu

Tabel 1. Ettevõtte poolt lahendusele esitatud nõuded.	17
Tabel 2. KAB-ide tarkvara positsioon turul populaarsuse järgi [6].	22
Tabel 3. Üldine KAB-ide lahenduste kirjeldus.	24
Tabel 4. Lahenduste kasutajate arvud.....	25
Tabel 5. Nõuded süsteemile lahenduste paigaldamiseks.	25
Tabel 6. Avatud lähtekoodiga lahenduste võrdlemine.	26
Tabel 7. KAB-ide tarkvara analüüsi tulemuste koondtabel.....	27
Tabel 8. Kasutusjuhu FN02 ümberformuleerimine.....	30
Tabel 9. Virtualiseerimistehnoloogiate võrdlemine.	36
Tabel 10. Funktsionaalse nõue lõplik formuleering.	47
Tabel 11. Lahenduses käsitletud objektid ja nende kirjeldused.	59
Tabel 12. Vastuvõtutestid ja nende kirjeldused.....	59
Tabel 13. Funktsionaalse nõue test FNT01.	77
Tabel 14. Funktsionaalse nõue test FNT012.	77
Tabel 15. Funktsionaalse nõue test FNT013.	78
Tabel 16. Funktsionaalse nõue test FNT04.	78

1 Sissejuhatus

Kiiresti arenevas maailmas areneb veelgi kiiremini IT – valdkond. Eesti suuremad ettevõtted võtavad kasutusele ITIL'i metoodika. ITIL'i edaspidiseks kasutamiseks ettevõttes peab jälgima reegleid ja häid tavasid, täitma teatud ülesandeid. Käesolevas magistritöös kirjeldatakse ja rõhutatakse mahuhalduse ja muudatusehalduse protsessi, mis on tugevalt seotud teiste protsessidega, mida kirjeldab metoodika. Töö eesmärgiks on planeerida ja luua suure ettevõtte jaoks automaattuvastusega andmebaas, mis toetaks eelnevalt kirjeldatud protsesse tagamas andmebaasi pidevat aktuaalsust.

1.1 Taust ja probleem

Suurtes Eesti ettevõtetes hakatakse rakendama või rakendatakse ITIL-i (edaspidi räägitakse ITIL versioonist 3). Paljude protsesside (nt. mahuhalduse, muudatusehalduse, probleemihalduse, intsidendihalduse) aluseks on korraliku konfiguratsioonihalduseandmebaasi (edaspidi KAB-i) olemasolu. Praegu aga peaaegu kõik KAB-id, mida ettevõtted kasutavad on karptooded, mida täidavad töötajad ise, tihti kasutatakse selle asemel Excel'i tabelit, mida täidetakse samuti käsitsi. Kuna praegusel ajal kõik muutub väga kiiresti, tulevad välja uued seadmed, rakendused, tarkvara jm. kõik need lahendused ei ole elujõulised. Probleem seisneb selles, et andmed nii kiiresti muutuvad, et töötajad (andmete sisestajad / andmebaaside administraatorid) ei jõua õigeaegselt neid sisestada (isegi keskmise suurusega ettevõttes).

Seda tööd on vaja, et toetada muudatusehalduse protsessi ettevõttes, mis rakendab ITIL'i. Nimelt teha nii, et täpne info riist-, tarkvara ja muude konfiguratsioonielementide kohta oleks automaatselt kantud andmebaasi, ehk täiesti inimressursi vältimine andmete kogumisel ja sisestamisel.

Antud tööst peaksid olema huvitatud paljud suured ettevõtted, kes soovivad tagada aktuaalse KAB-i olemasolu. Antud töö on tehtud ühe suure Eesti ettevõtte jaoks.

Magistritöö käigus kavandatakse ja realiseeritakse eespool kirjeldatud lahendus ja siis testitakse lokaalselt. Pärast testimist võetakse lahendus kasutusele, aga see samm ei ole magistritöö osa.

1.2 Ülesande püstitus

Antud töö eesmärgiks on kuluefektiivse automatiseeritud konfiguratsioonihalduse andmebaasi projekteerimine ja loomine läbi nõuete väljatöötamise, olemasolevate toodete analüüsi, nõuetele sobivate süsteemide valimise, ja nende täiustamise (programmeerimise).

1.3 Metoodika

Eesmärkide saavutamiseks antud töö autor alustab süsteemile esitatavate nõuete väljaselgitamisest, pärast tegeleb turul olemasolevate toodete analüüsiga, ehk vaatab, missugused need on ja vaatab nende eeliseid ja puudusi. Edaspidi saadud tulemuste põhjal valib süsteemi, mille alusel hakkab lahendust looma. Järgmiseks sammuks koostab täpsemaid nõudeid süsteemile ja realiseerimisplaani. Edaspidi toimub plaani ja nõuete järgi realisatsiooni projekteerimine ja loomine. Järgmise sammuna testitakse tehtud süsteemi testserveri ja – andmebaasiga. Kui kõik saab tehtud, toob autor välja ettepanekud süsteemi edaspidiseks arendamiseks.

1.4 Ülevaade tööst

Esmalt analüüsib antud töö autor turul olemasolevaid lahendusi ja valib parima edaspidiseks arendamiseks. Järgmise sammuna formuleerib täpsemalt ülesandepüstituse.

Vastavalt ülesandepüstitusele valib tehnoloogia, mille põhjal hakkab edaspidi lahendust looma.

Edaspidi pakub välja autor realiseerimisplaani ja hakkab selle järgi töötama välja lahenduse. Esmalt töötab välja autor arhitektuuri lahenduse jaoks ja selle pärast hakkab analüüsi käigus valitud toodet automatiseerima.

Töö lõpus teeb töö kohta järeldusi ja pakub võimalusi edasiseks arendamiseks.

2 Olemasolevate lahenduste ja probleemide analüüs

Järgnevas peatükis kirjeldatakse täpsemalt probleemi ja selle olemasolevaid lahendusi.

2.1 Probleemi kirjeldus

Nagu oli enne (vt jaotis 1.1) kirjeldatud, suuremates Eesti ettevõtetes hakatakse rakendama ITIL'i IT-teenuste halduseks. Antud metoodikas [2] kirjeldatud protsesside jaoks on väga oluline täpse (tegelikkusele vastava) IT taristu kohta andmebaasi olemasolu. Praegune olukord on selline, et mõned ettevõtted kasutavad juba valmis konfiguratsioonihalduse andmebaase, mis ei ole automatiseeritud, ehk neid peab käsitsi täitma. See aga tähendab, et andmeid ei uuendata pidevalt ja sellised andmebaasid ei vasta juba reaalsusele, ehk need sisaldavad vananenud infot KE-de kohta.

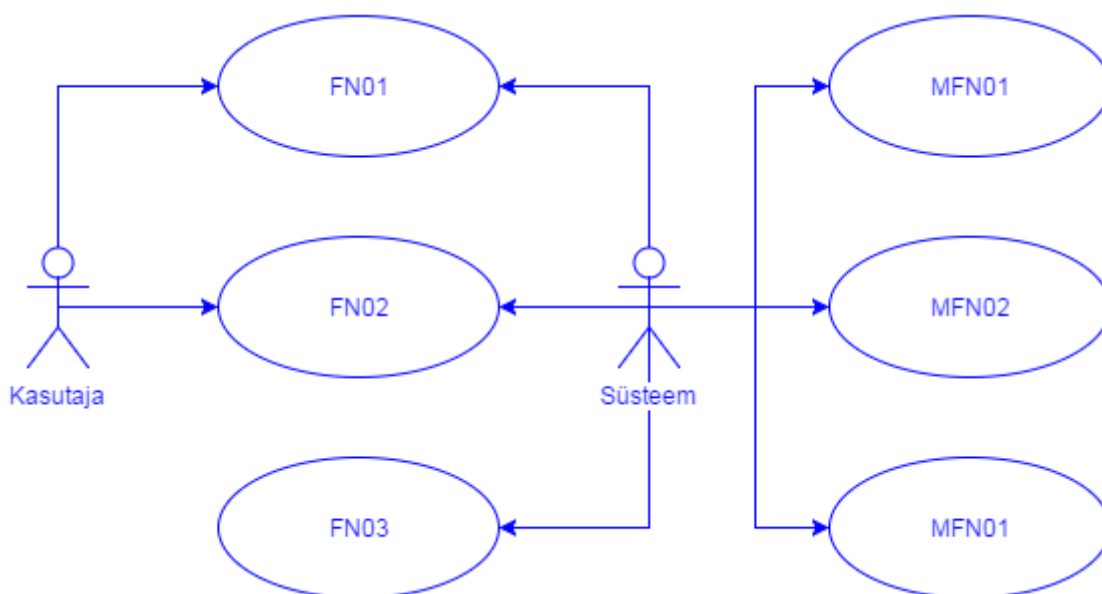
Probleemi põhjusi on mitu ja üks nendest on inimfaktor. Tihti kommunikatsioon ettevõtte osakondade vahel ei ole täiuslik, ehk vahel töötajad unustavad edastada infot uuenduste ja muudatuste kohta (nt. riistvara vahetus, tarkvara uuendus jm). Samas on ka palju rahvusvahelisi ettevõtteid, ja siis info vahetuse kvaliteet väheneb, kuna vahemaa suureneb. Ei tohi unustada ka seda, et inimesed vahel jäävad haigeks, puuduvad tööst mõjuvatel põhjustel jm. ITIL'i rakendavad ettevõtted on tavalised suured ja tõenäoliselt seadmete vahetus, uute virtuaalmasinate loomine, rakenduste ja tarkvara uuendus toimub pidevalt, ja mitte alati on inimene võimeline seda kõike jälgima ja õigeaegselt andmeid andmebaasi sisestama.

Hetkel turul leida ainult kommertstooteid (näiteks UCMDB – Hewlett Packard toode), mis automaatselt koguvad andmed ja panevad neid õigeaegselt andmebaasi. Kuna vananenud andmete probleem jääb alles, hakkavad ettevõtted ise arendama automaattuvastussüsteemi. See on aga aja- ja rahakulukas protsess. [3]

Ettevõtte X on suur eesti ettevõtte ja tänapäevani on kasutanud Exceli tabelit, et hoida infot IT-taristu elementide kohta. Kuna ettevõtte on suur ja süsteemid muutuvad pidevalt, ei ole võimalik tagada aktuaalset infot olemasolevate seadmete ja süsteemide kohta, mis omakorda takistab mahu- ja muudatusehalduse protsessi. Selleks, et koguda infot taristu elementide kohta, kulub selleks aega kuni nädala aega. Selleks ajaks juba kogutud info muutub mitteaktuaalseks, kuna IT-taristu elemendid pidevalt muutuvad ja näiteks serveri kõvaketta suurus ei pruugi olla samasugune nädala lõpus.

2.2 Süsteemile (tootele) esitatavad nõuded

Selleks, et ettevõttes oleks alati aktuaalne info taristu kohta, peab töötama välja lahenduse, mis automaatselt koguks infot nende kohta ühte andmebaasi. Järgnevalt on loetletud Tabel 1 ettevõtte X poolt esitatud funktsionaalsed ja mittefunktsionaalsed nõuded arendatavale lahendusele. Nõuded esitatakse kasutusjuhtudena (vt Joonis 1) ning iga kasutusjuhu kohta antakse järgmised andmed: ID, kasutusjuhu nimi (eesmärk), tegutseja(d), eeltingimused, järeltingimused ja põhistsenaarium.



Joonis 1. Kasutusjuhtude diagramm funktsionaalsetele ja mittefunktsionaalsetele nõuetele.

Tabel 1. Ettevõtte poolt lahendusele esitatud nõuded.

Kasutusjuhu ID:	FN01
Kasutusjuhu nimetus (Eesmärk):	Sisselogimine (Sisselogimise eesmärgiks on tagada andmete nähtavuse ainult teatud ettevõtte töötajate rühmadele)
Tegutsejad:	Kasutaja, süsteem
Eeltingimused:	Kasutajal on olemas kasutajanimi ja parool sisselogimiseks.
Järeltingimused:	Sisselogimine on õnnestunud siis, kui ekraanile kuvatakse info taristus olevate objektide kohta. Sisselogimine on mitteõnnestunud, kui ekraanile kuvatakse veateate.
Põhistsenaarium	
Kasutaja	Süsteem
1. Kasutaja sisestab oma kasutajanime ja parooli	2.a Kui sisestatud andmed on õiged, kuvatakse kasutajale andmed objektide kohta 2.b Kui sisestatud andmed on valed, kuvatakse ekraanile sellekohane teade.

Kasutusjuhu ID:	FN02
Kasutusjuhu nimetus (Eesmärk):	Andmete lisamine süsteemi (tagada IT üksuste töötajate teadlikkuse ettevõttes kasutatavate ressursside kohta)
Tegutsejad:	Kasutaja, süsteem
Eeltingimused:	Olemas andmebaas, kuhu lisatakse andmed
Järeltingimused:	Andmebaasi on lisatud andmed IT taristus olevate objektide kohta
Põhistsenaarium	
Kasutaja	Süsteem
1. Kasutaja käivitab andmete lisamise protsessi	2. Süsteem sisestab automaatselt andmed IT taristu objektide kohta andmebaasi
Kasutusjuhu ID:	FN03
Kasutusjuhu nimetus (Eesmärk):	Andmete kontrollimine (tagada alati tegelikkusele vastava info andmebaasis olemasolu)
Tegutsejad:	Süsteem
Eeltingimused:	Andmebaasis on juba andmed olemas objektide kohta
Järeltingimused:	Kui andmed erinevad olemasolevatest, siis vanad kutsutakse ja nende asemel sisestatakse uued
Põhistsenaarium	
Süsteem	
Ekspordib andmed IT taristu objektide kohta	
Võrdleb saadud andmed andmebaasis olevate andmetega	
3.a Kui andmed on sarnased, siis muudatusi ei tehta	
3.b Kui andmed on erinevad, siis kustutakse andmebaasis olevad andmed ja sisestatakse eksporditud andmed nende asemele	
Kasutusjuhu ID:	MFN01
Kasutusjuhu nimetus (Eesmärk):	Avatud lähtekoodiga lahenduste kasutamine (kuna eesmärgiks on kuluefektiivse lahenduse loomine, peab lahendusel olema võimalikult väike maksumus, seega kui kasutatakse lahenduse väljatöötamisel teisi süsteeme, peab kasutama avatud lähtekoodiga lahendusi)
Tegutsejad:	Süsteem
Eeltingimused:	Otsustatakse lahenduses kasutada teisi süsteeme
Järeltingimused:	Lahenduses kasutatakse avatud lähtekoodiga süsteemi
Põhistsenaarium:	Kui käesoleva töö autor otsustab kasutada oma lahenduses teisi süsteeme, need peavad olema avatud lähtekoodiga
Kasutusjuhu ID:	MFN02

Kasutusjuhu nimetus (Eesmärk):	Minimaalsed nõuded riistvarale (tagada tulevase süsteemi mobiilsuse)
Tegutsejad:	Süsteem
Eeltingimused:	Süsteemi arhitektuur sai väljatöötatud
Järeltingimused:	Süsteem on seadistatud
Põhistsenaarium:	Seadistamise ajal väljatöötatud lahendus ei kasuta rohkem ressursse, kui 8GB RAM'i ja 500GB kõvaketta mahtu
Kasutusjuhu ID:	MFN03
Kasutusjuhu nimetus (Eesmärk):	Inglisekeelse liidese olemasolu (tagada lahenduse vastavuse ettevõtte standartidele)
Tegutsejad:	Süsteem
Eeltingimused:	Süsteemi arhitektuur sai väljatöötatud
Järeltingimused:	Süsteemi keeleks on inglise keel
Põhistsenaarium:	Kui kasutaja logib süsteemi sisse, kõik andmed kuvatakse inglise keeles

2.3 Turul olemasolevad süsteemid ja praktikad

Tuleb veel kord mainida, et tooted, mida enamasti kasutavad täna ettevõtted oma IT halduseks on KAB-id. KAB on andmebaas, kus hoitakse konfiguratsiooni kirjeid läbi kogu nende elutsükli. Konfiguratsioonihalduse süsteem peab ülal ühte või mitut KAB-i ja iga KAB sisaldab konfiguratsioonielementide (edaspidi KE) atribuute ja KE-de vahelisi seoseid [1].

Integreeritud KAB-id võimaldavad IT - teenuste tugirühmadel teha tõhusat koostööd (juhul, kui andmebaasides olev info on aktuaalne) ja muuta lihtsamaks mahu-, muudatuse-, intsidenti-, konfiguratsiooni- ja probleemihalduse protsesside juhtimist. Ettevõttes KAB-ide kasutamine tõstab IT-osakondade efektiivsust ja maandab riske [4]:

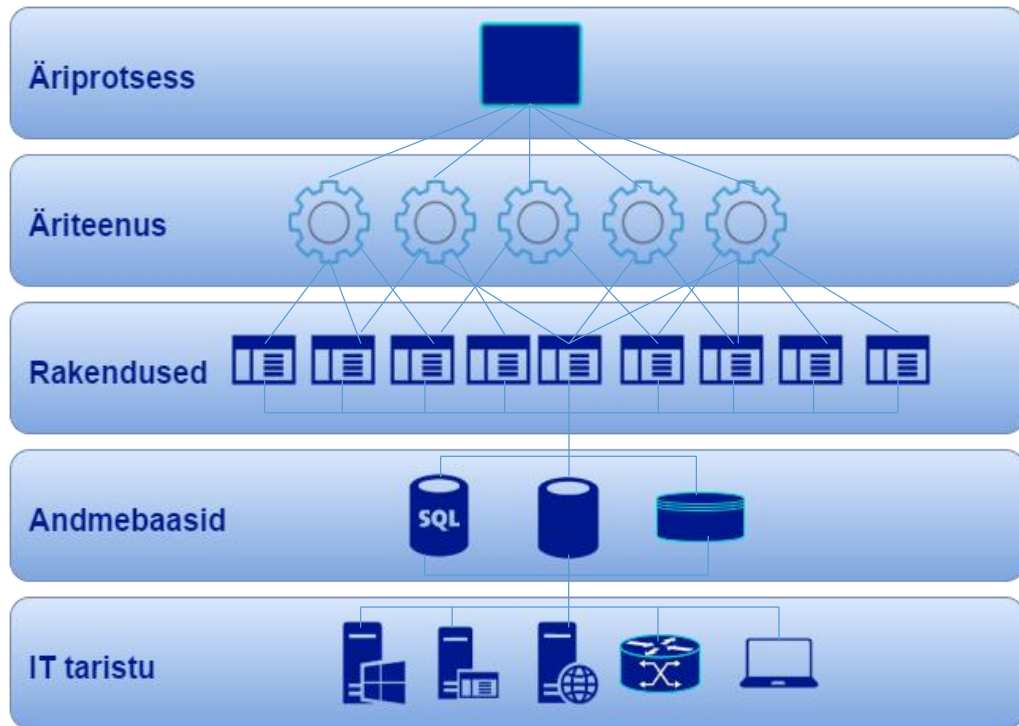
Intsidenti- ja probleemihaldus. Kuna KAB võimaldab kaardistada KE-de suhteid, ettevõttes intsidenti- ja probleemihaldusega tegelev personal saab paremini mõista juhtumite algpõhjused, seega kiirendades probleemide diagnoosimist ja lahendamist. Samuti võimaldab see saada aru probleemi mõju suurust äriks, mis omakorda aitab kaasa seadistada prioriteete, lähtudes ettevõtte ärieesmärkidest. Samuti KAB-is olevate KE-dest ja nende suhetest arusaamine võimaldab teha efektiivsemaks kasutajatoe tööd läbi järgmiste mõõdikute: intsidentide lahendamisaeg, intsidenti ja probleemi suhe ja reaktsiooniaeg.

Muudatusehaldus. Muudatusehalduse eest vastutavad isikud saavad paremini planeerida muudatusi, määrates mõju kavandavatest IT-taristu muutustest teenustele, vältides probleeme ja konflikte. Inimene, kes valdab seda informatsiooni, võib jälgida kõiki muutusi edaspidise auditi jaoks.

Konfiguratsiooni- ja reliisihaldus. Personal, kes vastutab konfiguratsiooni- ja reliisihalduse eest võib kasutada KAB-is olevat infot seadmete konfiguratsioonide kohta, selleks, et võimaldada neil edaspidi planeeritud uusi reliise vastu võtta.

Mahuhaldus. Kui KAB-is aktuaalse info olemasolul saab efektiivsemal planeerida ressursse.

Nagu eelnevalt oli mainitud, parimad KAB-id sisaldavad ka seoseid. Seoste kajastamine aitab näha ja aru saada kogu pildist, alates äriprotsessist ja liikudes alla toetava taristu juurde (vt Joonis 2, mis on tehtud kasutades draw.io tarkvara [5]). Nagu Joonis 2 on näha, on kokku viis kihti, mis on omavahel seotud. Kõige alumiseks kihiks on IT taristu, mis toetab ettevõtte äriprotsesse. Riistvaral luuakse andmebaase, mida omakorda kasutavad rakendused jne. Kui KAB-is on aktuaalne info rakenduste, andmebaaside jm KE-de kohta, siis kui ilmneb mingi probleem, võib lihtsalt jälgida, mis on probleemi aluseks (kuna KAB-is on defineeritud ka KE-de seosed). Näiteks, kui mingi äriprotsess on toetatud rakendusega, mis omakorda aeglaselt töötab, KAB-ist võib näha, millist andmebaasi antud rakendus kasutab ja siis võib probleem peituda andmebaasis või kui vaadata andmebaasi seoseid, võib, nt. tuvastada, et andmebaas asub kas vananenud serveris või lihtsalt andmebaasi serveril on vähe ressursse (nt. RAM-i). KAB-id on suurte ettevõtete jaoks väga kasulikud, kuna aitavad kiiremini probleeme tuvastada, kasutades KE-de omavahelisi seoseid.



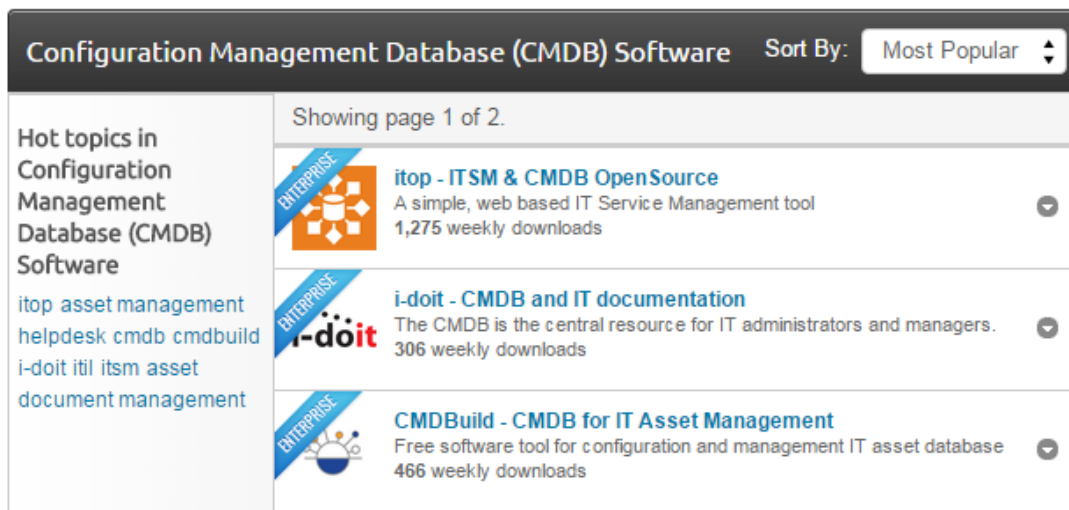
Joonis 2. Seosed ettevõtte äriprotsesside ja IT taristu vahel.

2.4 Olemasolevate KAB-ide analüüs

Arvestades seda, et antud töö eesmärgiks on kuluefektiivse süsteemi väljatöötamine, ja üheks võimaluseks on avatud lähtekoodiga süsteemide kasutamine (vt jaotis 2.2), otsustati, et lahendust hakatakse looma juba mõne olemasoleva avatud lähtekoodiga lahenduse põhjal, kuna nullist süsteemi arendamine on väga ajakulukas protsess. Seega edaspidi vaadeldakse parimaid avatud lähtekoodiga KAB-ide lahendusi, kuna neid saab hiljem täiustada. Antud peatükis analüüsitavate lahenduste eeliseks, nagu juba oli öeldud, on avatud lähtekood, kuna see aitab ettevõttel raha kokku hoida. Veel üheks eeliseks on tarnijalt sõltumatus. Kui otsustakse minna üle teisele lahendusele, võib lihtsalt eksportida andmeid KAB-ist. Ettevõttel ei tule probleeme, kui tarnija läheb ärist välja (lõpetab oma toote arendamise) või ootamatult pankrotistub.

Analüüsitavate KAB-ide arvu vähendamiseks valis antud töö autor kolm kõige populaarsemat avatud lähtekoodiga toodet ettevõtte jaoks, vt Joonis 3 (veebilehelt <https://sourceforge.net/>, kuhu pannakse välja kõik avatud lähtekoodiga lahendused) [6]. Joonisel (vt Joonis 3) on

kujutatud avatud lähtekoodiga ettevõttele mõeldud KAB-ide avatud lähtekoodiga lahendused, mis on järjestatud populaarsuse järgi. Populaarsust arvutatakse lähtudes sellest, kui palju kordi nädalas sai toode allalaetud.



Joonis 3. Kõige populaarsemad avatud lähtekoodiga KAB-ide tarkvarad ettevõtte jaoks sourceforge.net portaali kohaselt.

Järgnevalt on Tabel 2 näha ka lahenduste keskmise hinde, mis oli määratud toote kasutajate poolt.

Tabel 2. KAB-ide tarkvara positsioon turul populaarsuse järgi [6].

Jrk.	Tarkvara	Allalaadimiste arv nädalas	Keskmine hinne 5.0 skaalal
1	iTop	1275	4.8
2	CMDBuild	466	4.0
3	i – doit	306	5.0

Kõik tabelites Tabel 2 ja Tabel 5 esitatud andmed on võetud portaalist, kuhu laetakse avatud lähtekoodiga tarkvara. Tabel 2 on näha, et kõige populaarsem toode allalaadimiste järgi on iTop, i-doit tarkvara aga laetakse 4 korda vähem, kui esimest. Samas aga keskmine hinne on kõige kõrgem viimasel lahendusel (keskmine hinne arvutakse kasutajate arvamuste järgi).

Tabel 3 annab põhjaliku ülevaate eelnevalt loetletud tööriistade kohta, kus

- **Lühidalt** – lühikirjeldus iga tehnoloogia kohta
- **Esmane eesmärk** – igat tehnoloogiat võib kasutada erinevatel viisidel, kuid see on peamine
- **Tugevused** – peamised toote positiivsed küljed, mida rõhutati läbivaatamise käigus
- **Nõrkused** – peamised toote negatiivsed küljed, mida rõhutati läbivaatluse käigus

Tabel 3 võib teha järelduse, et iTop tarkvara on suunatud pigem kogu IT töö keskendumiseks ühes kohas, CMDBuild ja i – doit tarkvarad on keskendunud pigem korraliku KAB-i olemasolule, kuna toodete funktsionaalsus on suunatud sellele, et oleks kirjeldatud võimalikult palju erinevat tüüpi KE-e.

iTop'i üheks suureks puuduseks on see, et veebis olev dokumentatsioon on vananenud, seda ei uuendata pidevalt vaatamata sellele, et tarkvara funktsionaalsus pidevalt muutub. Samuti väga suureks puuduseks on ka see, et foorumid ehk kasutajatugi ei toimi. Tarkvara ametlikest foorumitest võib leida palju vastamata küsimusi, mis ootavad lahendusi juba mitu aastat.

CMDBuild'i suureks eeliseks on see, et peale suhete kaardistamist võimaldab see koostada graafikuid ja diagramme olemasoleva info alusel. Suureks puuduseks on aga see, et dokumentatsioonist võib leida blokke itaalia keeles, kus peaks olema info inglise keeles. Samuti üheks ebamugavuseks on ka see, et tarkvara katsetamiseks (demo) peab taotlema isikliku kasutajanime ja parooli, ehk koheselt katsetada süsteemi ei saa.

i – doit'i suureks eeliseks on see, et see tarkvara on keskendunud täpselt korraliku KAB-i olemasolule, kuna see võimaldab lisada palju erinevaid KE (tüüpide järgi). See sisaldab umbes kaks korda rohkem erinevaid KE-de tüüpe, kui iTop tarkvara. Puudusi peaaegu ei ole, kuna tarkvaral on kitsas realiseerimisvaldkond ning tarkvara on väga hästi läbimõeldud, kuna andmebaasi võib lisada kogu vajaliku info IT taristu kohta. Üheks puuduseks on see, et tasuta versioon ei toeta CSV importi, samas aga toetab XML importi.

Tabel 3. Üldine KAB-ide lahenduste kirjeldus.

Tarkvara	Lühidalt	Esmane eesmärk	Tugevused	Nõrkused
iTop	iTop – avatud lähtekoodiga veebipõhine produkt, mis on loodud IT ettevõtte protsesside automatiseerimiseks. Sisaldab KAB-i, kasutajaabi ja dokumendihalduse tööriista. Samuti sisaldab teisi tööriistu ja veebiteenuseid IT integreerimiseks. [7]	Lihtsustada kasutajate pöördumise protsessi, intsidendi-, probleemi-, muudatuse- ja teenuste kataloogi haldust ühes kohas. [8]	<ul style="list-style-type: none"> + Onlain [2] demo olemasolu + Toetab mitu ITIL protsessi + Interaktiivne CSV import + Lihtne seadistamine + Väikesed nõuded süsteemile 	<ul style="list-style-type: none"> - Osaliselt vananenud dokumentatsioon - Kasutajatoe kehv töö (aastate vanad küsimused jäävad vastamata) - Vähe KE-de tüüpe
CMDBuild	CMDBuild – ERP süsteem IT osakonna jaoks. See on loodud KAB-i haldamiseks ja äri toetamiseks, vastavalt ITIL parimatele praktikatele. [9]	Lihtsustada IT üksuste tööd ja tagada kontrolli varade üle, teades kogu aeg nende koosseisu, dislokatsiooni, funktsionaalseid suhteid ja kuidas pidevalt neid uuendada, samuti elutsükli terviklikult hinnata.	<ul style="list-style-type: none"> + Palju erinevaid liidesekeeli + Interaktiivne CSV import + Diagrammide ja graafikute tegemise võimalus + Võimaldab lisada süsteemi koostööpartnereid (tarnijaid) 	<ul style="list-style-type: none"> - Osa dokumentatsioonist on tõlkimata - Ei ole koheselt kättesaadavat demot, peab taotlema - Palju kaebusi seadistamise kohta
i – doIt	i - doIt – avatud lähtekoodiga KAB ja IT dokumentatsioon. [10]	Tagada korraliku KAB-i ettevõttele	<ul style="list-style-type: none"> + Minimaalsed nõuded süsteemile + Onlain demo olemasolu + Lihtne seadistamine + Toetab palju operatsioonisüsteeme + Sisaldab palju erinevaid KE-de tüüpe + XML import + Korralik sisukas dokumentatsioon + Kaablite lisamise võimalus 	<ul style="list-style-type: none"> - Ei toeta CSV importi

Tabel 4. Lahenduste kasutajate arvud.

Tarkvara	Kasutajad
iTop	klientide arv on avalikustamata, aga väidetakse, et klientide arv pidevalt suureneb. Nt, üheks suureks kliendiks on HP Intel Solution Center
CMDBuild	viimase 10 aasta jooksul allalaadimiste arv oli rohkem, kui 110 000, kus 90 000 on alates aasta 2011
i – doit	ei ole andmeid selle kohta

Nagu Tabel 4 on näha, tarnijad ei avalikustata infot klientide arvu kohta. Samas aga tabelist on näha, et iTop'il on olemas kuulsaid kliente, sealjuures kui aga vaadata Tabel 2, siis võib näha, et nädalas laetakse alla rakendus tihti. CMDBuild'i kohta on olemas info, et viimaste aastate jooksul klientide arv järsult suurenes, mille põhjuseks võib ole IT valdkonna kiire ja pidev areng.

Tabel 5. Nõuded süsteemile lahenduste paigaldamiseks.

Tarkvara	Nõuded	
	Tarkvarale	Riistvarale
iTop	<ul style="list-style-type: none"> • Apache Webserver / IIS / muu veebiserver PHP 5 jaoks • MySQL • Alfresco 3.4 • JDK 1.8 Oracle 	<ul style="list-style-type: none"> • Kaheüdamikuline protsessor • 1 GB muutmälu • 5 GB mahtu kõvakettal
CMDBuild	Apache Tomcat 7.0.32 või uuem PostgreSQL 9.0 või uuem PHP 5.4.x, 5.5.x	<ul style="list-style-type: none"> • Kaheüdamikuline protsessor, viimased generatsioonid • 4GB muutmälu • 100 GB mahtu kõvakettal
i – doit	<ul style="list-style-type: none"> • Apache Webserver 2.2, 2.4 • MySQL 5.1, 5.5 või MariaDB 5.5 • PHP 5.4.x, 5.5.x 	<ul style="list-style-type: none"> • Kaheüdamikuline protsessor • 1 GB muutmälu • 1 GB mahtu kõvakettal

Tabel 5 on näha, et kõige minimaalsemad nõuded süsteemile on i – doit tarkvaralahendusel. Teisel kohal on siis iTop ja kõige kõrgemad nõuded on CMDBuild tarkvaral. Samuti Tabel 3 võib näha, et üheks selle lahenduse puuduseks on see, et kasutajad kaebavad tarkvara paigaldamise probleemidele, mis võib olla tingitud sellest, et tarkvaral on kõrgemad nõuded riistvarale. Samas peab mõistma, et Tabel 5 esitatud

nõuded on minimaalsed nõuded süsteemile, ehk kui plaanis on suure andmebaasi olemasolu, siis peavad tabelis toodud näitajad olema paremad, nt. kui suurendada muutmälu suurust, siis muutub ka andmebaasist lugemise kiirus.

Allolev tabel (vt Tabel 6) sisaldab mõned spetsiifilisemad eristajad tarnijate vahel. Need omadused on valitud eesmärgiga konkreetselt näidata iga tarnija erinevaid lähenemisi, rakendusvaldkondi ja meetoodikat. Tabel 6 on kasutatud järgmised võrdlemispunktid:

1. Green IT – mõned tööriistad annavad teavet selle kohta, et toetavad projekte, et vähendada IT energiatarvet. Kuna see valdkond muutub aina populaarsemaks [11], otsustati lisada selle kriteeriumite hulka.
2. Füüsiline asukoht – võime dünaamiliselt jälgida seadme füüsilist asukohta, vastavalt sellele, mis kommutaatorisse see on ühendatud.
3. Kas on võimalik seiretarkvara kasutamine? Passiivne (aruandlus) või aktiivne (kasutamise kontroll).
4. Tarkvara kajastamise andmebaas – info kogumine tarkvara versioonide kohta.
5. Suhete kaardistamine – võime kaardistada suhteid KE-de vahel.
6. Metaandmed – võimalus lisada täiendavat infot varade kohta. See on eriti oluline, kui ei kasutata aktiivselt varahaldust. Nt, võimalus hoida infot ostmise kohta või viiteid teistele süsteemidele.
7. Hoiatamine (alarmid) – mõned tööriistad pakuvad aja analüüsimise punkti. (Kas kõik näeb välja nii, nagu esmaspäeval? Kas kõik näeb välja nii, nagu kolmapäeval? Võrdleme.)
8. Migratsiooni toetus – mõned tööriistad pakuvad stsenaariumite modelleerimist, mis ennustab ühilduvust tuleviku platvormide või koostega. Nt. näita mulle varad, mis ühilduvad Windows 10-ga.

Tabel 6. Avatud lähtekoodiga lahenduste võrdlemine.

		iTop	CMDBuild	i – doit
1	Äriteave võimsuste, nominaalsete võimsuste ja teise näitajate kohta, mis puutuvad Green IT projekte.	✗	✗	✗
2	Automaatne füüsilise asukoha jälgimine	✗	✗	✗
3	Seiretarkvara kasutamine	✓	✗	✓
4	Sisseehitatud tarkvara tuvastamise moodul	✗	✗	✗

		iTop	CMDBuild	i – doit
5	Suhete kaardistamine	✗	✓	✓
6	Täiendav info (ostu kuupäev, seadme number, garantii, üksus jne)	✓	✓	✓
7	Hoiatamine	✗	✗	✓
8	Tööriistad migratsiooni toetamiseks	✗	✗	✗
KOKKU ✓ :		2	2	4
✓ - olemas / toetab / korras/ aktiivne ✗ - ei ole / ei toeta / ei ole korras / passiivne				

Tabel 6 on näha, et tarkvara tarnijad ei toeta Green IT projekte. Lisaks sellele ükski tarkvarast ei paku automaatse asukoha leidmise võimalust, aga kõikides KAB-ides on olemas võimalus lisada asukoht käsitsi. Samuti KAB-id võimaldavad registreerida lisaks ettevõtte asukohale hoonesiseseid ruume. iTop ja i – doit pakuvad samuti seiretarkvarakasutamise võimalust, CMDBuild pakub aga ainult aruandluse võimalust. CMDBuild ja i – doit pakuvad suhete kaardistamise võimalust, CMDBuild'il on aga parem lahendus, kuna pakub visuaalset lahendust, i – doit'is aga info esitatakse tabelites. Iga tarkvaralahendus pakub salvestada andmebaasi info ka ostu kuupäeva, seadme numbriga, garantii jm kohta. Kuna i – doit'i on sisseehitatud seiretarkvara moodul, see võimaldab võrrelda seisundid erinevate ajahetkedel. Ükski tarkvarast ei toeta migratsiooni jaoks ühilduvuse kontrolli.

2.5 KAB-ide analüüsi tulemused

Peatükis 2.3 analüüsiti kolm populaarsemat avatud lähtekoodiga KAB-i tarkvaralahendust: iTop, CMDBuild ja i – doit. Kõik andmed olid võetud tarkvarade veebilehtedelt ja foorumitelt. Info funktsionaalsuse kohta saadi läbi demode testimise. Järgnevalt on esitatud koondtabel (vt Tabel 7) saadud tulemustega. Näitajaid, mis puudusid kõikidest lahendusest või mõne lahenduse kohta olid puudulikud, ei kajastu järgnevas tabelis.

Tabel 7. KAB-ide tarkvara analüüsi tulemuste koondtabel.

		iTop	CMDBuild	i – doit
1	Suur allalaadimiste arv nädalas	✓	✗	✗

		iTop	CMDBuild	i – doit
2	Kõrge keskmine hinne tagasiside põhjal	✓	✗	✓
3	Onlain demo	✓	✓	✓
4	Korralik dokumentatsioon	✗	✗	✓
5	Palju KE-de tüüpe	✗	✓	✓
6	Palju liidesekeeli	✓	✓	✗
7	Seadistamise lihtsus	✓	✗	✓
8	Minimaalsed nõuded tark- ja riistvarale	✓	✗	✓
9	Aktiivne seiretarkvara kasutamine	✓	✗	✓
10	Suhete kaardistamine	✗	✓	✓
11	Täiendav info (ostu kuupäev, seadme number, garanti, üksus jne)	✓	✓	✓
12	Hoiatamine	✗	✗	✓
KOKKU ✓ :		8	5	10
✓ - olemas / toetab / korras ✗ - ei ole / ei toeta / ei ole korras				

Nagu Tabel 7 on näha, on parimad lahendused on iTop ja i – doit, mis vastab keskmistele hinnetele Tabel 2. I – doit on aga kõige parem lahendus, kuna vaatamata sellele, et sellel tarkvaralahendusel on minimaalsed nõuded süsteemile (aga üheks nõudeks on ressursiefektiivse süsteemi loomine, vt jaotis 2.2), sisaldab see palju erinevaid eelnevalt defineeritud KE-de tüüpe (mis võimaldab edaspidi arendada süsteemi), võimaldab vaadata KE-de omavahelisi suhteid ja kasutada lisaks seiretarkvara. Samas nõuete kohaselt peab olema sisseehitatud inglisekeelne liides, mis on olemas mõlemal tootel. iTop'il on aga madalam hinne analüüsi lõpus ja põhjuseks on see, et selle lahenduse esmaseks eesmärgiks ei ole sisuka ja korraliku KAB-i olemasolu, vaid IT üksuse töö keskendumine ühes kohas.

2.6 Olemasoleva olukorra eelised ja puudused

Antud olukorra eeliseks on see, et turul on mitu head toodet avatud lähtekoodiga, mis aitab ettevõttel raha kokku hoida. Eeliseks on ka see, et alati võib oma lisamooduleid sisse ehitada, kuna need on avatud lähtekoodiga tooted.

Puuduseks on aga see, et avatud lähtekoodiga tooted ei ole automatiseeritud. Üldjuhul need on paremad lahendused, kui Excel tabeli kasutamine, aga töö jääb mehhaaniliseks ning töötajad peavad käsitsi info sisestama. Kuna andmete sisestamise protsess ei ole automatiseeritud, tähendab see et andmed KE-de kohta ei pruugi olla aktuaalsed ja probleem, mis oli kirjeldatud peatükis 2.1 jääb samaks.

2.7 Täpsem ülesande püstitus

Ettevõtte IT keskkond on väga keeruline. See hõlmab palju varasid ja keerulisi seoseid inimeste, protsesside ja tehnoloogiate vahel. Peale seda, IT valdkond muutub pidevalt. Samuti muutuvad ka IT taristu komponendid ja sõltuvused. Inimesed tulevad, lähevad ära ja muudavad rolle. Protsessid muutuvad. Virtualiseerimine oluliselt kiirendab muutmise protsessi. Seega KAB-i käsitsi täitmine ja uuendamine ei ole otstarbekas. Samuti väga oluline on mitte ainult see, et KAB sisaldaks mitte ainult kogu vajalikku infot, vaid ka tagada, et see info oleks aktuaalne. Selleks, et rahuldada seda vajadust, peab KAB toetama automaattuvastust, mis tuvastab kõiki varusid IT taristus, sh järgmisi:

- Füüsiline vara – kogu riistvara, k.a kliendid, serverid, suurarvutid, võrguseadmed ja koguda infot konfiguratsioonide, soojuse hajumise ja energiatarbe kohta
- Virtuaalne vara – kogu virtuaalne vara (virtualiseerimistehnoloogial põhinev vara)
- Tarkvara – kogu tarkvara, k.a rakendused ja andmebaasid

Kuna kõikide varade tuvastamise realiseerimine ei ole võimalik suure töömahu tõttu, käesoleva töö käigus peab olema realiseeritud teine punkt, ehk virtuaalse vara tuvastamine ja registreerimine andmebaasi. Töö väljundis peab olema töötav lahendus virtuaalse vara tuvastamise ja KAB-i lisamisega. KAB-ina peab kasutama peatükis 2.4 valitud lahendust.

Lisaks virtuaalmasinate lisamisele peab lisanduma masinate kohta järgnev info: *RAM*, *CPU*, võrgu ja virtuaalsete ketaste kohta info, mis omakorda peavad olema seotud virtuaalmasinaga, ehk kuuluma sellele. Juhul, kui andmed virtuaalserverites muutuvad, peavad kirjed KAB-is nende kohta muutuma ka.

Üleval kirjutatud kokku võttes täpsustakse jaotises 2.2 esitatud nõue FN02 ja pärast muutmist on see (vt Tabel 8):

Tabel 8. Kasutusjuhu FN02 ümberformuleerimine.

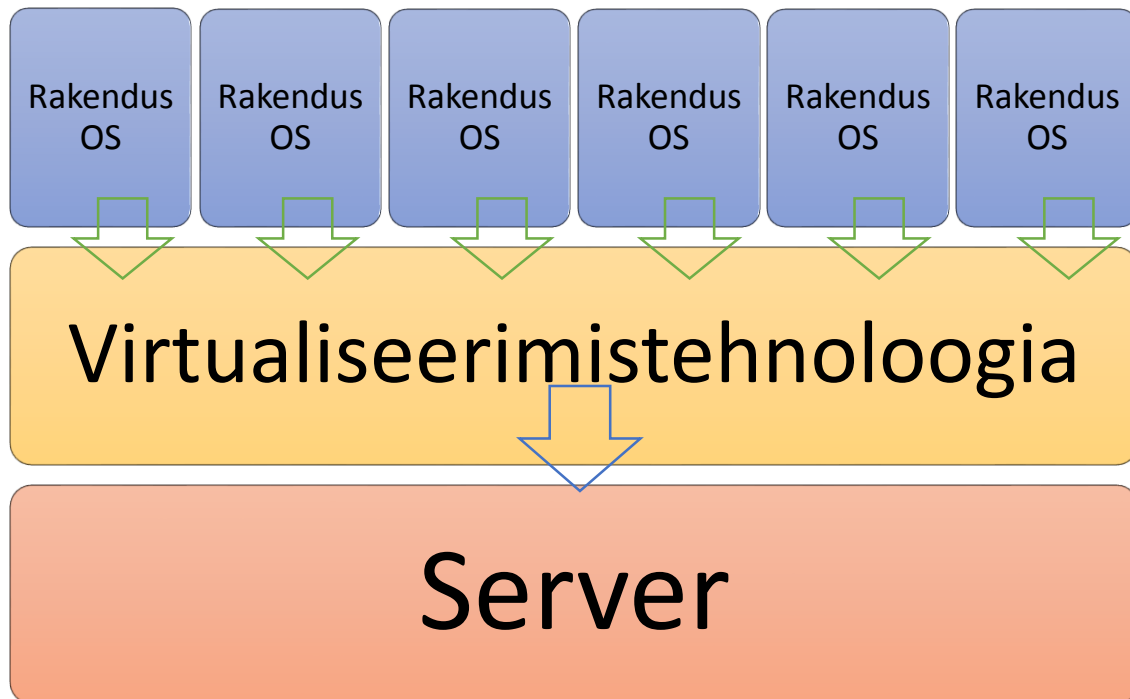
Kasutusjuhu ID:	FN02
Kasutusjuhu nimetus (Eesmärk):	VM-te ja nendega seotud KE-de (RAM, CPU, võrk, virtuaalsed kettad) lisamine süsteemi (eesmärgiks on tagada IT üksuste töötajate teadlikkuse ettevõttes kasutatavate ressursside kohta)
Tegutsejad:	Kasutaja, süsteem
Eeltingimused:	Olemas andmebaas, kuhu lisatakse andmed
Järelingimused:	Andmebaasi on lisatud andmed VM-te ja nendega seotud KE-de kohta
Põhistsenaarium	
Kasutaja	Süsteem
1. Kasutaja käivitab andmete lisamise protsessi	2. Süsteem sisestab automaatselt andmed VM-te ja selle komponentide kohta andmebaasi

3 Alternatiivid ja lahenduse valik

Peatükis 2.7 sai formuleeritud käesoleva töö täpsem ülesandepüstitus, ehk peab looma lahenduse, mis lisab automaatselt KAB-i andmed virtuaalmasinate ja nende komponentide kohta, juhul kui andmed said muudetud, peab KAB-is olev info muutuma ka, ehk vastama tegelikkusele. Kuna virtuaalmasinate ja riistvara vahel on olemas virtualiseerimistehnoloogia, antud peatükis valib autor virtualiseerimistehnoloogia (nagu oli otsustatud jaotises 2.7), mille jaoks hakkab autor lahendust looma, kuna ühe lahenduse loomine, mis toetaks kõiki virtualiseerimistehnoloogiaid ei ole antud töös võimalik, sest võtab palju rahalisi ja mitterahalisi ressursse.

3.1 Olemasolevate lahenduste alternatiivid

Virtualisatsioon - tõestatud tarkvara tehnoloogia, mis võimaldab kasutada mitu operatsioonisüsteemi ja rakendust (virtuaalmasinat edaspidi VM) samas serveris samal ajal (vt Joonis 4). See aitab kaasa kokku hoida raha, ruumi ja efektiivselt kasutada ressursse [12]. Joonis 4 on kirjeldatud virtualisatsiooni olemus, ehk ühel kandjal (masinal) virtualiseerimistehnoloogia abil (kiht riistvara ja virtuaalmasinate vahel, mis võimaldab käivitada viimaseid) võib samaaegselt eksisteerida mitu operatsioonisüsteemi, kus jooksevad rakendused.



Joonis 4. Virtualiseerimise olemus.

Virtualisatsiooni eelised:

- Vähendab kulusid
- Tagab rakenduste käideldavuse
- Minimeerib või kõrvaldab maasolekuaega
- Suurendab IT tootlikkust, efektiivsust, agiilsust, ja reageerimisvõimet
- Kiirendab ja lihtsustab rakenduste ja ressursside seadistamist

Kuna ettevõttes on kasutusel mitu virtualiseerimistehnoloogiat, otsustas autor, et virtualiseerimistechnoloogia valimiseks, peab esmalt otsustama, mis operatsioonisüsteeme peab see võimaldama installeerida.

Kuna arendatav süsteem peab edaspidi automaatselt saama andmeid ükskõik mis operatsioonisüsteemide kohta, otsustati valida virtualiseerimistechnoloogia, mis on ühilduv kõige populaarsemate serverite operatsioonisüsteemidega. Arvestades selliseid omadusi, nagu familiaarsus ehk lähedus teiste lahendustega, kasutamise lihtsus, nõuded installitavale süsteemile ja stabiilsus, parimateks ja kõige kasutatavateks osutusid järgmised [13]:

1. RHEL ja CentOS
2. Fedora
3. Ubuntu
4. Debian
5. SUSE Linux
6. Windows Server (alates 2008)

Kui lisada nõuetele ka 64-bitiline arhitektuur ja ettevõtte jaoks versioon, siis tuleb välja, et antud kriteeriumitele vastab ainult üks lahendus – VMware vSphere 6.0. Kui vähendada nõuded ja eemaldada serverite operatsioonisüsteemidest Debian'i ja SUSE Linux'i (vastavalt numbrid 3 ja 5), siis võimalikele virtualiseerimistehnoloogiatele lisandub Microsoft Hyper-V Server 2012R2 [14].

3.2 Alternatiivide võrdlemine

Nagu peatükis 3.1 selgus, on hetkel 2 alternatiivi, mis vastavad eelmises peatükis sätestatud nõuetele. Järgnevalt võrreldakse neid omavahel. Kriteeriumid, mille alusel võrreldakse neid on loetletud Lisas 1, mis omakorda olid võetud IT toodete võrdlemise maatriksist [15].

Kui aga vaadata Tabel 9, võib näha, et kõik kriteeriumid, mille järgi hakatakse tooteid omavahel võrdlema, on jagatud 13 rühmaks.

Kui vaadata, millal tulid välja viimased toodete versioonid, võib näha, et VMware toode on palju kordi uuem. Kui vaadata ettevõtte seisukohalt, siis tavaliselt pidevate relüüside olemasolu tähendab pidevat toote arendust. Seega võiks oletada, et just see faktor põhjustab seda, et vSphere 6.0 on juba teist aastat esimene toode virtualiseerimisturul (kui võrrelda tooteid ettevõtete kontekstis). Samas aga vSphere litsents on mitu korda kallim, kui Microsofti litsents, kuna selle hind arvutatakse iga sokli pärast, HyperV iga kahe sokli tagant. Kui rääkida haldamiskuludest, kuhu kuuluvad tugiteenused, siis samas VMware toode on kallim, kui HyperV, aga VMware pakub rohkem erinevaid haldamisvõimalusi. Suureks HyperV eeliseks on see, et litsentsi hinna sees on lisaks veel kaks Windows toodete litsentsi. Kui siis rääkida virtualiseerimistehnoloogiate võimalustest, siis mõlemal tootel on palju erinevaid võimalusi külalisoperatsioonisüsteemide halduseks, mõnede funktsioonide kasutamiseks peab aga installima lisasid. Ettevõtte kontekstis on väga oluline, et oleks võimalus varukoopiate tegemiseks, kuna andmed suurtes ettevõtetes muutuvad nii kiiresti, et vahel ei saa füüsiliselt

andmekadude puhul taastada dokumentide, rakenduste jm eelmist versiooni. Samas on väga hea, kui virtualiseerimistehnoloogia oskab häda korral ise väikesi parandusi teha. Selliste võimaluste vaates on HyperV rohkem suunatud sarnaste probleemide lahendamisele ja nende vältimisele.

Kuna suurtes ettevõtetes ilmuvad väga kiiresti uued virtuaalmasinad, on väga suureks eeliseks, kui virtualiseerimislahendusel on olemas erinevate mallide kasutamise võimalus. Juba ettevalmistatud mallide ja profiilide olemasolu suurel määral kiirustab tööd ja uute masinate seadistust. Mõlemad tooted pakuvad selliseid võimalusi, aga kui vaadata Tabel 9 paigaldamisega seotuid võimalusi, võib näha, et VMware tootel mõned puuduvad, HyperV tootel on aga kõik loetletud võimalused olemas, kuid ei ole realiseeritud täies mahus.

Kui aga vaadata toodete lisavõimaluste poolest, võib näha (vt Tabel 9, punkt 6), et mõlemad tooted pakuvad palju erinevaid lisavõimalusi, mis võiks lihtsustada IT taristu vastutavate üksuste tööd. Väga heaks lisavõimaluseks on, näiteks, töövoode kasutamise võimalus, mis tähendab seda, et tehnoloogiad võimaldavad automatiseerida tööd, mida tegi tavaliselt inimene.

Kui vaadata Tabel 9 punkti 7, siis võib öelda, et mobiilsuse poolest on mõlemad tooted head, kuna võimaldavad migreerida andmeid, masinaid onlain režiimis, see tähendab seda, et töötajad võivad teha oma igapäevast tööd ja IT üksuste töö või projektid ei takista seda.

Ettevõtte kontekstis on väga oluline see, et ärikriitilistel süsteemidel oleks tagatud kõrge käideldavus. Kui nt. mingi süsteem läheb katki oleks väga hea, kui virtualiseerimistehnoloogia võimaldaks seda kiiresti taastada, et tagada kõrge käideldavus. Selle kriteeriumi poolest on mõlemad tehnoloogiad head, kuna nendes on olemas sisseehitatud võimalused kõrge käideldavuse tagamiseks nii rakenduste, kui ka VM-te tasemel.

Kui võrrelda virtualiseerimistehnoloogiad hostide ja VM-te seadistamise poolest, võib näha Tabel 9 (punktid 9 ja 10), et VMware toode võimaldab kasutada rohkem ressursse, millest omakorda otseselt sõltuvad kiirus ja tootlikkus. Kui vaadata erinevaid võimalusi seadistamise küsimuses, siis vSphere samuti pakub rohkem erinevaid variante VM-te seadistamiseks (näiteks, USB toetus, onlain riistvara lisamise võimalus).

Kui vaadelda virtualiseerimistehnoloogiaid koostalitlusvõime poolest, siis mõlemad lahendused pakuvad palju erinevaid võimalusi teiste süsteemidega integreerimiseks. See punkt on väga oluline selles töös, kuna edaspidi hakatakse looma lahendust, mis kasutaks

virtualiseerimistehnoloogia liidest. Nagu Tabel 9 on 11.4 punktist näha, vSphere pakub rohkem erinevaid võimalusi teiste süsteemide ühildamiseks. Samas aga paljud ettevõtted liiguvad pilveteenuste poole, ja Microsoft'i toode pakub sellist liidest, mis on mõeldud just pilves olevate toodetega integreerimise jaoks.

Kui võrrelda tooteid salvesti ja võrgu poolest (vt Tabel 9, punktid 12 ja 13), siis võib näha, et salvesti poolest on parem VMware, kuna sellel on rohkem võimalusi muutimise poolest, võrgu liidest poolest siis on rohkem võimalusi HyperV'l.

Tabel 9. Virtualiseerimistehnoloogiate võrdlemine.

Kriteerium	VMware vSphere 6.0	Microsoft HyperV 2012R2
1 HINNANG		
1.1 Viimase relüüsi kuupäev	02.2015	10.2013
1.2 Turupositsioon	1. koht [16]	2. koht
2 HIND		
2.1 Virtualisatsiooni hind(dollarites)	2 875/sokkel 604 (tugiteenused)	882 (kuni) 2 soklit
2.2 Haldamiskulud (dollarites)	4 995 –Standard versioon (piiramata hostide arv) 1 049 – Server Foundation (kuni 3 vSphere hosti) + tugiteenused 1 aasta 1 249 – Production 24/7	3 607 – Datacenter või 1 323 kuni 2 CPU
2.3 Külalisoperatsioonisüsteemide litsentseerimine	Puudub	Jah, 2 WIN op.süsteemi
3 VÕIMALUSED		
3.1 Tsentraliseeritud juhtimine	Jah (vCenter Server Standard + veebiklient)	Jah (System Center 2012R2)
3.2 Virtuaalne ja füüsiline juhtimine	Piiratud (pluginid)	Jah
3.3 Integreeritud rollipõhine ligipääsu kontroll	Jah	Jah
3.4 Veebipõhine haldamine	Jah, olemas eraldi klient	Piiratud
3.5 Detailne teenuste haldamine	Piiratud	Jah

Kriteerium	VMware vSphere 6.0	Microsoft HyperV 2012R2
4 UUENDUSED JA VARUNDAMINE		
4.1 Hüperviisori uuendamine	Jah	Jah
4.2 VM-te hädaparandus	Piiratud (Update Manager riistvara uuendamine)	Jah (Windows Server Update Services/System Center Configuration Manager)
4.3 VM hetkvõte	Jah	Jah
4.4 Integreeritud taastamise rakendusliides	Jah (vStorage API Data Protection)	Jah (VSS API)
4.5 Integreeritud taastamine	Jah (vSphere Data Protection)	Jah (WSB & DPM, Azure Backup)
5 PAIGALDAMINE		
5.1 Automatiseeritud hostide paigaldamine	Puudub	Jah
5.2 VM mallid	Jah	Jah, vähem
5.3 Hostide profiilid	Puudub	Jah (füüsiliste arvutite profiilid)
5.4 Salvestite profiilid	Jah	Osaliselt
6 TEISED OMADUSED		
6.1 Ressursside poolid	Jah	Jah
6.2 Füüsiline – Virtuaalne	Jah (VMware Convertor)	Puudub
6.3 Iseteeninduse portaal	Puudub	Jah

Kriteerium	VMware vSphere 6.0	Microsoft HyperV 2012R2
		(Uus – Azure Pack)
6.4 Töövood	Jah (vCenter Orchestrator)	Jah (SC Orchestrator, SC Service Mgr)
6.5 Turvalisus	Tasuta: ESXi Firewall, vShield Endpoint + lisad	Windows Security, Hyper-V Extensible Switch
6.6 Süsteemide haldamine	SNMPv3 + CIM, esxcli, vMA	WMI + PowerShell
7 VM MOBIILSUS		
7.1 VM-te onlain migreerimine	Jah	Jah
7.2 Migratsiooni ühilduvus	Jah (EVC)	Jah (Processor Compatibility)
7.3 Hooldusrežiim	Jah	Jah
7.4 Automaatne elu migratsioon	Jah, osaliselt	Jah
7.5 Võimsuste haldamine	Jah (Host Power Management)	Jah (Power Optimization)
7.6 Andmete migreerimine	Jah (Live Storage vMotion)	Jah (Live and Shared Nothing)
8 KÕRGE KÄIDELDAVUS JA TAASTAMINE		
8.1 Klatri suurus	Maks. 64 sõlme / 8 000 VM klatri kohta	Maks. 64 sõlme / 8 000 VM klatri kohta
8.2 Sisseehitatud kõrge käideldavus (VM-a restart)	Jah (VMware HA)	Jah

Kriteerium	VMware vSphere 6.0	Microsoft HyperV 2012R2
		<i>(NIC and unmanaged Storage failure detection)</i>
8.3 Automaatne VM lähtestamine	Jah <i>(VMware HA)</i>	Jah <i>(Heartbeat)</i>
8.4 VM <i>Lockstep</i> kaitse	Jah <i>(Fault Tolerance)</i>	Puudub
8.5 Rakenduse / teenuse kõrge käideldavus	Puudub	Jah
8.6 Tiražeerimine	Piiratud <i>(vSphere Replication + lisa SRM)</i>	Piiratud <i>(Hyper-V Replica)</i>
9 HOSTI SEADISTAMINE		
9.1 Maksimaalne VM-te arv	1024	1024
9.2 Maksimaalne protsessori konfiguratsioon füüsilises hostis	480	320
9.3 Maksimaalne muutmälu protsessori kohta	Piiramata	Piiramata
9.4 Maksimaalne RAM hosti kohta	12 TB	4 TB
9.5 Maksimaalne virtuaalne CPU VM kohta	128 vCPU	64 vCPU
10 VM SEADISTAMINE		
10.1 Maksimaalne RAM VM kohta	4 TB	1 TB
10.2 Järjestikpordid	32 porti	Olemas
10.3 USB toetus	Jah	Puudub

Kriteerium	VMware vSphere 6.0	Microsoft HyperV 2012R2
	(USB 1.x, 2.x and 3.x) kuni 20 USB seadet VM kohta	
10.4 Onlain komponentide lisamine	Jah (<i>CPU, Mälu, Kettad, SSD ...</i>)	Ainult kettad ja mälu
10.5 Dünaamiline / Over-Commit	Jah (<i>Memory Ballooning</i>)	Jah
10.6 Mälu lehtede jagamine	Jah, vaikumisi väljalülitatud	Puudub
11 KOOSTALITLUSVÕIME		
11.1 <i>OVF (Open Virtualization Format)</i> toetamine	Jah	Jah
11.2 Riistvara ühilduvus	Väga põhjalik	Ainult <i>Windows</i> süsteemid
11.3 Külalisoperatsioonisüsteemide toetus	Väga põhjalik	Vähem
11.4 Skriptimine / API'd	<i>Web Services API/SDK, CIM, Perl, .NET, Java SDKs, Client Plug-In API, vSphere Clip, vMA</i>	Jah (<i>WMI API, PowerShell 4</i>)
11.5 <i>CLOUD API</i>	Puudub	Service Provider Foundation API, Azure Service Management API
12 SALVESTI		
12.1 Toetatud mäluseadmed	<i>DAS, NFS, FC, iSCSI, FCoE (HW&SW), vFRC, SDDC</i>	<i>SMB3, virtual FC, SAS, SATA, iSCSI, FC, FCoE</i>
12.2 Jagatud failisüsteem	Jah	Jah
12.3 Bootimine kettavõrgult	Jah (<i>FC, iSCSI, FCoE, SW FCoE</i>)	Yah (<i>iSCSI, FC</i>)

Kriteerium		VMware vSphere 6.0	Microsoft HyperV 2012R2
12.4	Buutimine USB pealt	Jah	Puudub
12.5	Maksimaalne ketta maht	62 TB	64 TB
13 VÕRK			
13.1	VLAN	Oleams	Olemas
13.2	PVLAN	Puudub	Olemas
13.3	IPv6	Olemas	Olemas
KOKKU:		99p	91p
+2p	Täiesti rakendatav, olemas, toetatud		
+1p	On piirangid, vähem võimalusi		
0p	Ei toeta, puudub, väga suur vahe		

3.3 Alternatiivide analüüs ja parima alternatiivi valik

Punktis 3.2 olid loetletud kriteeriumid ja nende rühmad, mille järgi olid võrreldud kaks virtualiseerimistehnoloogiat: VMware vSphere 6.0 ja Microsoft HyperV 2012R2, mis omakorda olid parimad serverite virtualiseerimislahendused aastal 2015. [16]

Kõik kriteeriumid olid jagatud 13. rühma.

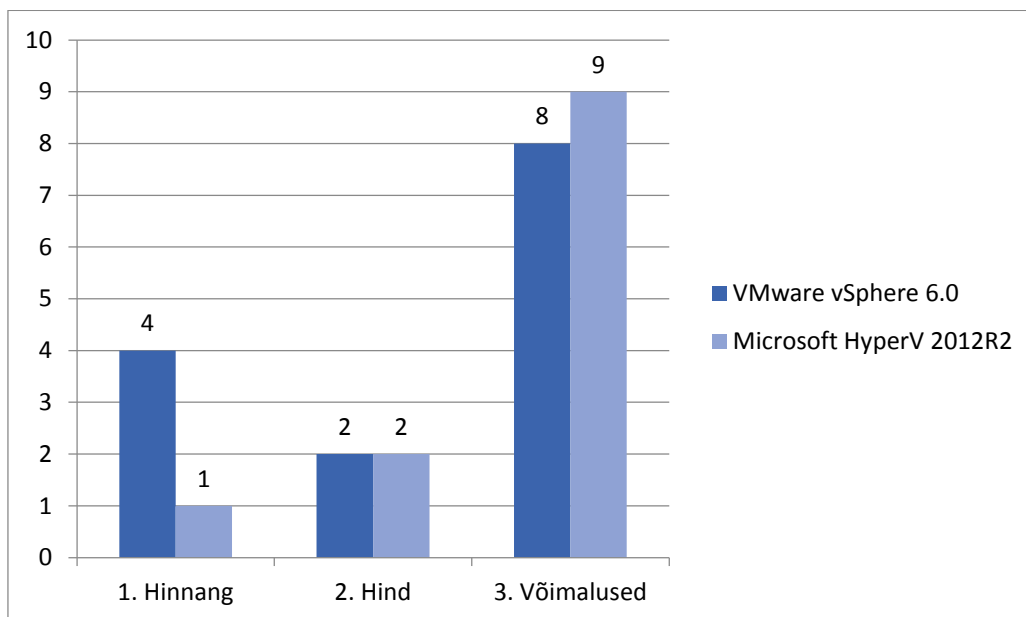


Diagramm 1. Virtualiseerimistoodete tulemused järgmistes kriteeriumite rühmades: 1. Hinnang 2. Hind 3. Võimalused .

Hinnang. Esmase hinnangu järgi on parim toode VMware vSphere 6.0 (vt Diagramm 1), kuna viimane versioon tuli välja eelmise aasta veebruaris. Microsofti toote viimane versioon oli aastal 2013 ja esmasel pilgul võib tunduda, et toodet ei arendata edasi. Samuti 2015 aasta seisuga on VMware vSphere 6.0 oli tunnustatud parimaks serverite virtualiseerimistooteks. Microsofti oma toode on aga teisel positsioonil. [16]

Hind. Hinna järgi on aga VMware toode kallim (vt Diagramm 1), kui Microsofti oma, samas VMware pakub rohkem erinevaid teenuseid.

Võimalused. Lisavõimaluste järgi on Microsofti oma toode parem (vt Diagramm 1), kui VMware oma, kuna pakub rohkem lisavõimalusi. Samuti võimaldab hallata VM-id ilma pliginat installimist.

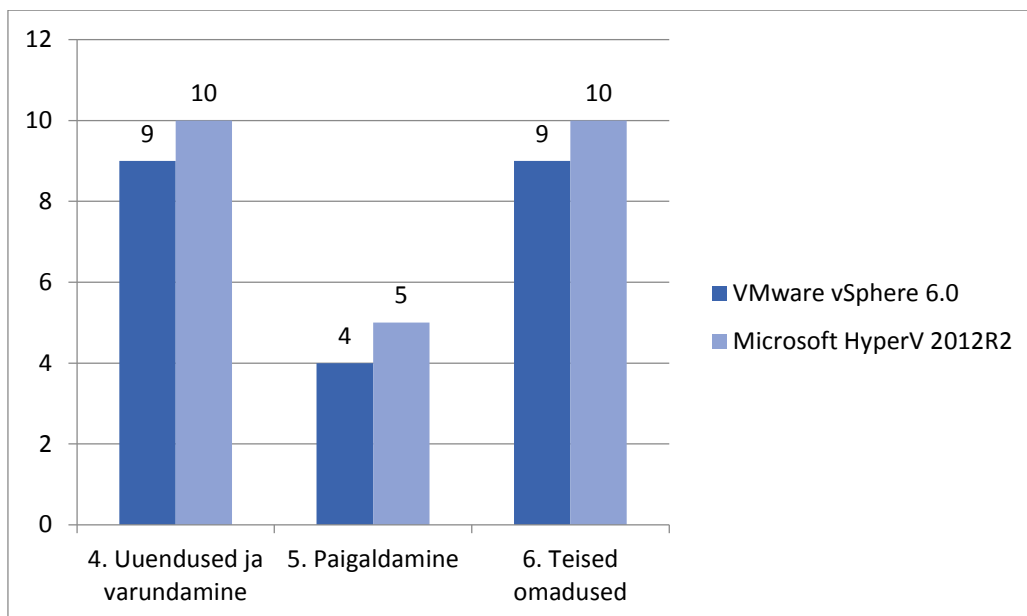


Diagramm 2 Virtualiseerimistoodete tulemused järgmistes kriteeriumite rühmades: 4.Uuendused ja varundamine 5. Paigaldamine 6. Teised omadused

Uuendused ja varundamine. Nagu Tabel 9 ja Diagramm 2 on näha, mõlemal tehnoloogial on hästi realiseeritud uuenduste ja varundamise funktsionaalsus, kuna on olemas rakendusliidesed ja juba integreeritud võimalused taastamise jaoks. VMware toote puuduseks on ainult see, et see pakub hädaparandust ainult riistavara tasemel.

Paigaldamine. Paigaldamise protsess on samuti natuke paremini realiseeritud Microsofti tootel (vt Diagramm 2), kuna see võimaldab automaatselt seadistada hoste.

Teised omadused. Selles kategoorias on ka natuke parem HyperV (vt Diagramm 2), kuna sellel tootel on olemas ka iseteeninduse portaal, seda kasutatakse tavaliselt pilvekeskkonnaga. Siin peaks mainima, et VMware tootel üldse puudub pilveteenuste jaoks rakendusliides. VMware aga pakub võimalust teisendada füüsilisi masinaid virtuaalseteks masinateks (VMware Converter abil), Microsofti tootel on aga selline funktsionaalsus puudub. HyperV samuti pakub rohkem turvalisusega seotuid võimalusi, VMware tootel on neid vähem integreeritud, aga saab kasutada lisasid.

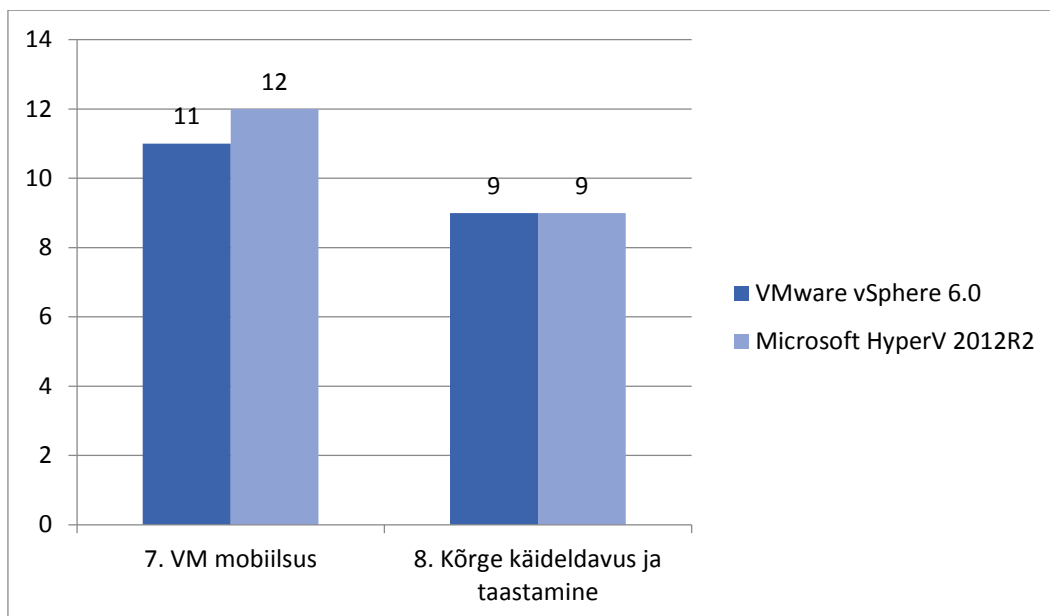


Diagramm 3 Virtualiseerimistoodete tulemused järgmistes kriteeriumite rühmades: 7. VM mobiilsus ja 8. Kõrge käideldavus ja taastamine

VM mobiilsus. Antud rühma kriteeriumite järgi on parem ka HyperV (vt Diagramm 3). Samas aga, kui vaadata migratsioonide täitmiseks kuuluvat aega, siis selgub, et vaatamata sellele, et HyperV on atraktiivsem, kuulub sellel rohkem aega andmete ja VM-te migreerimiseks [17], mis omakorda on parem mõõdik, kui lisafunktsionaalsus.

Kõrge käideldavus ja taastamine. Antud kategooriate rühmas on tooted saanud sarnased pallid (vt Diagramm 3), aga nt. VMware tootel puudub rakenduste kõrge käideldavuse tagamise võimalus, kuna see ei saa automaatselt avastada rakenduste probleeme. HyperV-l aga puudub võimalus kaitsta VM hosti rikke vastu ilma maasolekuta.

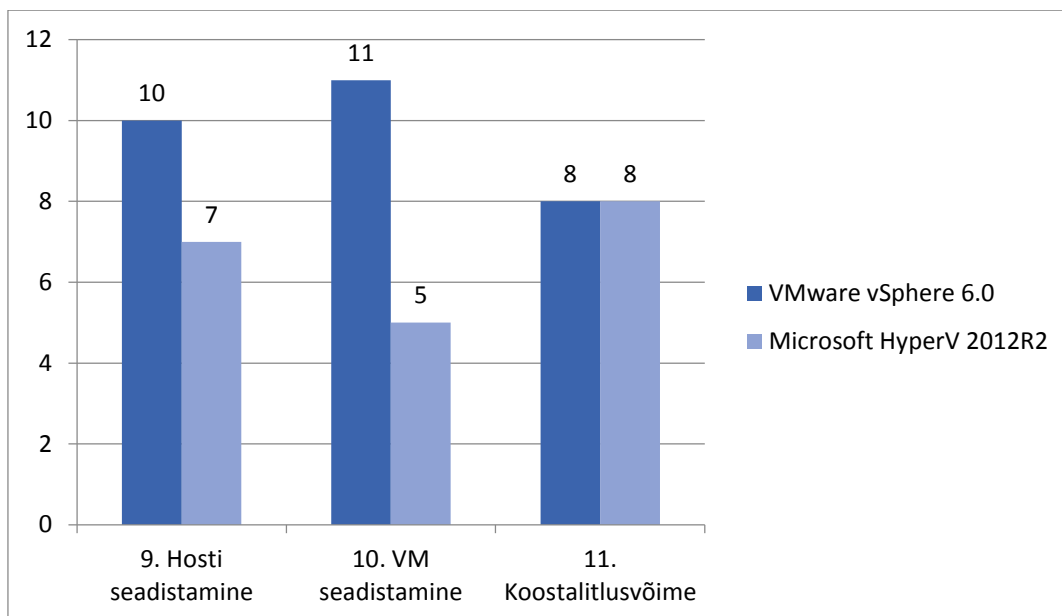


Diagramm 4 Virtualiseerimistoodete tulemused järgmistes kriteeriumite rühmades: 9. Hosti seadistamine
10. VM seadistamine 11. Koostalitlusvõime

Hosti ja VM seadistamine. Nagu on näha Diagramm 4, vSphere tulemused seadistamise kategoorias on mitu korda parem, kui HyperV'l. vSphere võimaldab kasutada rohkem ressursse hostidel ja VM-tel. Kui vaadata Tabel 9 tulemusi, siis võib näha, et mõned vSphere näitajad on mitu korda paremad, kui HyperV virtualiseerimistehnoloogial.

Koostalitlusvõime. Mõlemad lahendused sisaldavad palju erinevaid rakendusliideseid. Nagu eelnevalt oli öeldud, võrreldes HyperV'ga vSphere ei paku pilvelahenduste jaoks rakendusliidest. Microsofti lahendus pakub aga vähem võimalusi erinevate külalisoperatsioonisüsteemide installeerimiseks.

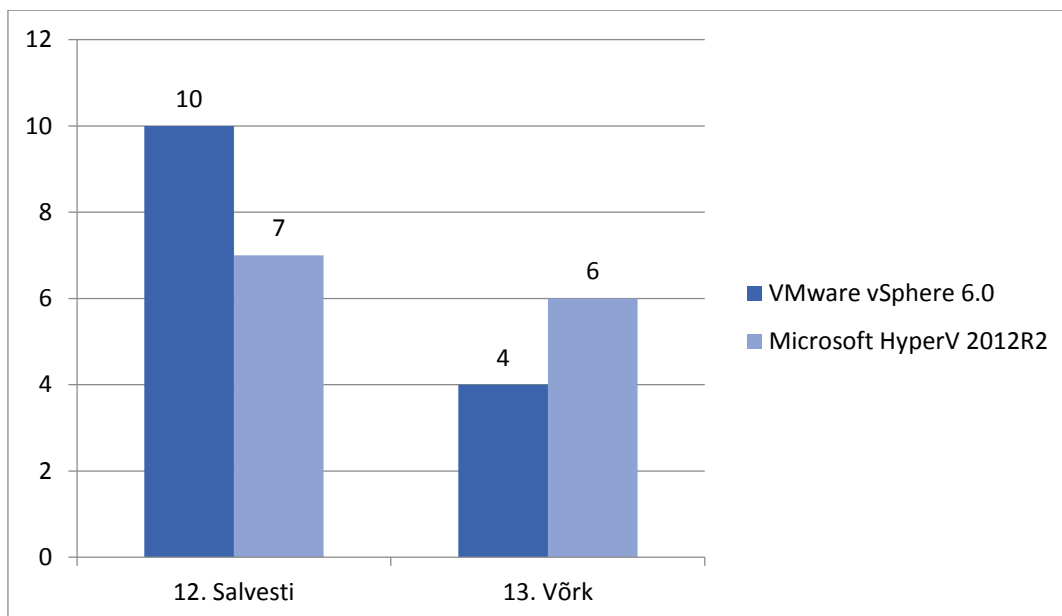


Diagramm 5 Virtualiseerimistoodete tulemused järgmistes kriteeriumite rühmades: 12. Salvesti ja 13. Võrk

Salvesti ja võrk. Nagu on Tabel 9 ja Diagramm 5 näha, vSphere toetab rohkem erinevaid mäluseadmeid. Üheks suuremaks eeliseks on see, et vSphere võimaldab bootida seadet USB pealt. Kui vaadata võrguga seotuid omadusi, võib näha, et vSphere ei toeta privaatsid võrke.

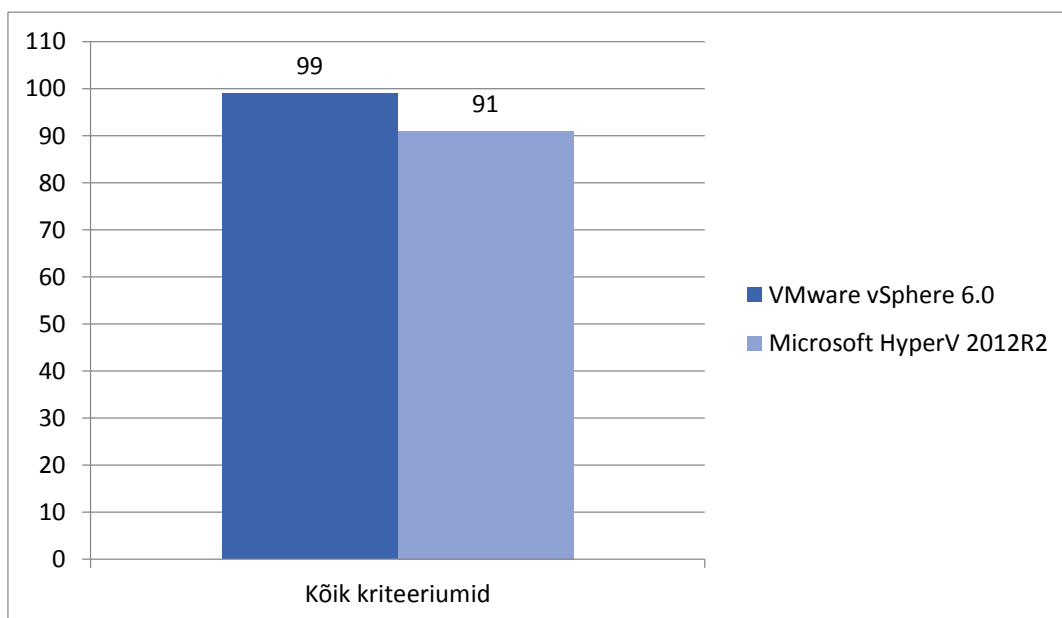


Diagramm 6 Võrdluse käigus kriteeriumite eest saadud punktid

Nagu eelnevalt oli öeldud vSphere ja HyperV on parimad serverite virtualiseerimistehnoloogiad turul. Punktis 3.2 tehtud võrdluse tulemusena on see, et

parimaks lahenduseks osutus VMware toode. Vahe tulemustes ei ole suur (vt Diagramm 6), aga suuremad erinevused tehnoloogiates olid hosti ja VM-te seadistamise kriteeriumites. VMware toote eeliseks on see, et see võimaldab kasutada rohkem ressursse, mis on väga oluline VM-te jaoks. Edaspidi hakatakse looma lahendust VMware vSphere 6.0 virtualiseerimistehnoloogia jaoks. Samas peab mainima, et ettevõtte, mille jaoks tehakse seda tööd, samuti kasutab VMware tooteid.

Kuna ülalpool nimetatud kaks toodet on parimad virtualiseerimislahendused, on palju lahkarvamusi selles osas, et kumb nendest on parim. Vaatamata sellele, et VMware toode on esimesel kohal ka maailmaturul [16], mõned otsustavad Microsoft'i toote kasuks, kuna sellel on olemas graafiline liides ja paar tasuta litsentsi. Samas mõnede jaoks on HyperV lihtsam omandamiseks, kui vSphere. [18]

Toetudes üleval kirjutatule, edaspidi hakkab autor looma lahendust vSphere toote jaoks, seega funktsionaalset nõuet FN02 võib täiendada ja nüüd on kasutusjuhu nimetuseks (vt Tabel 10):

Tabel 10. Funktsionaalse nõue lõplik formuleering.

Kasutusjuhu ID:	FN02
Kasutusjuhu nimetus (Eesmärk):	vSphere 6.0 virtualiseerimistehnoloogial põhinevate VM-te ja nendega seotud KE-de (<i>RAM</i> , <i>CPU</i> , võrk, virtuaalsed kettad) lisamine süsteemi (eesmärgiks on tagada IT üksuste töötajate teadlikkuse ettevõttes kasutatavate ressursside kohta)

4 Lahenduse realiseerimine ja hindamine

4.1 Realisatsiooni projekt

Järgnevalt on esitatud plaan, mille järgi saavutatakse punktides 2.2 ja 3.3 esitatud nõudeid

1. Süsteemide valimine, mille järgi hakatakse lahendust looma (sai valitud jaotistes 2.5 ja 3.3)
2. Lahenduse arhitektuuri väljatöötamine – tuginedes valitud süsteemidele töötakse välja lahenduse arhitektuuri
3. Lahenduse jaoks testkeskkonna loomine – tuginedes punktis 2 pakutud arhitektuurile seadistatakse vajalikku riist- ja tarkvara
4. Arendamine – kirjutatakse kood funktsionaalsete nõuete toetamiseks
5. Testimine – testitakse kõik nõuded, mis olid esitatud süsteemile punktides 2.2 ja 3.3

4.2 Realisatsiooni teostamine

4.2.1 Realisatsiooni arhitektuur

Eelnevates peatükkides (2.5 ja 3.3) olid valitud tooted, mille peal hakatakse lahendust looma.

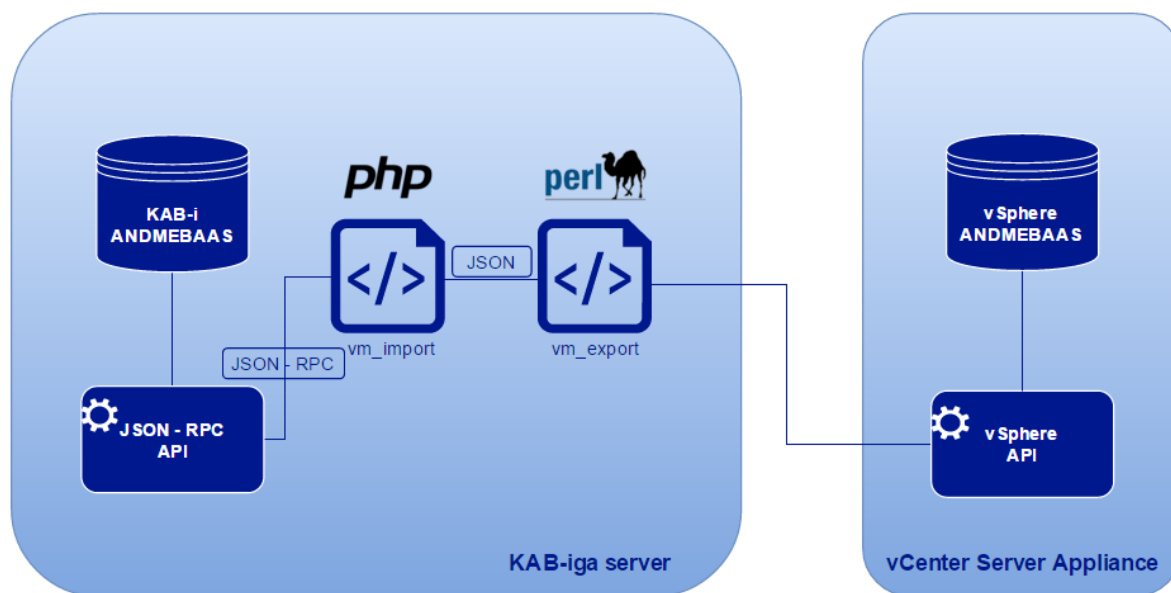
i-doit tootjal on olemas importimiseks mõeldud rakendusliides (edaspidi API), mis on kirjutatud PHP keeles [19].

Kuna i-doit KAB-is on palju erinevaid KE-de tüüpe, kõik need on defineeritud eraldi PHP klassidena (vt Joonis 5). Iga KE-i loomiseks on ettenähtud omaette klass. Juba eelnevalt defineeritud klasside olemasolu lihtsustab ja kiirendab käesoleva töö autori lahenduse väljatöötamise protsessi.

.gitkeep	0 KB	20.11.2015 18:37:53
Access.php	6 KB	27.04.2016 20:38:46
Accounting.php	14 KB	27.04.2016 20:38:48
Address.php	7 KB	27.04.2016 20:39:03
Aircraft.php	3 KB	27.04.2016 20:39:04
Application.php	12 KB	27.04.2016 20:38:45
AssignedCards.php	3 KB	27.04.2016 20:38:56
AssignedLogicalUnit.php	3 KB	27.04.2016 20:38:57
AssignedWorkstation.php	4 KB	27.04.2016 20:38:57
Audit.php	10 KB	27.04.2016 20:38:57
Backup.php	6 KB	27.04.2016 20:38:46

Joonis 5. Näide PHP klassidest, mis on eelnevalt API's defineeritud.

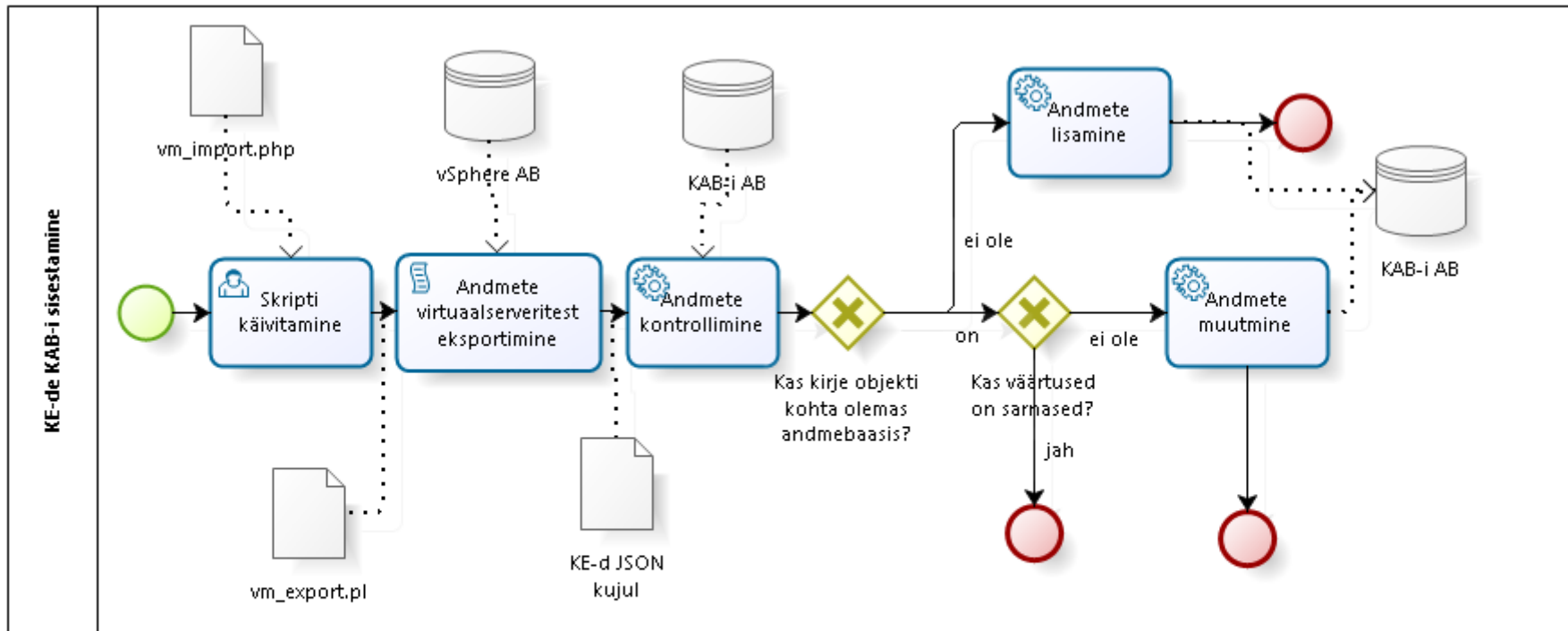
VMware VSphere 6.0 virtualiseerimistehnoloogia jaoks on olemas VSphere Perl SDK [20], mis võimaldab eksportida andmed. Antud töös on kasutatud 6.0 SDK versioon, seega lahenduse arhitektuur on kujutatud Joonis 6 (diagramm on tehtud kasutades draw.io tarkvara [5]):



Joonis 6. KAB-i ja vSphere andmevoog.

Nagu Joonis 6 on näha, eksporditakse virtuaalmasinate kohta andmed JSON kujul. Iga sisend (KE) on JSON'i objekti kujul, mille omadustes on jaotused, alamjaotused ning väärtused. i-doit API ja VSphere Perl SDK installitakse ühte serverisse, kus asub KAB.

Kogu protsessi kirjeldab järgmine tegevusdiagramm (vt Joonis 7, tehtud BPMN notatsioonis kasutades Bizagi Modeler tarkvara):



Joonis 7. Protsessi tegevusdiagramm.

Nagu on Joonis 7 näha, autoriseeritud kasutaja käivitab skripti (vm_import.php) mis omakorda kutsub välja vm_export.pl skripti, mis korjab vajalikke andmeid vSphere andmebaasist ja tagastab neid JSON kujul. Edaspidi toimub andmete kontrollimine ja siis on kaks stsenaariumi:

1. Kui andmebaasis ei ole kirjeid saadud KE kohta, siis vastavad andmed sisestatakse KAB-i andmebaasi. *Nt. kui KAB-is ei ole veel ühtegi virtuaalmasinat nimega „test-appliance“, siis see kohe lisatakse andmebaasi koos selle elementidega (nt. RAM-i suurus, virtuaalsed kettad, kirjeldused jm.).*
2. Kui andmed KE kohta KAB-i andmebaasis olemas ja
 - a. need on sarnased saadud andmetega, siis muudetusi ei tehta. *Näiteks, kui JSON-formaadis tuli virtuaalmasin nimega „test-appliance“, võrreldakse nende nimetused ja kirjeldused, kui kirjed on sarnased, siis edaspidi võrreldakse tulnud VM ja KAB-is oleva VM elemente, nt. RAM'i suurus, virtuaalsete ketaste kohta info ja kui kõik andmed on sarnased, siis muudatusi KAB-is ei tehta.*
 - b. need on erinevad (nt. andmed alamkategoriate kohta või VM kohta info), siis neid muudetakse. *Näiteks, kui JSON-formaadis tuli VM sama nimega, aga selle elemendid (nt. RAM'i surus ja ketaste mahud) tulid teiste väärtustega, siis muudetakse kirjed VM elementide kohta.*

4.2.2 Testkeskkonna loomine

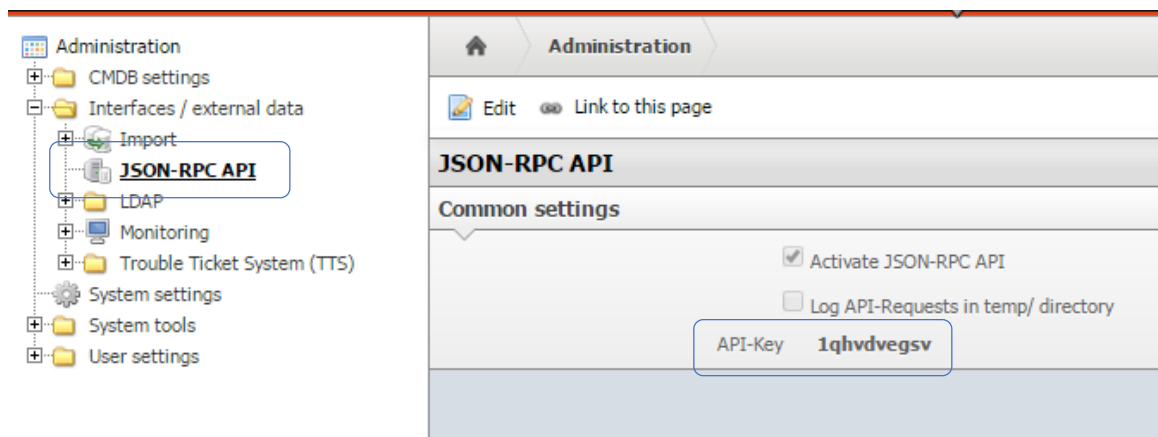
Selleks, et pakutud lahendust saaks hiljem testida, pidi autor looma keskkonna, mis oleks lähedane ettevõtte keskkonnale.

Antud töös kasutati Dell PowerEdge R610 püstikuserverit, kus on 8 CPU'd, ja installitud 98 291 RAM. Selle peale sai installitud VMware vSphere 6.0, nagu otsustati jaotises 3.3.

Selleks, et hosti oleks mugavam hallata ja edaspidi andmeid eksportida, installiti selle peale VMware vCenter Server Appliance 6.0, mille installeerimine koosneb kümnest sammust, sammud 5 ja installeerimise kokkuvõte on toodud välja Lisas 2. Pärast Appliance'installimist lisandub hostile automaatselt uus virtuaalmasin. Lisaks sellele sai lisatud testimiseks veel üks teise operatsioonisüsteemiga masin ehk kokku kaks virtuaalset masinat.

KAB-i tarkvara ja tarkvara API-desse ühendamiseks said installitud sülearvutile CentOS operatsioonisüsteemile. Sülearvuti ketta maht on 128GB ja 8GB RAM. Lisaks nendele said installitud lisatarkvara VSphere Perl SDK töötamiseks [21].

Kuna i-doit API ei ole automaatselt sisselülitatud, pidi selle sisse lülitama ja installima kliendi (vt Joonis 8), kus API-Key on võti ühendamiseks andmete sisestamiseks andmebaasi.



Joonis 8. i-doit'is API sisselülitamine.

API-ga ühendamiseks on olemas eraldi klass, kus on defineeritud link, mille kaudu saab ühenduda KAB-iga, kasutades API-võtit.

4.2.3 Arendamine

Nagu oli juba eelnevalt öeldud jaotises 4.2.1 esmalt toimub andmete eksportimine virtuaalmasinatest (vt Joonis 6). Eksportimise skript kirjutatakse Perl keeles, kuna antud töös kasutataval virtualiseerimistehnoloogial on olemas selleks vastav SDK.

Üldine skripti struktuur koosneb viiest sammust ja on järgmine:

1. vSphere SDK importimine, selleks, et kasutada Perli

```
use VMware::VIRuntime
```

2. Ühendumine serveriga

```
Util::connect ()
```

3. Objektide vaatlemine lokaalse serveri pool

```
# Anna välja kõik olemasolevad objektid, mis vastavad tüübile,
kus Vim::find_entity_views loob lokaalset Perl objekti(viidete
masiivi)
my $entity_type = Opts::get_option('entity');
my $entity_views = Vim::find_entity_views(view_type =>
$entity_type);
```

4. Andmete töötlemine ja tulemuste nt. väljaprintimine konsooli

```
foreach my $entity_view (@$entity_views) {
my $entity_name = $entity_view->name;
Util::trace(0, "Found $entity_type: $entity_name\n");
}
```

5. Ühenduse lõpetamine

```
Util::disconnect()
```

Nagu eelnevalt oli kirjutatud, Perl skripti kutsub välja PHP skript. Skripti kutsutakse välja parameetritega: server (\$vcServer), kasutaja (\$vcUser, enne '@' peab olema '\', kus vsphere.local on domeen), parool (\$vcPass), kus need on:

```
// vcenter server connection
$vcUser = 'administrator\@vsphere.local';
$vcPass = 'Muuda123.';
$vcServer = '192.168.1.208';
```

PHP skripti sees vm_import.php kutsutakse välja Perlis kirjutatud skripti vm_export.pl, selleks, et saada andmed KE-de kohta (kirjed VM-te ja nendega seotud objektide kohta), mis asuvad kandjal. Skripti välja kutsumine:

```
// get cluster export from perl script
exec("/usr/bin/perl /home/export/vm_export.pl --server $vcServer
--username $vcUser --password $vcPass",$output);
```

Järgmise sammuna tahetakse saada kõiki VM-aid ja saada nende kohta ainult loetletud andmed. "VirtualMachine" piirang tähendab seda, et tahetakse saada infot ainult VM-te kohta. Samas täpsustakse, mis andmed tahetakse näha VM kohta. Kui ei leita kirjeid VM-te kohta, antakse välja vastav teade.

```
# Fetch all VirtualMachines from SDK, limiting the property set
my $vmViews = Vim::find_entity_views(view_type =>
"VirtualMachine",
                                properties => ['name',
'runtime', 'datastore', 'guest', 'network', 'config',
'summary.storage', 'customValue', 'availableField']) ||
die "Failed to get VirtualMachines: $!";
```

Edaspidi toimub andmete kogumine ja kirjutamine JSON väljundi jaoks. Edaspidi on näide, kus kogutakse info virtuaalse riistvara kohta, kus vaikimisi määratakse tarnijaks VMware, kuna antud töös vaadeldakse ainult selle virtualiseerimistehnoloogiat, info *CPU* ja *RAM*'i kohta. Edaspidi kirjutakse andmed kategooriate ja alamkategooriate järgi:

```
# Print out disks and network devices
my $devices = $vm->get_property('config.hardware.device');
my $vendor = "VMware";
my $numCpu = ($vm->get_property('config.hardware.numCPU'));
my $vCpuType = "vCPU";
my $vhwVersion = ($vm->get_property('config.version'));
my $vRam = ($vm->get_property('config.hardware.memoryMB'));
my $vRamType = "vRAM";

$vmHash{"VirtualHardware"}{"Vendor"} = $vendor;
$vmHash{"VirtualHardware"}{"Version"} = $vhwVersion;
$vmHash{"VirtualHardware"}{"Cpu"}{"Type"} = $vCpuType;
$vmHash{"VirtualHardware"}{"Cpu"}{"NumCpu"} = $numCpu;
$vmHash{"VirtualHardware"}{"Ram"}{"Type"} = $vRamType;
$vmHash{"VirtualHardware"}{"Ram"}{"Capacity"} = $vRam ;
```

Kui info vajalike KE-de kohta sai korjatud, teisendatakse kogu info JSON formaati, mis tagastatakse PHP skripti `vm_import.php`:

```
my $vmsJSON = encode_json \%vmsHash;
printf($vmsJSON);
```

Info tuleb sellisel kujul (vt Joonis 9):

```

"VirtualHardware":
  {"NetworkAdapter":
    {"NIC1":
      {"Type": "VirtualVmxnet3",
       "Mac": "00:0c:29:b9:ee:27",
       "Network": "VM Network",
       "Connected": "True",
       "Status": "ok",
       "StartConnected": "True"}},
    "Disk": {
      "HDD1":
        {"Datastore": "datastore1",
         "Type": "VMDK",
         "FileName": "test-appliance/test-appliance.vmdk",
         "IsThin": "True",
         "SCSIaddress": "SCSI0:0",
         "Capacity": "12582912"},
      "HDD6":
        {"Datastore": "datastore1",
         "Type": "VMDK",
         "FileName": "test-appliance/test-appliance_6.vmdk",
         "IsThin": "True",
         "SCSIaddress": "SCSI0:5",
         "Capacity": "10485760"},
      "HDD8":
        {"Datastore": "datastore1",
         "Type": "VMDK",
         "FileName": "test-appliance/test-appliance_8.vmdk",
         "IsThin": "True",
         "SCSIaddress": "SCSI0:8",
         "Capacity": "10485760"},
    }
  }

```

Joonis 9. JSON'is VM-te info.

Edaspidi kõik teisendused ja KAB-i lisamine toimub PHP keeles. JSON kujul saadud andmed dekodeeritakse ja edaspidi luuakse objektid ja nendele määratakse saadud väärtused.

```

foreach ($output AS $json)
{
    $vms = json_decode($json);
    foreach ($vms AS $vm) {
        $vmHost = $vm->Runtime->Host;
        $vmName = $vm->Name;
        print ("Check vm $vmName \n");
        // VM name, object
        // $cmdbCiId = $vm->CMDB_CI_ID;
        $objID = ($l_objectApi-
>createOrGetObjectid($vmName,
ObjectTypeConstant::TYPE_VIRTUAL_SERVER));

        // RAM
        $vmRamSize = floatval($vm->VirtualHardware-
>Ram->Capacity);
        // $vmRamType = $vm->VirtualHardware->Ram-
>Type;
        check_memory($vmRamSize, $objID,
$l_apiClient);

        // CPU
        $cpuInfo = $vm->VirtualHardware->Cpu;
        check_cpu($cpuInfo, $objID, $l_apiClient);

        //...jne
    }
}
}

```

Kõige olulisem on siis muudatuste registreerimine. Nagu oli jaotises 4.2.1 kirjeldatud, antud skript ei tühjenda andmebaasi ja siis sisestab infot, vaid võrdleb omavahel seda, mida saadi, käivitades `vm_export.pl` tulemusena ja seda, mis on juba andmebaasis.

Selleks, et võrrelda juba andmebaasis olevat infot luuakse ühendus. Testsüsteemi juhul:

```

$l_apiClient = new ApiClient(
    new ApiConnection(
        $api_entry_point, $api_key, 'admin', 'admin'
    )
);

```

Ühenduse loomiseks peab olema eelnevalt lubatud API ühendus (vt Joonis 8) ja siis võetakse andmed klassist `config.php`, kus on defineeritud testitava süsteemi puhul:

```

$api_entry_point          = 'http://192.168.1.252/i-
doit/src/jsonrpc.php';
$api_key                  = '1qhvdvegsv';

```

Kui ühendus KAB-iga on loodud, käivitatakse kontroll. Näiteks edaspidi on kirjutatud meetod, mis kontrollib *RAM*'i. Esmalt võetakse KAB-ist olemasolevad andmed *RAM*'i

kohta ja siis hakatakse kontrollima. Kui andmed juba on KAB-is, siis ja sarnased, siis midagi ei muudeta, kui andmed muutusid, siis need muudetakse ka.

```
function check_memory ($ramSize, $id, $l_apiClient)
{
    $l_category      = new CMDBCategory($l_apiClient);
    $cmdbRam = $l_category->get($id,
CategoryConstant::G_MEMORY);
    print " check ram\n";
    $cleanup = false;
    $isInCmdb = false;
    if (!(empty($cmdbRam))){
        foreach ($cmdbRam as $key => $ram) {
            if ( $isInCmdb && ($ram['capacity']['title']
== $ramSize) && ($ram['unit']['title'] == 'MB') ) {

                @$cmdbRam[$key]['TODO'] = 'DELETE';
                print " duplicate: ".
$ram['capacity']['title'] . " " . $ram['unit']['title']. "\n";
                $cleanup = true;
            }
            elseif ( ($ram['capacity']['title'] ==
$ramSize) && ($ram['unit']['title'] == 'MB') ) {
                $isInCmdb = true;
                @$cmdbRam[$key]['TODO'] = 'OK';
            }
            else {
                $cleanup = false;
                if ( !(@$cmdbRam[$key]['TODO'] ==
'OK') ) {
                    $cleanup = true;
                    @$cmdbRam[$key]['TODO'] =
'DELETE';
                }
            }
        }
    }
    if (!$isInCmdb) {
        add_memory($ramSize, 'MB', $id, $l_apiClient);
        $isInCmdb = true;
    }
    if ($cleanup){
        cleanup_cmdb($cmdbRam, CategoryConstant::G_MEMORY,
$id, $l_apiClient);
    }
}
}
```

Kui antud elementi ei ole KAB-is, siis toimub lisamine:

```

function add_memory($size, $unit, $id, $l_apiClient)
{
    print "  add ram: $size $unit\n";
    $l_category = new CMDBCategory($l_apiClient);
    $l_ramData = new CMDBCategory\G\Memory();
    $unitList = $l_ramData->getDialogUnit($l_apiClient);
    $unitId = idByTitle($unitList, $unit);
    $manList = $l_ramData-
>getDialogManufacturer($l_apiClient);
    $manId = idByTitle($manList, 'VMware');
    $l_ramData->setManufacturer($manId);
    $l_ramData->setType('vRAM');
    $l_ramData->setCapacity($size);
    $l_ramData->setTitle('vRAM');
    $l_ramData->setUnit($unitId);
    $l_category->add($id, $l_ramData);
}

```

Kui aga ei ole, siis kustutakse (üks meetod kõikide objektide jaoks):

```

function cleanup_cmdb($cmdbElements, $cmdbType, $id,
$l_apiClient)
{
    $l_category = new CMDBCategory($l_apiClient);
    print "  cleanup CMDB\n";
    foreach ($cmdbElements as $key => $cmdbElement) {
        if (@$cmdbElements[$key]['TODO'] == 'DELETE') {
            if ($cmdbType == "C__CATG__MEMORY") {
                $title =
@$cmdbElement['title']['title']. "-
".@$cmdbElement['capacity']['title']. "
".@$cmdbElement['unit']['title'];
            }
            elseif ($cmdbType == "C__CATG__CPU") {
                $title =
@$cmdbElement['type']['title'];
            }
            else {
                $title = $cmdbElement['title'];
            }
            $l_category->delete($id, $cmdbType,
$cmdbElement['id']);
            print "  delete: ".$title."\n";
        }
    }
}

```

Sama loogika järgi on tehtud antud töös kontrolli ja lisamisega meetodid järgmiste objektide kohta, mis on loetletud Tabel 11:

Tabel 11. Lahenduses käsitletud objektid ja nende kirjeldused.

Objekt	Vastav klass i-doit API's ApiCMDB\Category\	Lühikirjeldus
VM	G\VirtualMachine.php	Andmed virtuaalmasina kohta
CPU	G\Cpu.php	Andmed CPU kohta
Viide	G\Access.php	Link, mille kaudu saab VM-le liigi ja sellega seotud andmed
Üldinfo	G\General.php	Andmed staatuste kohta, kirjeldused jm üldinfo
Mudel	G\Model.php	Andmed tarnija, id, seerianumbri, jm kohta
Võrguketask	G\SubcatStorageDevice.php	Andmed ketaste, nende staatuste, kirjelduste jm kohta
Võrguliides	G\SubcatNetworkInterfaceP.php	Võrguliidesega seotud info, nt. tootja, mudel jm.
Võrguport	G\SubcatNetworkPort.php	Võrgupordiga seotud andmed portide tüüpide, olekute ja aadresside kohta

4.3 Testimine ja tulemuste analüüs

Selles peatükis testitakse süsteemile esitatuid nõudeid ja testide tulemuste järgi tehakse tulemuste analüüsi.

4.3.1 Testimine

Järgnevalt on esitatud Tabel 12 testide kirjeldused ja viited nõuetele. Funktsionaalsete nõuete testimiseks, ehk testid FNT01, FNT02, FNT03, FNT04 sooritakse kasutades *Selenium WebDriver* veebipõhist tarkvara. *Selenium* – tarkvara, mis aitab lihtsustada ja automatiseerida testimise protsessi. Selle eeliseks on see, et seda saab paigaldada lisana veebilehitseja sisse. Antud töös otsustati kasutada just seda tarkvara, kuna autoril on juba olemas sellega töötamise kogemus. See sobib testimiseks antud töös, kuna autori poolt väljatöötatud lahendus on veebipõhine.

Tabel 12. Vastuvõtutestid ja nende kirjeldused.

Test ID:	FNT01
Nõude ID:	FN01
Kirjeldus:	Testitakse süsteemi sisselogimist
Sisend:	Sisestatakse õiged kasutajatunnuse ja parooli

Oodatav väljund:	Kuvatakse kõik KAB-i kategooriad
Test ID:	FNT02
Nõude ID:	FN01
Kirjeldus:	Testitakse süsteemi sisselogimist vale parooliga
Sisend:	Sisestatakse kasutajatunnuse ja vale parooli
Oodatav väljund:	Näidetakse veateadet
Test ID:	FNT03
Nõude ID:	FN02
Kirjeldus:	Testitakse VM-te kohta andmete sisestamist andmebaasi.
Sisend:	Käivitatakse vm_import.php skripti
Oodatav väljund:	KAB-i lisatakse 2 virtuaalmasinat nendega seotud KE-ga
Test ID:	FNT04
Nõude ID:	FN03
Kirjeldus:	Testitakse andmete muutmist KAB-is, kui VM-te pool sai midagi muudetud
Sisend:	Muudetakse serveri pool ühe VM RAM-i suurust ja siis käivitatakse vm_import.php skripti
Oodatav väljund:	Pärast skripti käivitamist muutub KAB-is VM RAM'i suuruse väärtus
Test ID:	MFNT01
Nõude ID:	MFN01
Kirjeldus:	Kontrollitakse, kas lahenduses kasutatud süsteemid on avatud lähtekoodiga
Sisend:	Vaadetakse kasutatud süsteemi veebilehelt, kas sellel on olemas GNU AGPL (litsents avatud lähtekoodiga lahendustel)
Oodatav väljund:	Eeldatakse, et on olemas
Test ID:	MFNT02
Nõude ID:	MFN02
Kirjeldus:	Testida, kui palju ressursse kasutab pakutud lahendus
Sisend:	Kontrollitakse, kus on seadistatud lahendus ja vaadata, kui palju ressursse see kasutab
Oodatav väljund:	Vähem, kui 8GB RAM'i ja 500GB kõvaketta mahtu
Test ID:	MFNT03
Nõude ID:	MFN03
Kirjeldus:	Testitakse, kas süsteemis on inglisekeelne liides
Sisend:	Pannakse lahenduse sisu tõlkimisvahendi sisse
Oodatav väljund:	Tõlkimisvahend tuvastab, et süsteem on inglisekeelne

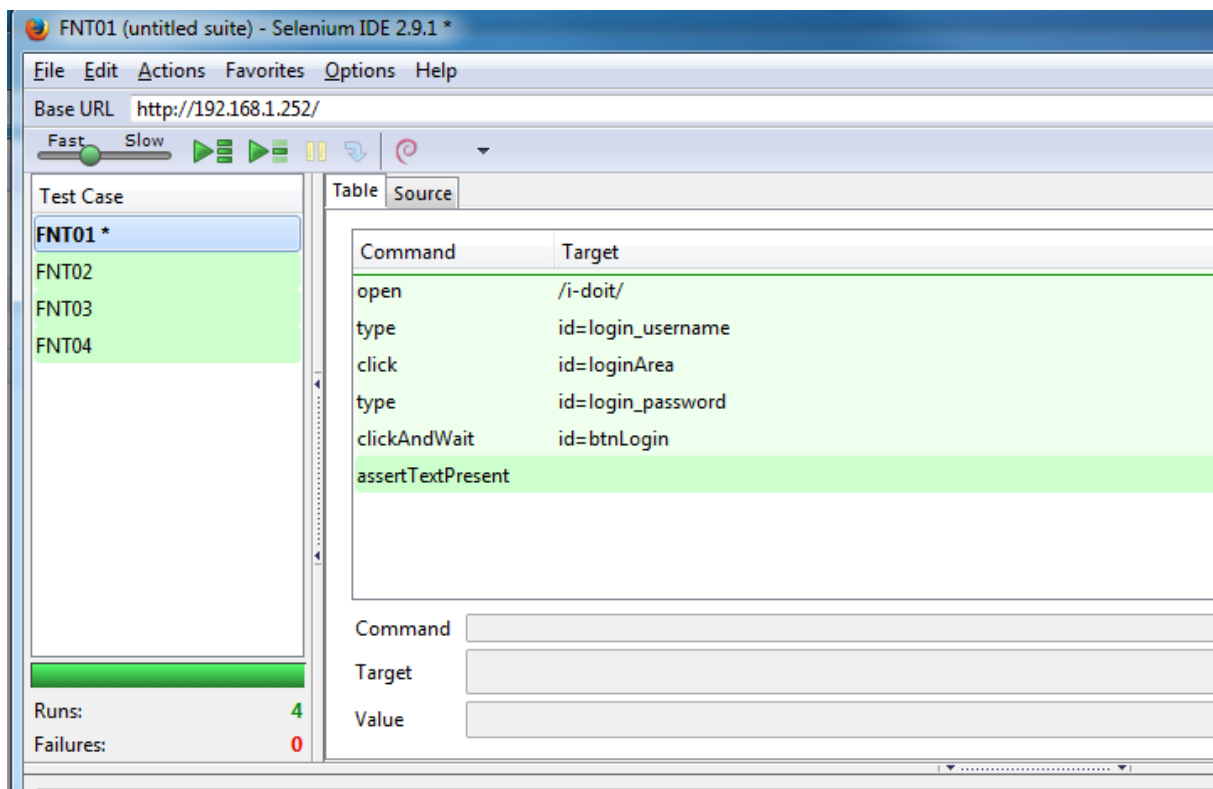
Testide tulemused on siis järgmised:

Testid **FNT01**, **FNT02**, **FNT03**, **FNT04** said läbitud, testiti *Selenium* tarkvaraga (vt Joonis 10), kokku oli koostatud 4 testi funktsionaalsete nõuete testimiseks. Kõik need said läbitud. Testid asuvad Lisas 3.

MFNT01 sai läbitud, kuna lahenduse aluseks kasutataval tarkvaral (i-doit) on olemas APLv3 litsents, seega see toode on avatud lähtekoodiga.

MFNT02 sai läbitud ka, kuna testitav süsteem oli paigaldatud sülearvutisse, kus oli 4GB RAM ja 126GB kõvaketas.

MFNT03 sai läbitud ka, kuna pärast teksti tõlkimisvahendisse sisestamist, tuvastas see, et kasutatud keeleks on inglise keel



Joonis 10. Funktsionaalsete testide läbiviimine Selenium tarkvaraga.

4.3.2 Tulemuste analüüs

Peatükis 4.1 oli koostatud realisatsiooni projekt eesmärgina luua suure ettevõtte jaoks lahendus, mis vastaks peatükkides 2.2 ja 3.3 loetletud nõuetele. Esimese sammuna valiti süsteemid, mille põhjal autor töötas välja lahenduse. Esimeseks süsteemiks sai valitud (vt jaotis 2.5) avatud lähtekoodiga KAB-i tarkvara i-doit, mis iseenesest on andmebaas, kuhu

saab lisada KE-te ja seoseid nende vahel. Teise sammuna pidi valima virtualiseerimistehnoloogia, mille põhjal luuakse lahendust. Kuna ettevõttes, mille jaoks antud lahendust tehti on kasutusel mitu erinevat virtualiseerimistehnoloogiat, siis otsustas käesoleva töö autor valida turul olemasolevate virtualiseerimistehnoloogiate vahel parima, mis tooks ettevõttele lahenduse. Selleks, et valida parim virtualiseerimistehnoloogia, autor loetles Lisas 1 kriteeriumid, mille järgi võrdles kahte parimat virtualiseerimistehnoloogiat. Pärast võrdlemist osutus parimaks tooteks (mis vastab loetletud kriteeriumitele) VMware vSphere 6.0, mis on olnud ka turuliider viimased kaks aastat.

Teise sammuna töötas töö autor välja tulevase süsteemi arhitektuuri. Autori pakutud arhitektuuri kohaselt toimub andmevahetus kahe serveri vahel. Ühes serveris asub KAB ja teises serveris on kõik andmed virtuaalmasinate kohta (andmebaasis), mis on üles ehitatud vSphere virtualiseerimistehnoloogial. Välja töötatud arhitektuuri kohaselt peab KAB serveris asuma kaks skripti. Üks nendest andmete teisest serverist eksportimiseks ja teine andmete KAB-i importimiseks.

Järgmise sammuna lõi antud töö autor testkeskkonna, mis oleks lähedane ettevõtte keskkonnale. Ehk lisaks vajaliku tarkvara paigaldamisele loodi kaks virtuaalmasinat, mille põhjal hakati edaspidi teste sooritama.

Järgmise sammuna said loodud kaks skripti (nagu oli juba eelnevalt kirjutatud), üks andmete eksportimiseks ja teine andmete importimiseks. Andmed eksporditakse JSON formaadis, kuna valitud KAB-i tarkvaral on olemas JSON-RPC rakendusliides. PHP keeles kirjutatud skripti sees toimub VM-te kohta saadud andmete võrdlemine KAB-is olemasolevate andmetega. Kui saadud väärtused (ehk need mis aktuaalsed) erinevad KAB-is olemasolevatest andmetest, siis KAB-i lisatakse need olemasolevate asemel, kui aga väärtused on sarnased, siis KAB-is olevat infot ei muudeta. Nii tagatakse KAB-is sisalduva info aktuaalsus.

Kui realisatsioon sai valmis, testis autor läbi kõik lahendusele esitatud nõuded. Funktsionaalsete nõuete jaoks kasutati Selenium tarkvara. Lahendusele esitatud nõuete testimise tulemusena said kõik testid läbitud, mis tähendab seda, et autori poolt pakutud lahendus vastab esitatud nõuetele ning antud töö tulemusena valmis automatiseeritud lahendus, mis kogub andmed VM-de ja nende komponentide kohta ning sisestab need

andmebaasi. Kui andmed said muudetud, siis muudetakse need ka KAB-is, mis tähendab seda, et ettevõttel on nüüd alati olemas aktuaalne info VM-de kohta, mis on üles ehitatud vSphere virtualiseerimistehnoloogial.

4.4 Pakkumised edasiseks arendamiseks

Nagu oli eelnevalt öeldud, KAB peab toetama automaattuvastust, mis tuvastab kõiki varusid IT taristus, mida võiks jagata kolmeks rühmaks:

- Füüsiline vara
- Virtuaalne vara
- Tarkvara

Antud töös oli realiseeritud automaatne KE-de lisamine KAB-i virtuaalse vara kohta, mis on omakorda realiseeritud VMware vSphere 6.0 virtualiseerimislahendusel. Edaspidi võib töötada välja skriptid teiste virtualiseerimistehnoloogiate jaoks, nt. Microsoft HyperV jaoks.

Edaspidi saab töötada välja skriptid füüsilise vara ja tarkvara jaoks. Näiteks töötajate kohta infot võiks saada ettevõtte domeenikontrolleritest, võrguseadmete kohta aga kasutades SNMP protokollid.

Antud teema on aktuaalne ja nõuab palju aega selleks, et saada täiuslikku lahendust, kuna ettevõtetes kasutatakse tooteid erinevate tarnijate poolt ja ühe reegli kasutamine kõigi seadmete jaoks ei ole võimalik. Teiselt poolt hea toote väljatöötamine, mis hõlmaks suurema osa KE-dest aitaks ettevõtet mahuhalduses, intsidendi- ja probleemihalduses, muudatusehalduses, konfiguratsiooni- ja reliisihalduses, kuna oleks aktuaalne KE-de andmebaas kõigi seostega, mis võimaldab saada hetkepilti kogu süsteemist.

5 Kokkuvõte

Käesoleva magistritöö eesmärk oli realiseerida kuluefektiivse automatiseeritud konfiguratsioonihalduse andmebaas ettevõtte ITIL protsesside toetamiseks. Eesmärgi täitmiseks kavandati ja seejärel realiseeriti olemasolevate virtuaalmasinate andmete automaatne sisestamine spetsiaalsesse andmebaasi, kus olid eelnevalt defineeritud IT taristu konfiguratsioonielemendid. Analüüsi käigus oli valitud parim konfiguratsioonihalduse andmebaasi lahendus, mis vastas ettevõtte poolt esitatud nõuetele. Seejärel oli valitud virtualiseerimistehnoloogia, mille jaoks töötas autor välja lahenduse. Autori poolt pakutud lahenduse jaoks oli loodud testkeskkond kahe virtuaalmasinaga, selleks, et testida lahenduse vastavust ettevõtte poolt esitatud nõuetele. Magistritöö tulemusena leiti, et konfiguratsioonihalduse andmebaasi automatiseerimine on võimalik. Autori arvates on automatiseeritud konfiguratsioonihalduse andmebaaside kasutamine aitab parendada järgmisi ITIL'is kirjeldatud protsesse: mahuhaldus, intsidendi-ja probleemihaldus ja muudatusehaldus.

Järgnevalt on nimetatud töö kõige olulisemad tulemused:

- Analüüsi käigus valiti parima avatud lähtekoodiga konfiguratsioonihalduse andmebaasi lahendus, mis võimaldab lisada erinevat tüüpi konfiguratsioonielemente
- Automatiseeriti konfiguratsioonihalduse andmebaasi (mis vajab veel täiendamist, kuid kõige olulisemad virtuaalmasinate elemendid lisanduvad automaatselt)
- Põhjalikult seletati uue lahenduse arhitektuuri ja olemust (enne ei olnud infot töös kirjeldatud toodete automatiseerimise kohta)
- Kirjeldati võimalikke pakkumisi konfiguratsioonihalduse andmebaasi edasiseks arendamiseks (füüsilise vara ja tarkvara kohta info automaatne sisestamine konfiguratsioonihalduse andmebaasi ja sama lahenduse väljatöötamine teiste virtualiseerimistehnoloogiate jaoks)

Töö kokkuvõtteks võib öelda, et antud lahenduse väljatöötamine oli väga huvitav ja hariv. Samas töö tegemisel oli palju takistusi, mis olid seotud vähese dokumentatsiooni ning näidete ja tarkvarade mitteühilduvusega. Vaatamata takistustele said eesmärgid

saavutatud ning töö väljundis on töötav kuluefektiivne automatiseeritud konfiguratsioonihalduse andmebaas, mis toetab vähemalt mahuhalduse protsessi.

Antud töö autor soovib tänada käesoleva magistritöö juhendajat Jekaterina Tšukrejevat igakülge osutatud abi eest, Viktor Larionovit ja Roman Pikalot huvitava teema ja suure abi eest.

Summary

The purpose of this master thesis is planning and development of a cost efficient system with automatic configuration management database (CMDB) integration for a large Estonian company to support ITIL workflows.

To achieve the objectives, we have developed a process of automatic addition of data about virtual machines to a special database, called CMDB, where types of configuration elements of IT infrastructure have been defined earlier. During the analysis stage the best open source CMDB solution that meets requirements of the company was selected. Then virtualization technology was chosen and the author began to develop a solution for the core problem that the subject company experienced. This core problem was that the data containing configuration elements were entered manually in various tables, and therefore did not necessarily correspond to reality, which lead to incorrect budgeting.

To test the solution proposed by the author a test environment with two virtual machines was created and it allowed to test whether the proposed solution met the set requirements. As the result of the conducted work the author has concluded that automation of CMDB is possible. In the author's opinion, using of automated CMDB can improve the following ITIL workflows: capacity management, change management, incident and problem management.

The following are the most significant results:

- During the analysis the best open source CMDB that allows to add different types of configuration elements was chosen
- Proposed a solution for automatization of CMDB (it still requires additional revisions, but the data about the most important elements of the virtual machines are added automatically)
- Described in detail the architecture and the essence of the new solution (previously there was no information about the automation products that are described in this paper)
- Describe possible areas for further development of the CMDB (automation of adding information about software and physical ware, and also developing a similar solution for other virtualization technologies)

It is worth noting that there were numerous obstacles in the course of the work, largely associated with the fact that little documentation and few examples of the selected in the work systems have been earlier published in this field. Despite the obstacles, the author achieved the stated goal and the core output of this thesis is a fully functional cost efficient automated system with CMDB, which supports, at least, capacity management process. First and foremost, I wish to thank my supervisor Jekaterina Tšukrejeva for her invaluable assistance, as well as Viktor Larionov and Roman Pikalo for providing a very engaging topic and assisting in the development phase.

Kasutatud kirjandus

- [1] „ITIL V3 terminite ja määratluste sõnastik,“
[WWW] http://www.itsmf.ee/itsmf/wp-content/uploads/2013/01/ITIL_V3_Glossary_100313.pdf. [Kasutatud 23 04 2016].
- [2] „E - teatmik,“
[WWW] <http://vallaste.ee/>. [Kasutatud 23 04 2016].
- [3] M. B. K., „Why Application Portfolio Reviews and CMDBs continue to be expensive?,“ 27 07 2015.
[WWW] <https://www.linkedin.com/pulse/why-application-portfolio-reviews-cmdb-continue-mohan-babu-k->. [Kasutatud 30 03 2016].
- [4] G. Roy, „The CMDB: The “Brain” Behind IT Business Value,“ 2010.
[WWW]
<http://www.compfort.pl/uploadfiles/press/8342759f9facbf33e8ba19195d4c8c317731d63281529e719047d1b5cceb0319.pdf>. [Kasutatud 30 03 2016].
- [5] „draw.io Diagram software,“
[WWW] <https://www.draw.io/>. [Kasutatud 01 02 2016].
- [6] „Open Source software for free,“
[WWW] <https://sourceforge.net/directory/business-enterprise/enterprise/bsm/cmdb/>. [Kasutatud 20 04 2016].
- [7] A. Bagul, „Configuration management database (CMDB),“ p. 1, 15 09 2015.
- [8] „iTop: open source ITIL ITSM CMDB Software,“
[WWW] <http://www.combodo.com/itop-193>. [Kasutatud 22 04 2016].
- [9] „CMDBuild, open source CMDB for IT asset management - ITIL,“
[WWW] <http://www.cmdbuild.org/en>. [Kasutatud 22 04 2016].
- [10] „CMDB & IT Documentation,“
[WWW] <http://www.i-doit.org/>. [Kasutatud 22 04 2016].
- [11] „GreenIT,“
[WWW] <http://www.greenit.net/whygreenit.html>. [Kasutatud 22 04 2016].
- [12] „VMware,“
[WWW] <https://www.vmware.com/virtualization/overview>. [Kasutatud 23 04 2016].
- [13] P. Venezia, „InfoWorld. How to choose the right Linux distro,“ 24 09 2014.
[WWW] <http://www.infoworld.com/article/2687088/linux/how-to-choose-a-linux-server-distribution.html>. [Kasutatud 23 04 2016].
- [14] „Compare Virtualization Software & Hypervisors,“
[WWW] <http://virtualization.softwareinsider.com/>. [Kasutatud 23 04 2016].
- [15] „What Matrix,“

- [WWW] <https://www.whatmatrix.com/comparison/Virtualization>. [Kasutatud 24 04 2016].
- [16] K. Ward, „Gartner Names Microsoft and VMware as Server Virtualization Leaders,“ 21 07 2015.
[WWW] <https://redmondmag.com/articles/2015/07/21/microsoft-and-vmware-as-server-virtualization-leaders.aspx>. [Kasutatud 24 04 2016].
- [17] VMware, „TOTAL COST COMPARISON: VMWARE VSPHERE VS MICROSOFT HYPER-V,“ 2012.
- [18] D. Papkin, „Microsoft 2012 R2 Hyper-V vs VMware Vsphere 6 - Clash of the Titans,“ 07 10 2015.
- [19] D. Stücken, „i-doit-api-clients / PHP,“
[WWW] <https://bitbucket.org/dstuecken/i-doit-api-clients>. [Kasutatud 25 01 2016].
- [20] „vSphere Perl SDK for vSphere 6.0,“
[WWW] <https://developercenter.vmware.com/web/sdk/60/vsphere-perl>.
[Kasutatud 22 04 2016].
- [21] „vSphere SDK for Perl Installation Guide. Overview of Linux Installation Process,“
[WWW] https://pubs.vmware.com/vsphere-50/index.jsp#com.vmware.perlsdk.install.doc_50/cli_install.3.3.html. [Kasutatud 04 02 2016].

Lisa 1 – Virtualiseerimistehnoloogiate võrdlemise kriteeriumid ja nende kirjeldused

1. HINNANG

1.1. Viimase relüüsi kuupäev – viimase versiooni avalikustamise kuupäev

1.2. Turupositsioon – sarnaste toodete hulgas positsioon

2. HIND

2.1. Virtualisatsiooni hind (dollarites) – virtualiseerimise litsentsi ja sellega seotud tugiteenuste (kui eksisteerivad) hind

2.2. Haldamiskulud (dollarites) – litsentsi haldamise ja sellega seotud tugiteenuste hind

2.3. Külalisoperatsioonisüsteemide litsentseerimine – kas operatsioonisüsteemide litsentsid on lisatud virtualisatsiooni litsentsi.

3. VÕIMALUSED

3.1. Tsentraliseeritud juhtimine – kas on olemas tsentraliseeritud multiserveri haldamine selle versiooni jaoks?

3.2. Virtuaalne ja füüsiline juhtimine – võime kasutada tarnija poolt pakutud juhtimisvahendeid, et hallata virtuaalset ja füüsilist taristut.

3.3. Integreeritud rollipõhine ligipääsu kontroll - läbi aktiivkataloogi sisselogimine

3.4. Veebipõhine haldamine – võimalus hallata läbi veebilehitseja

3.5. Detailne teenuste haldamine – hõlmab selliseid võimalusi, nagu jõudluse ja mälumahtude haldamine, virtuaalse keskkonna konfiguratsiooni / vastavuse kohta hoiatamine

4. UUENDUSED JA VARUNDAMINE

- 4.1. Hüperviisori uuendamine – sisseehitatud uuendused ja paranduse mehhanism hostide jaoks
- 4.2. VM-te hädaparandus – integreeritud hädaparanduse mehhanism VM-te jaoks (külalisoperatsioonisüsteemid)
- 4.3. VM hetkvõte – võime teha momentvõtet kuni töötab külalisoperatsioonisüsteem
- 4.4. Integreeritud taastamise rakendusliides– integreerimisvõimalused teiste taastamiskendustega
- 4.5. Integreeritud taastamine – kas on olemas pistikprogrammid VM-te taastamist vahenditeks

5. PAIGALDAMINE

- 5.1. Automatiseeritud hostide paigaldamine – võime automaatselt paigaldada virtualiseerimise hostid kasutades sisseehitatud automaatpaigalduse funktsionaalsust
- 5.2. VM mallid – võimalus luua ja hoida operatsioonisüsteemide kujutisi ja pärast nende käivitada masinaid
- 5.3. Hostide profiilid – võimalus võtta hosti konfiguratsiooni aspekte (turvalisus, võrk, kettad) põhimallina, et hiljem rakendada teistes hostides
- 5.4. Salvete profiilid – võimalus liigitada ressursse võimaluste järgi (näiteks RAID tasemed, käideldavus) selleks, et tagada proportsionaalset koormuste jaotamist

6. TEISED

- 6.1. Ressursside poolid – võime määrata füüsiliste ressursside prioriteete (nt. protsessorite ja mälu)
- 6.2. Füüsiline – Virtuaalne (FV) – integreeritud või lisatud FV ja VV (Virtuaalne-Virtuaalne) võimalused selleks, et teisendada füüsilisi masinaid virtuaalseteks masinateks

- 6.3. Iseteeninduse portaal – iseteenindusportaali ja/või teenuste kataloogi võimalused (tavaliselt seotud pilvekeskkonnaga)
 - 6.4. Töövood – võimalus automatiseerida käsitsi tehtavaid ülesandeid, kasutades töövoos mootoreid
 - 6.5. Turvalisus – sisseehitatud turvalisuse funktsionaalsus virtuaalse taristu jaoks, k.a. hüperviisori turvalisuse funktsioonid
 - 6.6. Süsteemide haldamine – hoiatamise ja monitooringu meetmed, käsuraapõhine haldamine (lokaalne ja kaughaldus)
7. VM MOBIILSUS
- 7.1. VM-te onlain migreerimine – võimalus migreerida VM-id ilma maasolekuta
 - 7.2. Migratsiooni ühilduvus – võimalus migreerida VM-id erinevate protsessorite generatsioonide vahel (varjates vastuolus olevaid piiranguid)
 - 7.3. Hooldusrežiim – võimalus panna hosti hooldusrežiimi nii, et see võimaldab viia kõik masinad teisele hostile ja siis panna esimene turvaliselt kinni
 - 7.4. Automaatne elu migratsioon – sisseehitatud võimalus automaatselt tasakaalustada töökoormust järgmiste komponentide vahel: Protsessor – P, Mälu – M, Võrk – V.
 - 7.5. Võimsuste haldamine – sisseehitatud toitehaldusfunktsioonid, mis võimaldavad migreerida VM-d ja lülitada välja kasutamata hoste ja lülitada tagasi sisse, kui on vaja.
 - 7.6. Andmete migreerimine – sisseehitatud võimalus onlain andmete migreerimiseks teisse asukohta
8. KÕRGE KÄIDELDAVUS JA TAASTAMINE
- 8.1. Klatri suurus - maksimaalne hostide arv klatri / poolis ja maksimaalne VM-te arv klatri / poolis

- 8.2. Sisseehitatud kõrge käideldavus (VM-a restart) – VM-te taastamine hostide tõrgete korral läbi taaskäivitamise alternatiivsetes hostides
- 8.3. Automaatne VM lähtestamine – automaatne üksikute virtuaalmasinate lähtestamine (host ei muutu), kui need ei vasta
- 8.4. VM *Lockstep* kaitse – võimalus kaitsta VM hosti rikke vastu ilma ühtegi maasolekuta. Nt. üleminek teisele VM-le.
- 8.5. Rakenduse / teenuse kõrge käideldavus – võime jälgida ja kontrollida rakendust VM-l ja taaskäivitada / parandada, kui probleem sai avastatud
- 8.6. Tiražeerimine - Andmebaasihalduses hajusandmebaaside hoidmine sünkroniseerituna sel viisil, et kogu andmebaasi tervikuna või selle osi kopeeritakse regulaarselt teistesse võrgus olevatesse serveritesse [2]

9. HOSTI SEADISTAMINE

- 9.1. Maksimaalne VM-te arv
- 9.2. Maksimaalne protsessori konfiguratsioon füüsilises hostis
- 9.3. Maksimaalne muutmälu protsessori kohta
- 9.4. Maksimaalne *RAM* hosti kohta
- 9.5. Maksimaalne virtuaalne *CPU* VM kohta

10. VM SEADISTAMINE

- 10.1. Maksimaalne *RAM* VM kohta
- 10.2. Järjestikpordid
- 10.3. USB toetus
- 10.4. Onlain komponentide lisamine – võimalus lisada VM komponente VM töötamise ajal

10.5. Dünaamiline / *Over-Commit* – võimalus esitada rohkem mälu VM-te jaoks, kui on füüsiliselt võimalik, mälu dünaamilise eraldamise kaudu, kui seda on vaja ja eemaldamine, kui seda pole vaja, selleks, et maksimaalselt suurendada konsolideerimise suhtarvu.

10.6. Mälu lehtede jagamine (Memory Page Sharing) – VM külalisoperatsioonisüsteemis *RAM*'is sarnaste lehtede otsingu mehhanism, kus sarnase lehe leidmise korral koopია asemel pannakse viide originaalsele mälu lehele.

11. KOOSTALITLUSVÕIME

11.1. *OVF (Open Virtualization Format)* toetamine – standart VM-te hoidmiseks ja jaotamiseks

11.2. Riistvara ühilduvus – viited tarnija sertifitseerimisele ja ühilduvuslehtedele

11.3. Külalisoperatsioonisüsteemide toetus – viited toetavatele süsteemidele

11.4. Skriptimine / API'd – nimekiri toetavatest liidestest

11.5. *CLOUD API* – liides pilveteenuste jaoks

12. SALVESTI

12.1. Toetatud mäluseadmed

12.2. Jagatud failisüsteem

12.3. Bootimine kettavõrgult

12.4. Buutimine USB pealt

12.5. Maksimaalne ketta maht – VM jaoks

13. VÕRK

13.1. *VLAN* - Loogiliselt sõltumatu kohtvõrk. Ühe füüsilise silla taga võib olla korraga mitu virtuaalkohtvõrku. [2]

13.2. *PVLAN* – *privaatne VLAN*

13.3. *IPv6*

Lisa 2 – VMware vCenter Server Appliance 6.0 installimine

Samm 5: Lokaalse domeeni loomine

VMware vCenter Server Appliance Deployment

- ✓ 1 End User License Agreement
- ✓ 2 Connect to target server
- ✓ 3 Set up virtual machine
- ✓ 4 Select deployment type
- 5 Set up Single Sign-on**
- 6 Select appliance size
- 7 Select datastore
- 8 Configure database
- 9 Network Settings
- 10 Ready to complete

Set up Single Sign-on (SSO)
Create or join a SSO domain. An SSO configuration cannot be changed after deployment.

Create a new SSO domain
 Join an SSO domain in an existing vCenter 6.0 platform services controller

vCenter SSO User name: administrator

vCenter SSO Password: [masked] ⓘ

Confirm password: [masked]

SSO Domain name: test.local ⓘ

SSO Site name: test ⓘ

⚠ Before proceeding, make sure that the vCenter Single Sign-On domain name used is different than your Active Directory domain name.

Seadistuste kokkuvõte:

VMware vCenter Server Appliance Deployment

- ✓ 1 End User License Agreement
- ✓ 2 Connect to target server
- ✓ 3 Set up virtual machine
- ✓ 4 Select deployment type
- ✓ 5 Set up Single Sign-on
- ✓ 6 Select appliance size
- ✓ 7 Select datastore
- ✓ 8 Configure database
- ✓ 9 Network Settings
- 10 Ready to complete**

Ready to complete
Please review your settings before starting the installation.

ESXi server info: 192.168.1.206
Name: test-appliance
Installation type: Install
Deployment type: Embedded Platform Services Controller
Deployment configuration: Tiny (up to 10 hosts, 100 VMs)
Datastore: datastore1
Disk mode: thin
Network mapping: Network 1 to VM Network
IP allocation: IPv4 , DHCP
Time synchronization: Synchronize appliance time with ESXi host
Database: embedded
Properties: SSH enabled = true
SSO User name = administrator
SSO Domain name = vsphere.local
SSO Site name = test

Back Next Finish Cancel

Lisa 3 – Testid Seleniumis

FNT01

Sisselogimislehel sisestatakse õiged sisselogimisandmed, ehk kasutajanimi: admin ja parool: admin, kui tuleb lehele sõna „Welcome“, mis kuvatakse ainult õnnestunud sisselogimisel, siis test sai läbitud (vt Tabel 13).

Tabel 13. Funktsionaalse nõue test FNT01.

FNT01		
open	/i-doit/	
type	id=login_username	admin
type	id=login_password	admin
clickAndWait	id=btnLogin	
assertTextPresent		Welcome

FNT02

Sisselogimislehel sisestatakse valed sisselogimisandmed, ehk kasutajanimi: admin ja parool: test, kui tuleb lehele plokk *id=loginMessages*, siis see tähendab, et sisestati valed andmed, kui see plokk tuleb, siis test sai läbitud (vt Tabel 14).

Tabel 14. Funktsionaalse nõue test FNT012.

FNT02		
open	/i-doit/	
type	id=login_username	admin
click	id=loginArea	
click	id=loginArea	
type	id=login_password	test
click	id=btnLogin	
waitForElementPresent	id=loginMessages	True

FNT03

Eelnevalt oli KAB tühi. Pärast käivitati skript *vm_import.php* pärast skripti käivitamist pidid tulema andmebaasi automaatselt kirjed kandjal olevate VM-te kohta. Üheks VM nimeks on „test-appliance“. Antud test otsib, kas KAB-is on tekkinud kirje VM kohta

antud nimega. Test peetakse sooritatuks, kui andmebaasi tekib kirje üllal nimetatud VM kohta. Test sai sooritatud, kuna andmed VM kohta automaatselt tulid KAB-i (vt Tabel 15).

Tabel 15. Funktsionaalse nõue test FNT013.

FNT03		
open	/i-doit/?viewMode=1001&objTypeID=3	
click	link=Infrastructure	
click	id=nodeVisibilityIcon	
click	id=smenu_tree36	
assertText	css=td.data-grid-mainList-column-Objectlink	test-appliance

FNT04

Eelnevalt oli muudetud ühe VM RAM'i suurus. Enne see oli 8GB, nüüd sai 4GB, ehk 4096.00 MB. Antud test kontrollib, kas pärast vm_import.php käivitamist sai VM RAM'i suuruseks 4GB. Test peetakse sooritatuks, kui vastavas veerus on väärtus 4096.00 MB. Antud test oli edukalt läbitud, kuna andmed said uuendatud (vt Tabel 16).

Tabel 16. Funktsionaalse nõue test FNT04.

FNT04		
open	/i-doit/?viewMode=1001&objTypeID=59	
click	link=Infrastructure	
click	id=nodeVisibilityIcon	
click	css=#smenu_tree36 > span	
click	css=#data-grid-mainList-1 > td.data-grid-mainList-column-ID	
click	id=smenu_tree26	
assertText	xpath=(//td[@onclick="document.location.href='index.php?objID=38&tvMode=1006&viewMode=1100&objTypeID=59&cateID=5&catgID=5';"])[3]	4096.00 MB