

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Tarvo Kesküla 233698IAAB

IT organisatsiooni teenusemudeli ja CMDB analüüs

Bakalaureusetöö

Juhendajad: Siim Vene

MSc

Pirgit Kahro

MSc

Tallinn 2025

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: Tarvo Kesküla

03.01.2025

Annotatsioon

Käesolev bakalaureusetöö käsitleb konfiguratsioonihalduse andmebaasi (edaspidi CMDB) loomise ja rakendamise strateegiaid kaasaegses IT-organisatsioonis, keskendudes lahendustele, mis toetavad nii igapäevast IT-halduse operatiivsust kui ka pikaajalisi strateegilisi vajadusi. Töö eesmärk on analüüsida organisatsiooni teenusemudelit, selgitada välja selle kitsaskohad ning kavandada CMDB andmemudel ja sisend pilootprojekti teostamiseks.

Autor teostab teenusmodeli ja andmestiku analüüsi, et saada kinnitust tuvastatud probleemidele. Samuti uurib IT-komponentide ja nende seoste kirjeldamise võimalusi, arvestades asutuse spetsiifilisi piiranguid, haldusvastutust ja operatsioonimudelit. Praktilises osas kaardistab Zabbixi monitooringusüsteemi komponendid ning pakub näidismudeli CMDB rakendamiseks.

Töös on autor toonud oma soovitusel Atlassiani Assets mooduli pilootprojekti rakendamiseks, keskendudes CI-tüüpide määratlustele, soovituslikele atribuutidele ja seoste tüüpidele. Samuti esitab juhised teenusmodeli struktuuri parendamiseks, et tagada paremad seosed tehniliste ja äriteenuste teenustingimuste vahel. Autori eesmärk on juhistega toetada CMDB integreerimist igapäevasesse IT-halduse protsessidesse. Töö tulemusena valminud teoreetiline CMDB mudel on kooskõlas organisatsiooni teenusemudeligaga, modulaarne ja paindlik, võimaldades seda hõlpsalt laiendada vastavalt muutuvatele vajadustele.

Töö tulemus pakub autori hinnangul väärtuslikke juhiseid IT-halduse ja konfiguratsioonihalduse spetsialistidele, kes soovivad oma ettevõttes CMDB-d juurutada või arendada. See parandab IT-komponentide haldust, vähendades turvariske ning tõstes teenusekvaliteeti.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 34 leheküljel, 4 peatükki ja 3 joonist ning 3 tabelit.

Abstract

„Analysis of the Service Model and CMDB of an IT Organization“

This bachelor's thesis addresses the strategies for creating and implementing a Configuration Management Database (CMDB) in a modern IT organization, focusing on solutions that support both daily IT management operations and long-term strategic needs. The aim of the thesis is to analyze the organization's service model, identify its shortcomings, design a CMDB model along with input for implementing a pilot project.

The author conducts an analysis of the service model and data to confirm the identified issues. Additionally, possibilities for describing IT components and their relationships are explored, considering the organization's specific limitations, management responsibilities, and operational model. In the practical part, author maps the components of the Zabbix monitoring and proposes a sample model for CMDB implementation.

The thesis includes the author's recommendations for implementing a pilot project using the Atlassian Assets module, with a focus on defining CI types, recommended attributes, and relationship types. It also provides guidelines for improving the service model structure to establish better connections between technical and business service terms. The author's goal is to support the integration of CMDB into daily IT management processes through these guidelines. The resulting theoretical CMDB model aligns with the organization's service model, is modular and flexible, and can be easily expanded according to changing needs.

According to the author, the results of this thesis offer valuable guidance for IT and configuration management specialists aiming to implement or develop a CMDB in their organizations. This improves IT component management, reduces security risks, and enhances service quality.

The thesis is in Estonian and contains 34 pages of text, 4 chapters, 3 figures, 3 tables.

Lühendite ja mõistete sõnastik

API	(Application Programming Interface) Tarkvaraliides, mis võimaldab erinevatel rakendustel või süsteemidel omavahel suhelda ja andmeid vahetada.
Chef	Chef on avatud lähtekoodiga konfiguratsiooni- ja infrastruktuurihalduse tööriist, mis võimaldab automatiseerida IT-ressursside juurutamist, haldamist ja haldusprotsesse.
CI	Konfiguratsiooniüksus (<i>Configuration Item</i>). CI on IT-infrastruktuuri või teenuste komponent, mida saab hallata ja jälgida konfiguratsioonihalduse raames.
CMDB	Konfiguratsioonihalduse andmebaas (<i>Configuration Management Database</i>). CMDB on andmebaas, kuhu salvestatakse IT-süsteemi ja teenuste konfiguratsioonandmed.
ISKE	Infosüsteemide Kolmeastmeline Etalonturbe süsteem on Eesti riiklik standard ja raamistik.
ITSM	ITSM (<i>IT Service Management</i>) platvorm on tarkvara või süsteem, mis aitab ettevõtetel efektiivselt hallata ja optimeerida oma IT-teenuseid ja IT-komponente.
KVM	KVM (<i>Kernel-based Virtual Machine</i>) on Linuxi tuuma integreeritud virtualiseerimise tehnoloogia, mis võimaldab füüsilise masina ressursse jagada ja hallata mitme virtuaalmasina vahel.
MVP	MVP (<i>Minimum Viable Product</i>) on minimaalsete funktsioonidega toode, mis lahendab põhivajaduse ja võimaldab koguda kasutajate tagasisidet edasisteks täiustusteks.
OLA	OLA (<i>Operational Level Agreement</i>) on kokkulepe organisatsiooni siseste osakondade või meeskondade vahel, mis määratleb, kuidas nad koos töötavad, et toetada teenuse taseme kokkuleppes sätestatud nõudeid. OLA keskendub tehniliste ja operatiivsete protsesside koordineerimisele teenuste pakkumisel.
Pod	Pod on Kubernetese väikseim ja lihtsaim juurutamise ja haldamise üksus, mis sisaldab ühte või mitut kontainerit.
RPO	RPO (<i>Recovery Point Objective</i>) ehk taastamispunkti eesmärk määratleb maksimaalse ajavahemiku, mille jooksul andmed võivad kaotsi minna süsteemi rikke või katastroofi korral.

RTO	RTO (<i>Recovery Time Objective</i>) ehk taastamise aja eesmärk määratleb maksimaalse aja, mille jooksul süsteem või teenus peab olema pärast katkestust taastatud.
SLA	SLA (<i>Service Level Agreement</i>) on teenusepakkuja ja kliendi vaheline kokkulepe, mis määratleb pakutava teenuse ulatuse, kvaliteedi ja mõõdikud.
VMware SRM	VMware SRM on katastroofijärgse taaste (<i>disaster recovery</i>) ja ärikontinentsuse tagamise lahendus, mis võimaldab automatiseerida ja hallata virtuaalmasinate taasteprotsesse erinevate andmekeskuste vahel.

Sisukord

1 Sissejuhatus	11
1.1 Teema ajakohasus.....	13
1.2 Ülesande püstitus ja metoodika	14
1.3 Kitsendused ülesandele.....	15
2 Olemasoleva olukorra analüüs.....	16
2.1 Teenusemudel.....	17
2.1.1 IKT teenus	19
2.1.2 Tehniline teenus.....	22
2.1.3 Funktsioon	23
2.1.4 CI-d (Chefist).....	24
2.2 Zabbix süsteemi kaardistus.....	25
2.3 Atlassiani IT Assets mooduli praktikad.....	30
2.4 CMDB loomise raamistik.....	32
3 Muudatuste ettepanekud ja pilootprojekti sisend	33
3.1 Äriteenus.....	33
3.2 Tehniline teenus.....	34
3.3 Tehniline teenuse teenuspakkumus	35
3.4 Meeskond.....	36
3.5 CMDB ja CI-de loomine	36
3.6 Objektiskeemid.....	37
3.7 CI atribuudid.....	38
3.8 Virtuaalmasina CI.....	39
3.9 Operatsioonisüsteemi CI.....	39
3.10 Klaster CI.....	40
3.11 Seoste tüübid.....	40
3.12 CMDB haldusrollid ja dokumenteerimine	41
4 Kokkuvõte	43
Kasutatud kirjandus	45

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	46
Lisa 2 – Teenusmudeli näidis joonis	47

Jooniste loetelu

Joonis 1. Teenuse mudel/struktuur [4]	17
Joonis 2. Zabbix monitooringu komponendid.....	26
Joonis 3. Teenusmudeli näidis.....	47

Tabelite loetelu

Tabel 1 Virtuaalmasina CI atribuutide loend	39
Tabel 2 Operatsioonisüsteemi CI atribuutide loend	39
Tabel 3 Klaster CI atribuutide loend	40

1 Sissejuhatus

Iga organisatsiooni jaoks on oluline omada ülevaadet oma ressurssidest. Arvestades tänast ettevõtluskeskkonda koosneb üks osa ressursse IT-komponentidest. Üksikult vaadates on võimalik arvestust nende üle pidada nii varaarvestuse programmides kui ka väiksema ettevõtte puhul Excelis. Mida keerukamaks muutub IT-infrastruktuur, seda olulisem on jälgida ja hallata oma IT-komponente nii, et see ei oleks staatiline informatsioon.

Lisandväärtust annab, kui ettevõttel on ülevaade, millist rolli omavad need IT-komponendid äriteenuste pakkumisel ning kuidas need on omavahel seotud. Hea ülevaade oma komponentidest ja protsessidest toetab ettevõtete ümberkujundamist ja digitaalseid äriprotsesse.

IT-komponentide haldamiseks kasutatakse konfiguratsioonihalduse andmebaasi (edaspidi CMDB). Selle edukaks loomiseks peab mõistma, et IT varade haldus või IT komponentide register eraldiseisvalt ei paku piisavalt väärtust, CMDB peab olema korelatsioonis teenusmudeliga, protsessidega, haldusvastutus- ja operatsioonimudeliga. Nii nagu iga suure projekti puhul, tuleb ka edukaks konfiguratsioonihalduse loomiseks lähtuda põhimõttest: „Alusta väikestest sammudest ja laienda järk-järgult. Loo kindel CI-de register, mis on seotud konkreetsete kasutusjuhtumitega. Kui avastad, et täidad oma CMDB-d elementidega, mis ei seostu ettevõtte prioriteetide või kasutusjuhtumitega, oled tõenäoliselt õigelt teelt kõrvale kaldunud.“ [1] - Tea, kus sa praegust asud, kuhu jõuda tahad ja milliste (väikeste) sammudega sa sinna jõuad.

Efektne IT-komponentide haldamine sh eduka CMDB loomine aitab tagada ka ettevõtte küberturvalisust. Nagu sõnas Riigi Infosüsteemi Ameti (RIA) peadirektor Margus Noormaa [2], on küberturvalisus digiriigi vundament. Seega ka aina enam ettevõtteid digitaliseerides, tuleb arvestada turvalisuse aspektidega.

Ülevaade haldusvastutusest ja eskalatsiooni ahelast, see aitab kiiremini ja paremini reageerida veasituatsioonidele. IT-komponentide puudulik haldamine, nende sidemete ja

teenuste vaheliste seoste jälgimise keerukus võib kaasa tuua riske, nagu vigane muudatuste juhtimine, turvariskide alahindamine ja teenusekvaliteedi langus.

Tulevikku vaadates on heal konfiguratsioonihaldusel oluline ülesanne nii äritegevuses, kui ka turvalisuse tagamisel. Kiiresti muutuv keskkonnas on hea CMDB-lahendus, mida saab edasi arendada, kriitilise tähtsusega. See peab toetama ettevõtte vajadusi ja arenema koos ettevõttega.

1.1 Teema ajakohasus

Antud lõputöö baseerub asutusel, mille peamine tegevusala on IT-teenuste pakkumine. Asutus paneb suurt rõhku teenuste arendamisele ja küberturvalisusele. Seega on asutuses olemas teenuste portfelli ja kõik äriteenused on kirjeldatud. Lõputöö raames vaadeldakse äriteenuste pakkumise jaoks vajalikke IT-komponentide haldamise parendamise võimalusi.

Hetkel puudub asutuses ühtne register IT-komponentidest ja neis kasutusel olevatest baastarkvara versioonidest, mis võimaldaks luua efektiivselt raporteid ning saada terviklik ülevaade tarkvara ja riistvara elutsükli juhtimisest. Samuti puudub selge ülevaade IT-komponentide ja teenuste nõuete vahelistest seostest, mis on oluline monitooringualarmide koondamiseks, intsidentide efektiivselt koordineerimiseks, muudatuste planeerimiseks ja finantskulumudelite arvutamiseks.

Puudub keskne andmebaas IT-komponentide asukohtade ning võrgusõltuvuste kaardistamiseks ja visualiseerimiseks, mis raskendab infrastruktuuri haldamist ja sõltuvuste analüüsi. Lisaks pole IT-teenuseid sidustatud teenuspakkumiste ja sisemiste *Operational Level Agreement* (edaspidi OLA) kokkulepetega, mis on vajalikud äriteenuste haldamise ja teenuse tasemete kvaliteedi jälgimiseks. On olemas igal äriteenusel selle pakkumise kokkulepped.

Haldusvastutus ja eskalatsiooniahelate kirjeldused on hetkel manuaalselt kirjeldatud äriteenuste juures ja mitte masinloetavalt integreeritavalt haldusvahenditega. Muudatuste mõju hindamine on keeruline ja aeganõudev, samuti on raskendatud teenusekvaliteedi juhtimine.

1.2 Ülesande püstitus ja metoodika

Käesoleva töö eesmärk on kavandada ja uurida konfiguratsioonihalduse andmebaasi (edaspidi CMDB) loomise ja rakendamise mudelit, mis vastaks kaasaegse IT-organisatsiooni vajadustele. Selgitada välja vajalikud muudatused organisatsiooni teenusemudelis ning luua CMDB andmemudel, mis toetab nii igapäevast IT-halduse operatiivsust kui ka pikaajalisi strateegilisi vajadusi IT-teenuste juhtimisel. Lahenduse analüüsimisel keskendutakse iteratiivsele lähenemisele alustades minimaalselt vajalikest komponentidest. Mudelite loomisel arvestab autor, et CMDB-d on võimalik järk-järgult laiendada vastavalt asutuse vajadustele

Peatükis 1.1 Teema ajakohasus on toodud puudused, mis vajavad töö autori hinnangul parendamist. Antud töö raames teostatakse analüüs, et hinnata teenusmudeli ja andmestiku vastavust tuvastatud probleemidele. Lisaks kaardistab autor IT-komponentide ja nende vaheliste seoste kirjeldamise võimalused, võttes arvesse antud asutuse piiranguid, teenusmudelit, haldusvastutust ja organisatsiooni operatsioonimudelit. Samuti viib töö autor läbi analüüsi selgitamiseks võimalusi CMDB sidumiseks IT-teenuste ja nende pakkumistega, mis tagaks selge ülevaade IT-komponentide rollist äriteenuste pakkumisel. Analüüsi käigus teostatakse järgnevad ülesanded:

- IT-komponentide (nt riistvara, tarkvara, asukohad, virtuaalmasinad) ja nende vaheliste seoste kirjeldamiseks vajaliku CI andmemudeli loomine.
- Haldusvastutuse ja eskalatsiooniahelate kirjeldamise analüüs.
- Teenusmudeli ja tehniliste teenuste sidumine CMDB-ga, et võimaldada OLA-de ja SLA-de rakendamist ja jälgimist.
- Parimate praktikate uurimine Atlassian Asset mooduli kasutamisel ja üldiste CMDB ehitamise raamistike analüüs.

Bakalaureuse töö metoodika baseerub juhtumianalüüsil (*case study*) ehk vaadeldakse ühe asutuse näitel tekkinud probleeme ja nende lahendamise viise. Samuti kasutab töö autor erialast kirjandust, et luua mudelid, mis põhinevad tõestatud metoodikatel.

1.3 Kitsendused ülesandele

CMDB loomisel ja kavandamisel lähtub töö autor järgnevatest kitsendustest:

- Töös käsitletavas asutuses on kasutusel Atlassian Service Desk ja IT Asset moodulid, mis määravad tööriistad CMDB loomise ja andmemudeli realiseerimiseks. Teiste CMDB platvormide analüüs ja võrdlus ei kuulu töö skoopi.
- Antud töö raames lähtutakse pilootprojekti piiratusest - Pilootprojekti raames keskendutakse ühe konkreetse süsteemi (antud töö raames on selleks valitud Zabbix monitooringusüsteem) kaardistamisele ja selle seoste analüüsimisele. Eesmärgiks on luua minimaalne toimiv prototüüp (edaspidi MVP), mis võimaldab testida CMDB andmemudelit ja integreerimist. Selline lähenemine aitab tuvastada kiirelt võimalikud kitsaskohad andmemudeli ja protsesside rakendamises. Pilootprojekti käigus piiratakse käsitletavate CI-de ja atribuutide hulk miinimumini, et tagada lihtne ja tõhus õppimisprotsess edasiste iteratsioonide jaoks.
- Autori ligipääsu piirangud - autoril puudub tehniline ligipääs süsteemidele, mistõttu piirdub töö teoreetilise mudeli ja andmemudeli kavandamisega. Pilootprojektiga saab töö autor tuua välja lahenduse realiseeritavuse. Reaalne teostust (sh andmete sisestamine või integratsioonide seadistamine) ei ole antud töö skoopis. Bakalaureusetöö tulemuseks on teoreetiline lahenduse ettepanek, mis loob aluse praktilisele rakendamisele organisatsioonis. Organisatsiooni juhtkonna poolt on olemas valmisolek antud töö tulemuste rakendamisega edasi minna.

2 Olemasoleva olukorra analüüs

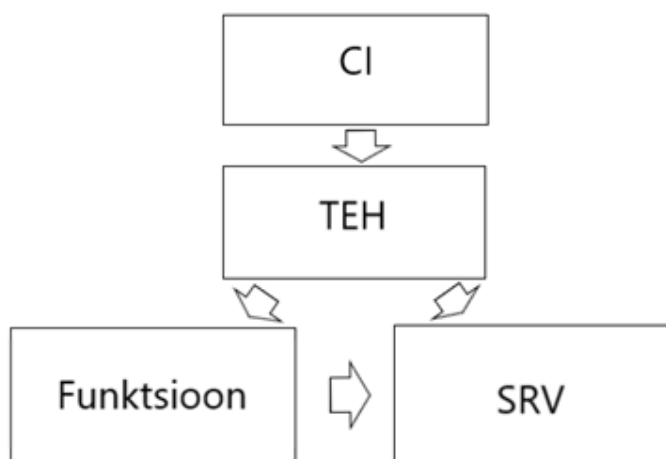
„IT-varahalduse ja teenuste konfiguratsioonihalduse rollid on mõlemad olulised suurepärase IT-teenuste halduse (ITSM) tagamiseks“ [3]. Sellise lausega algab sissejuhatus Atlassiani IT-varade ja teenuste konfiguratsioonihalduse käsiraamatus. See tõendab veelkord, et CMDB-d ei ole mõtet käsitleda kui lihtsalt eraldi IT komponentide registrit. Väärtusloome mida selline piiratud skoobiga süsteem pakub, on vähene. Tõeline kasu väärtusahelas avaneb alles siis kui IT-varade register on seotud teenusmodeliga, haldusrutiinide ja protsessidega, vastutusmodeliga. Samuti tuleb see integreerida tugisüsteemidega nagu näiteks haldusserverid, monitooring, varundus ja kliendihaldussüsteem.

Paljud konfiguratsiooni komponentide atribuudid ja andmemodeli disain on suures sõltuvuses sellest milline on organisatsiooni struktuur, opereerimise mudel, haldusvastutus ja tööprotsessid. Sellest tulenevalt on autor otsustanud esimesena analüüsida teenuse mudelit ja sellest tulenevaid nõudeid konfiguratsioonihaldusele. Vastavalt analüüsi tulemustele koostab autor muudatuste vajaduste kirjeldused teenusemodelile, et toetada püstitatud eesmärke.

Asutuses on alustatud konfiguratsioonihalduse protsessi juurutamist 10 aastat tagasi. Tänapäevaks on kogu keskkond muutunud. Aastaid tagasi loodud protsessi ei ole hoitud ajakohasena, seega peab mitmeid protsesse alustama uuesti.

2.1 Teenusemudel

Ettevõttes ei ole täna kõikide teenuste osas kasutusel ühtselt mõistetav teenusemudel. Töös on võetud aluseks teenuse mudeli, mis on kirjeldatud asutuse sisemises dokumendis „Teenuste portfelli ja IKT haldusprotsessid – lühi tutvustus“ [4] ja toodud joonisel 1.



Joonis 1. Teenuse mudel/struktuur [4]

Dokument defineerib joonisel 1 olevad objektid järgmiselt:

- SRV – Teenus, mida asutus pakub partnerile või kasutab ise teiste teenuste osutamiseks.
- FREG – Funktsioon, mille näol on tegemist mingi abstraktse kehaga, mis hõlmab endasse asutuse poolt loodud rakendusi/tehnilisi lahendusi, mille abil partner saab sooritada omatöös vajalikke tegevusi.
- TEH – Tehniline teenus on mingi rakenduse paigaldus või loogiline paigalduste kogum üle keskkondade.
- CI – server/andmebaas

Asutuses on kõik äritaseme teenused kirjeldatud SRV teenustena. Samas asutuse sisestest teenustest on nii kirjeldatud vaid umbes pooled. SRV teenuskaardile on asutus seadnud nii oma sisemiste kordadega, kui välistelt osapooltelt tulenevaid nõudeid. Täpsemalt on kirjeldatud neid nõuded järgmises peatükis.

Funktsioonide register (FREG) on asutuses täna vaid mõne näitega sisustatud. Antud seoste haldamine on oluline tagamaks taaskasutuse võimekuse ja parema reageerimise intsidentide lahendamisel. Täpsemalt on kirjeldatud funktsioonide osa peatükis 2.1.3.

Tehnilised teenused on asutuse vaates kõige laialdasemalt kirjeldatud, kuid kuna puuduvad kokkulepitud ühtsed alused, ei ole kirjeldused kõikidel samaväärsed. Tehnilistest teenustest on autor pikemalt kirjutanud peatükis 2.1.2.

CI objekti on kirjeldatud täpsemalt peatükis 2.1.4.

2.1.1 IKT teenus

Asutuses on kasutusel IKT teenuse osas definitsioon „IKT teenus on kliendile väärtust pakkuv IKT tehnoloogiat kasutav ning IKT teenuse osutaja poolt IKT ressurside baasil pakutav teenus, mis toetab asutuse poolt pakutavaid teenuseid.“ [5]. Samuti on kehtestatud IKT teenuste üldtingimused ja nõuded IKT teenuste haldamiseks ja kirjeldamiseks, millest autor toob välja järgnevad:

1. Teenusega seotud minimaalse vajaliku informatsiooni hulga otsustab portfelli juht ja konfigureerib info haldamiseks vajaliku keskkonna.
2. Teenuste portfellis on IKT teenus seotud nii avaliku teenusega kui ka infosüsteemiga. Samuti kajastab portfelli teenuse seoseid teiste IKT teenuste ja kolmandate osapoolte teenustega.
3. Teenused liigitatakse:
 - 3.1. Äriteenused – teenused mida osutatakse Siseministeeriumi valitsemisala asutustele;
 - 3.2. Siseteenused – teenused, mis on vajalikud äriteenuste osutamiseks või asutuse protsesside haldamiseks;
 - 3.3. Partnerlusteenused – kolmanda osapoolte teenused, kus asutuse on vahendajaks;
 - 3.4. Välisteenused – teenused mida osutatakse asutustele, mis ei kuulu Siseministeeriumi valitsemisalasse;
 - 3.5. System – teenuse liik, mis on vajalik spetsiifiliste teenuse portfelli lisamiseks.
4. Aktiivsel IKT teenustel on olemas alljärgnev informatsioon:
 - 4.1. intsidentide ja pöördumiste lahendamise seotud kriitiline informatsioon kasutajatoele;
 - 4.2. teenuse meeskonna ja asutuse peakasutaja kontaktandmed;
 - 4.3. arendus- ja halduspartneri kontaktandmed;
 - 4.4. kirjeldus teenuse osutamise tingimustest;

- 4.5. teenuse arhitektuuri joonis või viide selle asukohale muus süsteemis;
- 4.6. kontrollküsimustik kasutajatoele teenuse intsidendi ulatuse kiireks määramiseks;
- 4.7. teenuse arendamisega seotud dokumentatsioon või viide selle asukohale muus süsteemis;
- 4.8. teenuse monitooringukaart ja reageerimisjuhend;
- 4.9. viited teenusega seotud intsidentidele, probleemidele ja kvaliteedimõõdikutele.
- 4.10. teenusega seotud standardtellimuste täitmise kirjeldused [6]

Autori hinnangul on asutuses kõigile kättesaadava IKT-teenuse dokumentatsioon üldiselt pealiskaudne. Enamik vajalikke teadmisi teenuse kohta on kättesaadavad teenuse realiseerimisest Atlassiani platvormil. Samuti on võimalik leida teenuste kohta detailsemat infot arhitektuurijoonistelt *Enterprise Architect* (edaspidi EA) keskkonnas. Mõlemal juhul on info leidmine ajakulukas.

Teenuste register ja kirjeldamine toimub asutuse Jira platvormil Teenuse Portfelli projektis. Teenuste portfellis teenuse loomise kirjelduse ja sellele kohalduvate nõuete analüüsi põhjal saab autor järeldada, et portfellis teenuse kirjeldamise eesmärk on peamiselt teenuse nõuete, seotud kontaktide, dokumentatsiooni algpunktide ning eskalatsiooniahela juhendi pakkumine esimese liini kasutajatoe jaoks. Autori hinnangul keskendutakse eelkõige teenusega seotud baasinformatsiooni koondamisele, mis peab lihtsustama ja kiirendama probleemide lahendamist ja haldust.

Teenusega seotud informatsiooni kirjeldamiseks on üle 100 erineva välja (atribuudi), millest paljud on vaba teksti väljad. Autor ei pea vajalikuks kogu atribuutide nimekirja eraldi välja tuua ja üks haaval neid analüüsida, kuid esmasel ülevaatusel on autori arvates teenuse objekti üritatud liiga palju infot koondada, osa sellest infost sinna ei kuulu ja ei kohaldu kõikidele teenuse alustele tehnilistele teenustele või tehnilistele komponentidele.

Uue teenuse loomise juhend sätestab vaid minimaalse komplekti vajalikke sisestatavaid andmeid. Teenuse atribuutide loendi loomise eest vastutab tooteomanik või muu meeskonna liige. Vajaliku info olemasolu ja selle õige käsitlemine on hetkel

tootemeeskonna vastutusel. See aga ei taga ühtsetel alustel kõikide vajalike atribuutide täitmist.

Oluline osa teenuse kirjeldamisel on teenuse teenustaseme kokkulepe *Service Level Agreement* (edaspidi SLA), mille käigus kirjeldatakse teenuse nõuded käideldavusele.

Meie analüüsitava organisatsioonis on IKT teenuste käideldavuse kirjeldamiseks vastavalt ISKE-le jäänud neli käideldavusklassi [7]:

K3 – suurem või võrdne 99,9 % aastas ja maksimaalne lubatud ühekordsete katkestuste summaarne pikkus nädalas teenuse aktiivsel kasutusajal kuni 10 minutit;

K2 – suurem või võrdne 99% aastas ning maksimaalne lubatud ühekordsete katkestuste summaarne pikkus nädalas teenuse aktiivsel kasutusajal kuni 2 tundi;

K1 – suurem või võrdne 90% aastas ning maksimaalne lubatud ühekordsete katkestuste summaarne pikkus on kuni 24 tundi nädalas;

K0 – töökindlus pole oluline ja jõudlus pole oluline.

Autori arvates on selline määratlus teenustaseme kirjeldamiseks piisav, aga kohati on teenus ise liiga laia skoobiga või sisaldab rohkem kui ühte keskkonda või süsteemi, näiteks „Dokumendihaldus Delta“ teenuse objektis kirjeldatakse ainult tootmiskeskonna käideldavuse nõuded, kuid sama teenuse alla loodud ka muud süsteemid ja keskkonnad. Kõikidele teenustele peab autori hinnangul eraldi kirjeldama arendustegevusteks, testimiseks või koolitusteks loodud keskkondade nõuded. Kõikidele keskkondadele ei kohandu samad käideldavuse nõuded kui tootmiskeskondadele.

Autori analüüsi tulemusel võib tuua välja, et üldiselt on teiste nimetatud keskkondade nõuded kas kirjeldamata või suusõnaliste kokkulepetena kehtivad. See omakorda tähendab, et ka IT halduse süsteemides on selle kohased seadistused manuaalsed ja kohati sõltuvad töö tegijast, näiteks monitooringu alarmide aktiveerumise ajad. Autori arvates kirjeldades ära ka teised keskkonnad eraldi teenusena, on võimalik seotud IT haldamise süsteemidega masinloetavalt käideldavuse tingimusi ja reaktsiooniaegu siduda komponentidega.

2.1.2 Tehniline teenus

Tehniliste teenuste dokumentatsioonis [8] on defineeritud tehniline teenus järgmiselt: „Tehniline teenus on mingi rakenduse paigaldus või loogiline paigalduste kogum üle keskkondade. Tehniline teenus võib olla äriteenuse/siseteenuse omanduses, aga ei pea olema. Ühes serveris olevaid mitut rakendust võib lugeda üheks tehniliseks teenuseks, "komplektiks". Tehniline teenus pakub mingit funktsiooni. Lihtsustatud öeldes on tehniliste teenuste register lihtsalt paigalduste register, mis on üle keskkondade. Ehk kui samast komponendist on mitu erineva andmekoosseisuga paigaldust, siis on mitu erinevat tehnilist teenust. Aga kui on näiteks samade andmete pealt tehtud ajutine paigaldus vea tuvastamiseks, siis on sama tehniline teenus“ [8].

Tehniliste teenuste andmete analüüsimisel on autor tuvastanud, et tehniliste teenuste ulatus ja tüübid varieeruvad ning puudub järjepidevus üle erinevate üksuste ja platvormide. Näiteks on osa tehnilisi teenuseid kirjeldatud kui platvormi- või tehnoloogiaspetsiifilised teenused, kuhu on koondatud kõik vastava tehnoloogia komponendid. Teised seevastu on defineeritud asukohapõhiste või süsteemispetsiifiliste lahendustena. Näitena võib tuua "TEH-736 Valga kontorivõrk", mis hõlmab mitmeid erinevaid tehnoloogilisi komponente, või "TEH-10 Monitooring" või „TEH-40 MariaDB“, mis kirjeldab terviklikku lahendust koos paljude erinevate komponentidega.

Autori hinnangul muudab selline varieeruvus tehniliste teenuste klassifitseerimise ja haldamise keeruliseks, kuna puudub selge järjepidevus ning eristatavus haldusvastutuse alusel. Lisaks ei võimalda antud lähenemine määrata selgeid *Operational Level Agreements-e* (OLA-sid) üksikute komponentide lõikes. See raskendab vastutuse ja teenusnõuete täpset määratlemist ning täitmist. Samuti ei ole võimalik tagada äriteenusele kehtestatud SLA-d, kui puuduvad alusteenuste kokkulepped.

2.1.3 Funktsioon

Teenuste funktsiooni dokumentatsioonis on funktsioon kirjeldatud järgnevalt: „Funktsiooniks on oma olemuselt **tegevus või tegevuste kogum** (nt millegi loomine/muutmine/täiendamine/asendamine/kasutamine vms). Teiste sõnadega on funktsiooni näol tegemist mingi abstraktse kehaga, mis hõlmab endasse asutuse poolt loodud rakendusi/tehnilisi lahendusi, mille abil klient/partner saab sooritada **oma töös vajalikke tegevusi**. Äriteenus võib koosneda kas ühest või mitmest funktsioonist. Oluline on mainida ka seda, et funktsiooniks võib olla ka mitte tehniline teenus - näiteks kasutajatugi või konsultatsioon“ [9].

Autori hinnangul on väga oluline kirjeldada funktsioone kui eraldi objekte. Näiteks on see oluline alus teenuste monitooringu kirjeldamisel. Äriteenustel on palju erinevaid funktsionaalsusi, mida kasutajad vajavad või kasutavad. Defineerides igale teenusele kriitilised funktsionaalsused, on võimalik funktsioonide monitooringu staatuste alusel selgelt tuvastada, kas intsident aluskomponentides on põhjustanud teenuse häirituse või katkestuse. Kasutades sama loogikat saab funktsiooni kasutada tehniliste teenuste puhul. Näiteks, kui mõnel tehnilisel teenusel on mitu erinevat funktsiooni seda kasutavate süsteemide või teenuste jaoks ja nende funktsioonide häirete puhul on mõju teistele süsteemidele või teenustele erinev, siis nende eraldi kirjeldamine võimaldab paremat mõjuanalüüsi ja monitooringu arhitektuuri planeerimist.

Funktsiooni objekt on autori uuritavas organisatsiooni suhteliselt uus funktsionaalsus. Antud töö koostamise ajal on seda hakatud realiseerima ja andmetega täitma alles ühe äriteenuse puhul pilootprojektina, seega sügavamat analüüsi selle objekti kohta autor käesolevas töös teha ei saa.

2.1.4 CI-d (Chefist)

Hetkel on CMDB rakendatud Atlassiani Service Deski projektina nimega CMDB (CI). CI-d on realiseeritud Service Deski piletitena, millel on kolm alamtüüpi: „Virtuaalne andmebaas“, „Server“ ja „Server Chefist“. Iga tüübi jaoks on määratletud tüübi spetsiifilisi atribuute, aga üldistatuna saab jagada atribuudid järgnevalt:

- Tehnilised atribuudid: sisaldavad andmeid alates protsessorite arvust, IP-aadressist kuni ketta mahuni gigabaitides.
- Organisatsioonilised atribuudid: kajastavad CI seotust teenuse, tehnilise teenuse, meeskonna või valdkonnaga.
- Kulumudeliga seotud atribuudid: kirjeldavad taristu ja andmemahtudega seotud hindasid.

„Server Chefist“ tüübi puhul on lisatud ka Chefi haldusvahendile omased atribuudid, mis ei ole praeguseks enam asjakohased, kuna Chef kui haldusvahend on asutuses kasutuselt eemaldatud. Analüüsitavas asutuses on viimase kahe aasta jooksul läbi viidud projekt, mille käigus kolis enamik servereid ja teenuseid uuele taristule. Selle protsessi käigus ei loodud integratsiooni, mis oleks IT komponentide kohta CI-sid automaatselt loonud.

Autori poolt analüüsi läbiviimise ajal on CI Jira projektis kokku üle viie tuhande objekti, millest aktiivseid on veidi üle kuuesaja. Viimase saja päeva jooksul on aktiivsetest CI-dest uuendatud vaid kümnet CI-d. Autor analüüsis toodud andmeid vaid põgusalt ning andmeanalüüsi tulemusena leidis, et andmed on järjepidevad ainult tehniliste atribuutide osas, mida on automaatselt uuendanud Chef haldusvahend. Muudes kategooriates varieerub andmete täpsus sõltuvalt valdkonnast ja sellest kui manuaalne on olnud nende täitmine. Paljud atribuudid on kaotanud oma relevantsuse seoses organisatsioonis toimunud muutustega ja andmed on ilmselgelt aegunud. Autori arvates ei ole piisav IT süsteemide puhul kaardistada ainult servereid ja andmebaase, seda eriti mõjuanalüüsi teostades.

Autor leiab, et olemasoleva CMDB atribuutide või arhitektuuri põhjalikum analüüs ei lisaks antud töö kontekstis olulist lisandväärtust.

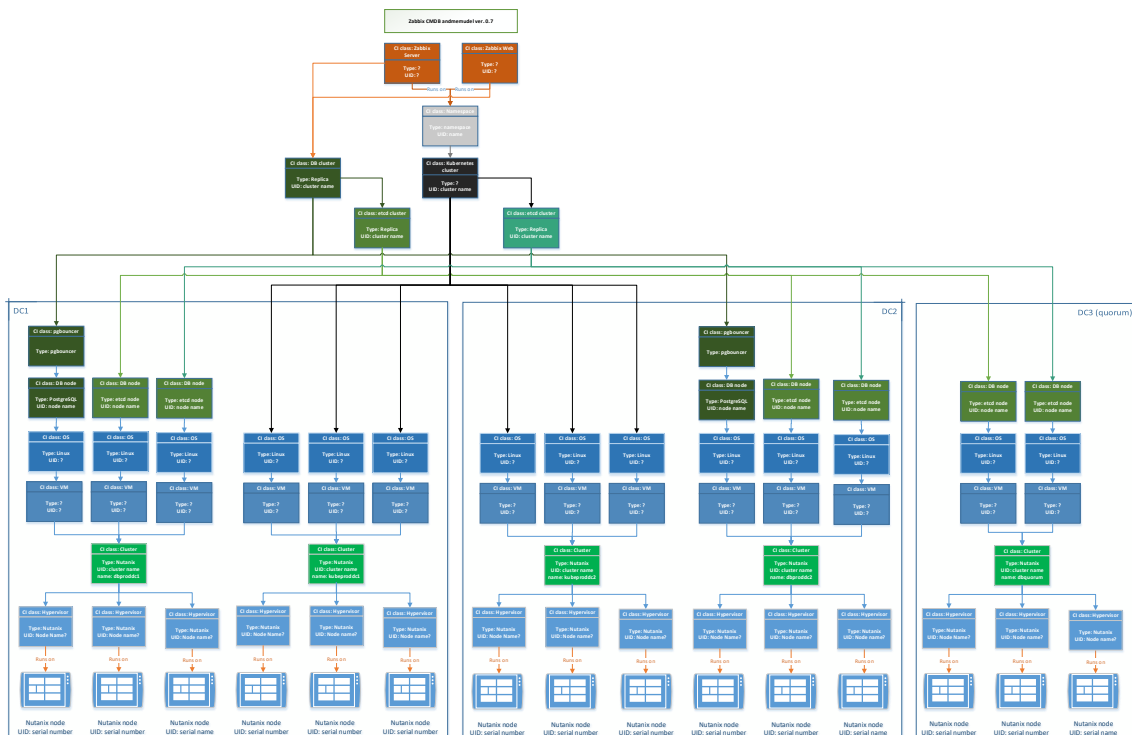
2.2 Zabbix süsteemi kaardistus

CMDB andmemudel sõltub paljudest erinevatest aspektidest, organisatsiooni üles ehitusest, organisatsiooni struktuurist, operatsioonimudelist, haldusvastutusest, tehniliste lahenduste konfiguratsioonist ja teenusmudelist. Planeerimaks CMDB andmemudelit on autor otsustanud kaardistada mõne reaalse süsteemi komponendid ja nende vahelised seosed. Sõltuvalt süsteemi konfiguratsioonist võib võtta aluseks, et olulised komponendid võiksidki esindada erinevaid CI klasse või tüüpe millel on väärtus ja mis vajavad jälgimist, kontrolli või haldamist. Tuginedes ITIL 4 konfiguratsioonihalduse praktilisele juhendile [10], Gartneri artiklile [11] ja autori enda varasemale CMDB arendamise kogemusele, on autor enda jaoks defineerinud teatavad põhimõtted mis võiks osutada millist tehnilist komponenti peaks CI-na käsitlema.

CI-d määratletakse järgmistel juhtudel:

- Kui haldusvastutus asub erinevates organisatsioonilistes üksustes (näiteks andmebaasimeeskond vs operatsioonisüsteemi meeskond, andmekeskuse tehnikud vs võrgu administraatorid).
- Kui objekti varundamine või monitoorimine toimub eraldi või erinevalt (näiteks andmebaas vs operatsioonisüsteem).
- Kui objekt on tehniliselt eraldiseisev kiht lahenduses ja oluline mõjuanalüüsi mudelis.

Kuna monitooringusüsteemid kuuluvad meie vaadeldavas asutuses autori vastutusvaldkonda ja monitooringu platvorm on piisavalt keerulise arhitektuuriga ning sisaldab erinevate taristu ja platvormi kihtide komponente ning on kirjeldatud teenuste portfellis nii äriteenuse kui tehnilise teenusena, on autor valinud Zabbixi monitooringusüsteemi näidiseks, mida kaardistada ja kasutada CMDB andmemudelili põhimõtete rakendamise illustreerimiseks.



Joonis 2. Zabbix monitooringu komponendid

Joonisel on kujutatud Zabbixi monitooringusüsteemi komponendid, järgides põhimõtteid, mille kohaselt eristatakse komponente, mida haldab eraldi üksus, või mis on olulised mõjuanalüüsi seisukohalt. Kaardistus on koostatud süsteemi veakindluse arhitektuuri arvesse võttes, et esile tuua erinevad asukohad ning andmete sünkroniseerimise ja kontrollimise mehhanismid. Kaardistuse põhjal saab eristada järgmised CI tüübid:

- **Asukoht** - Asukoht on piisavalt laia tõlgendamise piiridega objekt, mis tähistab erinevat tüüpi asukohti, kus paiknevad serveri või võrgu komponendid. See võib ulatuda andmekeskusest kuni kontorihoone kindlal korrusel asuva kommunikatsioonikapini. Selline lähenemine võimaldab luua lisanduvaid andmeobjekte, nagu aadress, korrus, kapp ja muid seotud detaile, ning täiendada asukoha abstraktset objekti läbi seoste.

- **Riistvara** - Serveri riistvara CI klassi kirjeldamisel on oluline arvestada, et serveri riistvara koosneb erinevatest lahendustest ja konfiguratsioonidest, mis varieeruvad tootjate ja arhitektuuride lõikes. Laiendatava ja paindliku mudeli loomiseks võiks CI andmemudelil eristada serveri riistvara ja serveri riistvara komponenti kui eraldi CI klasse. Serveri riistvara CI klass hõlmaks terviklikke seadmeid, nagu füüsilised serverid ja andmekeskuste serveriplatvormid, samas kui serveri riistvara komponenti CI klass kirjeldaks nende seadmete spetsiifilisi osi, nagu protsessorid, mälumoodulid, kõvakettad, toiteallikad ja muud olulised komponendid.
- **Hüperviisor** - Põhimõtteliselt töötab hüperviisor riistvara peal nagu operatsioonisüsteem. Kuigi võiks luua ühe CI klassi, mis kirjeldab operatsioonisüsteemi ja määratleda hüperviisor selle alam-tüüpi CI klassina, pean vajalikuks käsitleda hüperviisorit eraldi CI klassina. Selle põhjuseks on asjaolu, et hüperviisori atribuutide nõuded ja teave erinevad sageli tavapärase serveri operatsioonisüsteemi atribuutidest. Seetõttu väärrib hüperviisor eraldi CI klassi loomist, kus erinevad alamtüübid saab määratleda konkreetsete tehnoloogiate alusel, näiteks Nutanix, VMware, KVM jt.
- **Klaster** - on teatud määral abstraktne objekt, kuna see on kontseptsioon, mis viitab mitme seadme või süsteemi ühtsele kogumile, mis toimib ühe tervikliku üksusena. Klasterit defineeritakse kui grupi süsteemide loogilist kooslust, mis koos töötades jagavad ressursse ja töökoormust, kuid selle realiseerimiseks on vaja konkreetseid füüsilisi või virtuaalseid komponente. Meie vaates on oluline klasterit kirjeldada kui eraldi objekti nii mõju mudeli vaates kui ka klient-süsteemide vaates, kuna kasutajad ja süsteemid suhtlevad klasteriga kui ühe tervikuga, mitte eraldi komponentidega. Meie analüüsitava süsteemis on mitu erinevat kihis erinevat tehnoloogia tüüpi klasterit; PostgreSQL klaster, Nutanix klaster, etcd klaster, Kubernetes klaster.
- **Virtuaalmasin** - virtuaalmasina eristamine operatsioonisüsteemist on oluline teatud konfiguratsioonides kus operatsioonisüsteemi ja virtuaalmasina elutsükli on erinevad, haldusvastutus on erinev või replikatsioonilahenduste puhul kus virtuaalmasin kui virtuaalne riistvara tekitakse või reserveeritakse ilma sinna peale aktiivset operatsioonisüsteemi käivitamata. Sellistes situatsioonides on

oluline pidada arvestust reserveeritud virtuaalse riistvara üle ressursi haldamise eesmärgil, näiteks VMware SRM lahenduse puhul.

- **Operatsioonisüsteem** - operatsioonisüsteemi eristamine eraldi CI-na on oluline, kuna operatsioonisüsteemi elutsüklid, haldusvastutus ja konfiguratsioon võivad erineda aluseks olevast riistvarast või virtuaalmasinast. Näiteks võivad replikatsioonilahendused või teatud taastestsenaariumid nõuda operatsioonisüsteemi käsitlemist sõltumatult. Operatsioonisüsteemi eristamine võimaldab täpsemat haldust, näiteks erinevate versioonide, paigalduste ja konfiguratsioonide jälgimist ning selgemaid vastutuse piire süsteemiadministraatorite ja muude haldusüksuste vahel.
- **Andmebaasi sõlm** - andmebaasi CI-na kirjeldamine sõltub konkreetse süsteemi lahendusest ja kasutatavast tehnoloogiast. Erinevate lahenduste, nagu PostgreSQL, MySQL või Oracle DBMS, puhul võivad nõuded ja arhitektuurilised mudelid olla erinevad, olenevalt sellest, millisel tasemel klaster funktsionaalsus on lahendatud. Süsteemides, kus klaster on oluline komponent, on andmebaasi sõlme kirjeldamine vajalik mõjuanalüüsi ja halduse seisukohalt. Käesolevas töös piirduakse andmebaasi sõlmede käsitlemisel vastavalt Zabbixi arhitektuurile ja ei analüüsi teisi andmebaaside platvorme või arhitektuure.
- **Pgbouncer** - *pgbouncer* on PostgreSQL-i ühenduste vahendaja. See on kergekaaluline ühenduste vahendamise tööriist (*connection pooler*), mida kasutatakse PostgreSQL andmebaasi jaoks, et hallata ja optimeerida ühenduste kasutamist. Meie süsteemis on oluline *pgbouncer* eraldi CI-na kaardistada, kuna selle kasutamine või mittekasutamine erinevates süsteemides võib oluliselt mõjutada monitooringu seadistusi ja mängida rolli mõjuanalüüsis, eriti muudatuste ja intsidentide haldusprotsessides.
- **Nimeruum** - on Kuberneteses loogiline eraldusüksus, mida kasutatakse erinevate ressursside ja töökoormuste eristamiseks samas Kubernetes'e klasteris. Kubernetese klasteri lihtsustatud mõjuanalüüsi mudelis meil ei ole vaja eraldi kaardistada *pod*-e, üksiku *pod*-i olekut või elutsüklit ei ole meil vaja arvestada. Meie analüüsitavas süsteemis on terviksüsteemid eraldatud nimeruumidesse ja see oleks minimaalne abstraktsioonikiht mida kirjeldada püstitatud ülesande skoobis.

- **Zabbix web, Zabbix Server** – Järgnevad loogilised tehnoloogilised kihid vajavad põhjalikumat analüüsi, et määratleda, kuidas neid kõige sobivamalt kirjeldada ja kas need komponendid võiksid kuuluda rakenduse CI klassi alla. Autor jätab selle küsimuse hetkel lahtiseks ja keskendub kitsamale skoobile.

2.3 Atlassiani IT Assets mooduli praktikad

Atlassiani dokumentatsioonis [12] soovitatakse Assets moodulis luua *object schemas* (edaspidi objektiskeemid), võttes arvesse andmete tundlikkust, haldusmeeskondade vastutusalasid ja andmete mahu. Mitme skeemi loomine võib olla kasulik järgmistel põhjustel:

- **Andmete haldamise lihtsustamine:** Andmete jagamine väiksemateks osadeks aitab auditeerida ja hoida andmed täpsed.
- **Juurdepääsupiirangud:** Kui tegu on tundlike andmetega, näiteks töötajate teave, võib olla lihtsam hoida kogu see teave ühes objektiskeemis, millel on piiratud juurdepääsuload.

Objektiskeemid on lisaks ka haldusüksus mille kaudu hallatakse kasutajate rolle ja õiguseid objektide suhtes, seega objektiskeemide puhul tuleb arvestada mis tüüpi andmeid plaanid objektiskeemi skooopi ja millised grupid ja kasutajad haldavad andmeid ja millised kasutajagrupid vajavad ainult lugemise õiguseid.

Oluline on märkida, et kui on vaja luua kahepoolseid seoseid objektide vahel erinevates objektiskeemides on vaja kasutajal kirjutamise õigust mõlema objektiskeemis. See mõjutab seoste tüüpide ja suundade disaini olulisel määral kui mõtleme CI-de loomise ja teenustega sidumise protsessi peale. Kui teises skeemis on ainult lugemisõigus, saab luua ühesuunalise seose, mis algab oma skeemist ja viitab teise skeemi objektile. Kui eeldame, et serveri CI loob serveri administraator ja ka tema peaks märkima millise teenusega server seotud on, ei ole mõistlik, et serveri administraatoril on kirjutusõigused teenuste või inimeste või ruumide ja asukohtade objektiskeemidesse. Teisest küljest ei saa eeldada, et tehniliste komponentide sidumine teenustega jääb teenuseomaniku kohustuseks, see ei ole korelatsioonis tarneportsessidega. Kui kasutada seoste loomiseks integratsioonivahendeid (nt API), peab kontrol olema vastavad õigused mõlemas skeemis. Sel juhul soovitab autor igale integratsioonile eraldi konto loomist mis hajutab riske ja võimaldab paremat muudatuste halduse ja vigade jälgimise rakendamist.

Objektitüüpide loomisel soovitatakse need määratleda vastavalt sarnastele objektidele, mis jagavad ühiseid atribuute ja omadusi. Näiteks võib luua eraldi objektitüübid serverite, rakenduste ja võrguseadmete jaoks. Iga objektitüüp võib omada ühte vanem- ja mitut

alamtüüpi, võimaldades parandada atribuute vanematelt tüüpile. Samuti saab määrata teatud objektitüübid abstraktseteks, mis tähendab, et need ei sisalda oma objekte, kuid saavad parandada oma atribuute alamtüüpile.

Atlassiani Assets-i moodul pakub vaikimisi objektiskeemi malle, mis hõlbustavad erinevate kasutusjuhtumite jaoks vajalike objektitüüpide ja atribuutide loomist. Need mallid sisaldavad eelmääratletud struktuure ja atribuute, mis kiirendavad konfiguratsioonihalduse andmebaasi ülesehitamist. Peamised saadaval olevad mallid:

- **IT-varade haldus (*IT Asset Management*):** Sisaldab objektitüüpe nagu serverid, tööjaamad, tarkvara ja võrguseadmed. Iga objektitüüp on varustatud asjakohaste atribuutidega, näiteks seadme nimi, seerianumber, asukoht ja vastutav isik.
- **Inimesed (*People*):** Sisaldab objektitüüpe nagu töötajad ja meeskonnad. Atribuutide hulka kuuluvad nimi, ametinimetus, kontaktandmed ja osakond.
- **Rajatised (*Facilities*):** Sisaldab objektitüüpe nagu hooned, ruumid ja töökohad. Atribuudid võivad hõlmata asukohta, ruumi numbrit, otstarvet ja seotud varasid.

Konfiguratsioonihaldur saab neid malle kohandada vastavalt vajadustele, lisades või eemaldades atribuute ning muutes objektitüüpide vahelisi seoseid, aga autor märkas, et kasutades eelmääratud malle koos eelmääratud atribuutidega, puudub võimalus nende eeldefineeritud atribuutide nimesid või andmetüüpe muuta. Autor soovitab, objektiskeemi mallidest eelmääratud struktuuride teadmisi kasutades, siiski luua objektitüübid malle kasutamata.

2.4 CMDB loomise raamistik

Gartneri artikkel „Kuidas moderniseerida konfiguratsioonihalduse andmebaasi“ [11] käsitleb üsna laialdaselt põhiprobleeme ja võimalikke strateegiaid kuidas moderniseerida juba eksisteerivat konfiguratsioonihalduse andmebaasi või kuidas luua see organisatsioonis mis alles teeb oma esimesi samme ITILi põhimõtete juurutamisel. Antud töö kontekstis on autor enda jaoks mõtestanud järgmised põhimõtted:

- **Esmane vajaduste analüüs** - alustada tuleb organisatsiooni vajaduste, teenusmodeli, tehnilise arhitektuuri ja haldusmodeli analüüsist, et disainitav lahendus suuremas skaalas arvestaks nende eristustega ja oleks paindlik ning laiendatav.
- **Alusta väiksel ja iteratiivselt** - rakenda minimaalset tehnoloogiliste kihtide ja atribuutide kogumit, keskendudes vaid informatsioonile, millel on selge väärtus äri vajaduste ja teenusehalduse seisukohalt. Esimeste CI-de andmete sisestamine ja integreerimine võimaldab kiiresti tuvastada kasutatavate platvormide ja lahenduste kitsaskohad ning teha vajalikke kohandusi enne ulatuslikuma mastaabi rakendamist.
- **CI ja komponentide elutsükli korrelatsioon** - CI-de elutsükkel peab olema kooskõlas komponentide elutsükliga. See tagab, et CMDB andmed peegeldavad täpselt IT-komponentide seisundit ja rolli organisatsiooni infrastruktuuris.
- **Selged vastutused ja omanikud** - igal CI-l peab olema selgelt määratletud vastutaja ja andmetel omanik. CMDB ehitamine ja hooldus on organisatsiooni kollektiivne ülesanne mida ei saa delegeerida ainult konfiguratsioonihalduritele. Andmete uuendamise eest vastutavad otseselt need, kes loovad ja haldavad IT-komponente, millele CI-d viitavad.

3 Muudatuste ettepanekud ja pilootprojekti sisend

Kehtiva teenusmudeli puhul autor on jõudnud järeldustele, et tehnilise teenuse ja äriteenuse piirid on hägusad, paljuski jäänud tootemaniku enda otsustada mida ta defineerib äriteenusena ja mida tehnilise teenusena.

Tehniliste teenuste kirjeldamine on väga erinev, mis tekitab ebaühtlust. Mõnel juhul on tehniline teenus määratletud üksiku platvormi või tugiteenusena, näiteks „TEH-5 Artifactory“. Teistel juhtudel on teenus kirjeldatud asukoha- või installatsioonipõhise süsteemina, nagu „TEH-658 Kontorivõrk Ida-Harju KP“. Kolmandates olukordades on see esitatud kui üldine teenuspakkumine, mis hõlmab kõiki sarnaseid tehnoloogilisi platvorme, näiteks „TEH-42 PostgreSQL“. Samuti puudub tehnilistel teenustel sageli käideldavuse klass või ei eristata tootmiskeskondade nõudeid teistest keskkondadest.

Autor on välja töötanud uue potentsiaalse teenusemudeli arhitektuuri, mis võimaldaks paremini kirjeldada teenustingimusi süsteemi, platvormi või tehnilise teenuse lõikes. Mudeli täpsemaks illustreerimiseks on autor loonud skeemi koos fiktiivsete näidisedandmetega (vt Joonis 3).

3.1 Äriteenus

Autor soovib defineerida äriteenuse objekti kui teenuselepingut kliendi ja teenusepakkuja vahel. Üks teenusleping ehk äriteenus (SRV) võib sisalda rohkem kui ühe süsteemi teenust ja rohkem kui ühe keskkonda. Sarnast platvormi või teenust võib vajada mitu erinevat klienti, aga erinevatel tingimustel ja nõuetel, sel juhul luuakse äriteenus (SRV) iga kliendi puhul eraldi. Äriteenus võimaldab määratleda ja parandada teenustingimused, aktiivse tööaja, reageerimisnõuded ja kliendikontaktid järgmistele teenuse ja komponentide kihtidele kust neid saab omakorda kasutada haldusvahendite ja monitooringu integratsioonide ehitamisel. Äriteenusel lähtudes peab tootemanik jaotama teenuse edasi loogilisteks tehnilisteks teenusteks, mis on eristatavad platvormi, keskkonna või funktsiooni alusel.

3.2 Tehniline teenus

Tehniline teenusena soovib autor kirjeldada loogilist kogumit rakenduse, süsteemi või platvormi paigaldustest, mis on vajalikud konkreetse funktsionaalsuse või teenuse pakkumiseks. Tehniline teenus kirjeldab süsteemi, platvormi või rakenduse tööks vajalikku terviksüsteemi. Sama äriteenust võivad toetada mitmed tehnilised teenused, millel on erinevad teenusnõuded, tingimusel, et need koos täidavad äriteenuses kehtestatud SLA (Service Level Agreement) süsteemi arhitektuuri kaudu. Erinevate platvormide või komponentide käideldavusenõuded võivad olla erinevad kuni nad täitavad üldise kogumina tehnilise teenuse käideldavuse nõudeid. Näiteks riistvara teenuse puhul võib käideldavus olla K1 kui tegemist on mitme-õlalise klastriga kus ühe riistvara komponendi häire puhul ei ole K2 taseme platvormi teenus häiritud. Seetõttu koosneb tehniline teenus tavaliselt erinevate arhitektuuridega platvormidest ja komponentidest, mida haldavad erinevad meeskonnad erinevatel alustel. Autor soovib nende kirjeldamiseks luua eraldi tehnilise teenuse tüübi: tehnilise teenuse teenuspakkumus.

3.3 Tehniline teenuse teenuspakkumus

Kuna erinevaid platvorme või keskkondasid saab sõltuvalt äriteenuse käideldavuse tingimustest erinevalt seadistada või erineva tasemega hallata, siis tuleks defineerida tehniliste teenuste teenuspakkumised. Teenuspakkumus kirjeldab süsteemis osalevat erinevate platvormide või komponentide teenustasemeid. Teenuspakkumuse objektis saame kirjelda:

- **OLA:** Sisemised kokkulepped, mis määravad, kuidas tehniline teenus täidab äriteenuse SLA-s sätestatud nõudeid.
- **Reageerimisajad:** Aeg, mille jooksul tõrkele reageeritakse või monitooringualarm aktiveerub, sõltuvalt teenuse tähtsusest ja prioriteedist.
- **Aktiivne tööaeg:** Teenuse saadavuse ja tagamise ajavahemik (nt tööpäevadel 8:00–18:00 või 24/7).
- **Hooldusaknad:** Planeeritud ajad süsteemi hoolduseks või uuendusteks, mille jooksul teenus võib olla osaliselt või täielikult kättesaamatu.
- **Taasteplaan ja RPO/RTO tingimused:** Strateegiad ja meetmed, mis määratlevad, kuidas teenus taastatakse rikete või hädaolukordade korral. Need hõlmavad nii teenuse taastamise protsessi kui ka taastamise ajakriteeriume, sealhulgas RPO (*Recovery Point Objective*) ehk maksimaalne lubatud ajavahemik, mille jooksul võib andmeid kaotada enne taastamist ja RTO (*Recovery Time Objective*) maksimaalne aeg, mille jooksul teenus peab olema taastatud, et äriteenuses kehtestatud SLA ja andmekao tingimustest kinni pidada.
- **Haldusmeeskonnad:** Milline meeskond vastutab konkreetse teenuspakkumuse täitmise eest (nt serverite haldus, andmebaaside haldus), seda läbi seose meeskond (MSK) objekti.

3.4 Meeskond

Meie vaadeldavas organisatsioonis on loodud meeskonna nimeline objekt (MSK) mis on abstraktne ja alati ei ühti organisatsiooni struktuuriga, aga on vajalik teenuste haldusvastutuste, kasutaja õiguste määramiseks, automaatika ja statistika realiseerimiseks. Meeskonna objekt sisaldab juba infot vastutaja, liikmete, gruppide ja kontaktide kohta.

Autor teeb ettepaneku lisada meeskonna objekti ka eskalatsiooniahela informatsioon. Hetkel on eskalatsiooniahelad kirjeldatud teenuse objektide juures ning need on koostatud käsitsi, sõltuvalt sellest, millised platvormid ja komponendid on seotud konkreetse äriteenuse funktsionaalsuse pakkumisega. See omakorda tähendab, et informatsioon kuidas näiteks andmebaasi platvormi häirete puhul käituda, on mitmekordselt dubleeritud erinevate teenuste puhul. Kui omakorda peaks andmebaasihalduse tööruutiniidid või eskalatsiooniahela loogika muutuma, vajaks see käsitsi muutmist kõikide teenuste kirjeldustes mis kasutavad andmebaasi platvorme. Autori hinnangul oleks efektiivsem kui see info oleks kirjeldatud meeskonna objekti juures ja kättesaadav läbi teenusemudeli seoste. Ka võimaldaks selline lahendus ehitada loogika kus monitooringu alarmiga kaasa antakse ka käitumisjuhise kasutajatoe spetsialistile sõltuvalt teenuspakkumise ja meeskonna ja IT-komponendi objektis kirjeldatud parameetritest.

3.5 CMDB ja CI-de loomine

Infrastruktuuri ja platvormi komponentide kirjeldamine on üldjuhul CMDB loomise lihtsaim osa, kuna nende maht ja sarnasus teevad kaardistamise ja haldamise hõlpsamaks. Organisatsioonides on tavaliselt kasutusel piiratud arv tehnoloogiaid ja versioone, mis on kuluefektiivsemad ja lihtsamini hallatavad. Lisaks on sageli nende haldamiseks kasutusel tsentraliseeritud tööriistad, mida saab ära kasutada CMDB andmete laadimisel. Üks levinumaid komponente, mida CMDB loomisel esimesena kaardistatakse, on serverid. Sageli ühendatakse operatsioonisüsteem ja riistvara üheks CI-ks, et lihtsustada andmete haldamist, aga antud juhul on nendel kahel objektil erinev elutsükkel ja erinevad haldusmeeskonnad. Riistvara tasandi kaardistamisel ja andmete laadimisel võib aga väiksemates organisatsioonides probleemiks saada tsentraliseeritud haldusvahendite puudumine, kuna need võivad olla kulukad.

3.6 Objektiskeemid

Pilootprojekti käigus soovitab autor luua kaks objektiskeemi:

1. „**Asutuse IT varade haldus**“:

- Tühi skeem, kuhu luuakse IT-komponentide CI tüübid.
- Skeem koondab taristu- ja platvormikomponendid ühte kohta.

2. „**IT varade halduse mall**“:

- Atlassiani poolt eeldefineeritud IT varade halduse mall.
- Kasutatakse järgmiste iteratsioonide analüüsiks ja juhendiks CI-tüüpide defineerimisel.

Esialgse plaani kohaselt kirjeldatakse taristu- ja platvormikomponendid ühes objektiskeemis, et tagada lihtsus andmete ja seoste haldamisel. Selle skeemi objekte haldavad süsteemiadministraatorid, kellel on sarnased õigused, mis võimaldab ühtlustatud juurdepääsu ja vastutust. Meeskondade, inimeste, rajatiste objektiskeemide loomine jääb järgmisesse etappi.

3.7 CI atribuudid

Kõikidele CI-dele loob Atlassiani Asset moodul neli vaikimisi atribuuti:

- **Nimi (*Name*)** - CI äratuntav nimetus, valdavalt inimloetav, mis võimaldab kiiret objekti tuvastamist. Näiteks serveri nimi, rakenduse nimi või teenuse nimi. Atribuuti saab seadistada unikaalsust kontrollivaks väljaks.
- **Võti (*Key*)** - unikaalne identifikaator CI-le luuakse automaatselt. CI loomise järel ei saa võtit muuta. CI võti põhineb objekti skeemi võtmel, millele lisandub unikaalne identifitseeriv täisarv. Kui luua mitu CI-d korraga, ei pruugi nende võtmed olla alati järjestikused.
- **Loodud (*Created*)** - CI loomise kuupäev ja kellaaeg. Seda atribuuti ei saa muuta.
- **Uuendatud (*Updated*)** - CI viimase muutmise kuupäev ja kellaaeg. Seda atribuuti uuendatakse iga kord, kui CI-d või mõnda selle atribuuti muudetakse.

Lisaks soovitab autor minimaalset järgmist atribuudi kogumit üle infrastruktuuri- ja platvormi komponentide objektiskeemi CI-de:

- **Haldusmeeskond** - meeskond, kes vastutab CI haldamise, ajakohasuse ja jõudluse eest, väli on refereeritud läbi seose meeskonna (MSK) objektist.
- **Olek** - IT-komponendi hetkeseis, näiteks aktiivne, plaanitud, kasutuselt kõrvaldatud, testimisel või hoolduses. Olekute variandid võivad varieeruda ka sõltuvalt komponendi tüübist, näiteks tarkvaral ja riistvaral võivad kasutuses olla erinev komplekt olekuid. Oluline on olekud dokumenteerida ja ühtlustada üle erinevate objektiskeemide, et tagada ühtne struktuur ja andmete kvaliteet.
- **Unikaalne identifikaator** - identifikaator mis võimaldab üheselt seostada CI-d komponendiga mida CI peegeldab. Autori kogemus toetab UID defineerimist objekti enda atribuudina või atribuutide kombinatsioonina, kuna CMDB poolt genereeritud identifikaatorit teatud tehnoloogiate puhul ei ole võimalik salvestada IT komponenti ja sellega võib kaduda võimalus objekti ja CI-d omavahel selgelt seostada. UID ei pea olema eraldi väli vaid igal CI-l peab olema tähistatud atribuut mis täidab UID rolli.

3.8 Virtuaalmasina CI

Virtuaalmasina esialgsete atribuutidena esimeses lahendusetapis võiks kasutada järgmisi:

Tabel 1 Virtuaalmasina CI atribuutide loend

Väli	Välja tüüp	Näide	Seletus
Key	tekst	BT-001	Automaatselt loodavid/serial, schema_id + CI_id
Created	aeg	14:00 10.12.2024	CI loomise kuupäev, muutumatu
Updated	aeg	23:59 15.12.2024	CI viimase muutmise kuupäev
Name*	tekst (unikaalne)	cba7dc40-a45f-4823-8a8d-7b2e7119be64	Nutanix VM UUID
Olek	valik	Aktiivne	Plaanitud, aktiivne, rikkis, eemaldatud
Meeskond (MSK)	viide	MSK-222	haldusmeeskond, viide MSK objekti nime väljale
vCPU	number	4	virtuaalsete protsessorite arv tuumades
vMEM	number	12	mälu maht gigabaitides

3.9 Operatsioonisüsteemi CI

Operatsioonisüsteemi CI atribuutidest on autor teadlikult välja jätnud IP. Ühel operatsioonisüsteemil võib olla rohkem kui üks IP aadress ja mõnes järgmises iteratsioonis võib kaaluda lahendust kus on kaasatud „võrguadapteri“ CI tüüp koos „IP aadressi“ ja „võrgu segmendi“ CI tüüpidega, täpne lahendus vajab veel täiendavat analüüsi.

Tabel 2 Operatsioonisüsteemi CI atribuutide loend

Väli	Välja tüüp	Näide	Seletus
Key	tekst	BT-001	Automaatselt loodavid/serial, schema_id + CI_id
Created	aeg	14:00 10.12.2024	CI loomise kuupäev, muutumatu
Updated	aeg	23:59 15.12.2024	CI viimase muutmise kuupäev
Name*	tekst (unikaalne)	server1-ds-msk.domeen.ee	Täielikult kvalifitseeritud domeeninimi (FQDN)
Olek	valik	Aktiivne	Plaanitud, aktiivne, rikkis, eemaldatud
Meeskond (MSK)	viide	MSK-111	Viide MSK objekti nime väljale
OS domain	valik	Windows, Linux, Unix	OS vendor/brand/tüüp
OS	valik	RedHat, CentOS, Ubuntu, Windows Server, Solaris	Distributsiooni ja/või versiooni nimi
OS Version	tekst	2003, 8.2, 2020R2 EE,	Versiooni väljalase koos litsentsitüübiga
OS patch level	tekst	19.2.0.4.1	Kogu versioon kuni viimase paiga numbrini
Monitooringu URL	tekst	https://zabbix.asutus.ee/check/item.htm	URL konkreetse serveri monitooringule

3.10 Klaster CI

Vaadeldavas asutuses on Nutanixi klastrid ehitatud vastavalt keskkondadele, turvaklassidele, teenuse tüüpidele ja asukohtadele. Selle CI tüübi ja andmete sidumine serveritega pakub olulist lisaväärtust, eriti tarkvarauuenduste ja serverite halduse kontekstis.

Tabel 3 Klaster CI atribuutide loend

Väli	Välja tüüp	Näide	Seletus
Key	tekst	BT-001	Automaatselt loodav id/serial, schema_id + CI_id
Created	aeg	14:00 10.12.2024	CI loomise kuupäev, muutumatu
Updated	aeg	23:59 15.12.2024	CI viimase muutmise kuupäev
Name*	tekst (unikaalne)	web-db-clu.domeen.ee	Klastri unikaalne nimi
Olek	valik	Aktiivne	Plaanitud, aktiivne, rikkis, eemaldatud
Meeskond (MSK)	viide	MSK-222	haldusmeeskond, viide MSK objekti nime väljale
Tüüp	valik	Postgres multimaster replica	Klastri tüüp

3.11 Seoste tüübid

Autori soovitusel võiks pilootprojekti raames esialgu defineerida minimaalse hulga seoseid:

- **Sõltub** - näitab, et teenuse või komponendi toimimine sõltub konkreetsest teisest komponendist või funktsioonist või teenusest.
- **Haldab** - näitab, milline teenus või meeskond vastutab CI haldamise eest.
- **Kuulub** - näitab millise tehnilise teenuse koosseisu komponent kuulub.

Projekti hilisemas faasis vajame tõenäoliselt juurde rohkem seoseid nagu näiteks „osaliselt sõltub“, „omab“, „kasutab“, „ühendab“, mis võimaldavad täpsemalt kirjeldada komponentide vahelisi keerulisi suhteid ja nende mõju süsteemi üldisele toimimisele.

Autori hinnangul on selline andmemaht esialgse pilootprojekti jaoks piisav. Pilootprojekti peamine eesmärk on koguda kogemusi integratsioonide loomisel ning CI-de ja IT-komponentide andmete elutsükli ühtlustamisel, et saadud teadmisi rakendada andmemudeli edasiseks laiendamiseks ja täiustamiseks.

3.12 CMDB haldusrollid ja dokumenteerimine

CMDB edukaks juurutamiseks asutuses on oluline mõista, et selle loomise ja arendamise protsessi ei saa delegeerida üksnes konfiguratsioonihaldurile või ühele meeskonnale, kes vastutaks kogu organisatsiooni IT-komponentide tuvastamise ja kaardistamise eest. CMDB loomine ja pidev arendamine on kollektiivne tegevus, milles osalevad kõik organisatsiooni liikmed.

Lisaks ei saa eeldada, et konfiguratsioonihaldur omab eksperttasandil teadmisi kõigist kasutatavatest tehnoloogiatest, arhitektuuridest ning nende elutsüklitest. Seetõttu on CMDB tõhusaks haldamiseks ja arendamiseks vajalik rollide ja vastutusala selge jaotus. Autori ettepanek on jaotada CMDB haldus kolmeks peamiseks rolliks:

- **Konfiguratsioonihaldur** – CMDB „peatöövõtja“, seab strateegilised eesmärgid konfiguratsioonihaldusele, koostab arhitektuuri ja põhimõtted, veendub, et andmemudel oleks ühtlane ja sidus erinevates kihtides ja tehnoloogiates. Tagab, et CMDB-s loodud mudelid toetavad oma eesmärgi ja pakuvad väärtust protsessidele ning kasutajatele. Veendub, et konfiguratsioonielemendid (CI-d) on korrektsed koos nende tüübikirjelduste ja andmemudelig ning aitab CI tüübi omanikke sellega seotud küsimustes. Loob ja haldab CI tüüpe vastavalt CI tüübi omanike taotlustele. Tagab, et andmete loogika CMDB-s on selge ja dokumenteeritud. Veendub, et andmeid muutvad ja kasutavad integratsioonid oleksid dokumenteeritud ja tekkivad konfliktid hallatud. Loob ja haldab CMDB-ga seotud kasutajaliideseid CMDB kasutajate jaoks (CI vaated, menüüd, eelseadistatud väärtused jne). Loob CMDB andmete kvaliteedi mõõtmise ja visualiseerimise lahendused.
- **CI tüübi haldur** – Valdkonna arhitekt või vanem-administraator kes koostöös konfiguratsioonihalduriga disainib oma valdkonna CI tüübid, seosed, elutsükli ja teenuse juhtimiseks vajalikud atribuudid. Registreerib muudatused CI tüübi nõuetes ja kaardistab andmeformaadid erinevate tehniliste lahenduste vahel. Tagab, et CI tüübi nõuded ja andmeformaat on dokumenteeritud. Vastutab oma CI klassi andmete kvaliteedi eest, algatab andmekvaliteedi parandamise projekte ja kaasab sellesse CI haldureid.

- **CI haldur** – Administraator või insener kes haldab asutuses IT-komponente. Vastutab selle eest, et kõik tema hallatavad objektid oleksid CMDB-s CI-dena kajastatud. Vastutab CI-de uuendamise eest vastavalt CI tüübi nõuetele ja dokumentatsioonile. Ehitab andmete automaatselt uuendamise integratsioone haldusvahenditega ja dokumenteerib need vastavalt nõuetele.

CMDB-ga seotud dokumentatsioon peab olema selgelt struktureeritud ja kättesaadav kõigile asjakohastele osapooltele. Kõik loodud CI tüüpide ja atribuutide tehniline dokumentatsioon peab olema ajakohane, täpne ning hõlpsasti arusaadav. Lisaks tehnilisele dokumentatsioonile on soovitatav luua visuaalsed mudelid, mis illustreerivad CI-de vahelisi seoseid ja erinevaid konfiguratsioone. Need mudelid, sarnaselt Zabbix monitooringusüsteemi kaardistusele, aitavad paremini mõista andmemudeli terviklikkust.

Kõik CMDB andmeid loovad, muutvad või kasutavad integratsioonid peavad olema põhjalikult dokumenteeritud. Dokumentatsioon peab hõlmama nende toimemehhanisme, kasutatavaid andmeid ning vastastikuseid sõltuvusi, et tagada läbipaistvus ja probleemide ennetamine. Lisaks peab CI tüüpide ja nende atribuutide dokumentatsioonis olema selgelt välja toodud, millised integratsioonid kasutavad või muudavad konkreetseid atribuute. See aitab vältida konflikte, mis võivad tekkida atribuudi või selle formaadi muutmisel, ning tagab, et integratsioonid jäävad funktsionaalseks ka pärast muudatuste rakendamist.

CMDB kvaliteedi mõõtmine ja haldamine peab olema regulaarne ja konkreetne tegevus. Kvaliteedi mõõdikud ja seireraportid tuleb siduda IT-teenuste ja haldusrutiinide praktiliste vajadustega. Vajalik on jälgida CI-de andmete ajakohastamise regulaarsust ning tuvastada komponendid, mille andmeid ei uuendata või mis on süsteemist eemaldatud, kuid jäänud CMDB-sse aktiivsesse rolli. IT-komponendid, mis peaksid CMDB-s kajastuma, kuid mille kohta kirjed puuduvad, tuleb käsitleda intsidentidena. Selliste olukordade puhul tuleb analüüsida protsessi vigu, et tuvastada puudujäägid haldusrutiinides või vastutuselades, mis on vea põhjustanud, ja viia sisse vajalikud parandusmeetmed.

CMDB ei ole kunagi lõplikult valmis; see on pidevalt arenev süsteem, mille väärtus seisneb selle järkjärgulises täiustamises ja integreerimises IT-organisatsiooni protsessidesse. Iga arenguetapp suurendab selle tõhusust ja kasulikkust, pakkudes paremat tuge IT-teenuste haldamisel ja strateegiliste otsuste tegemisel.

4 Kokkuvõte

Käesoleva bakalaureuse töö eesmärgiks oli kavandada ja uurida konfiguratsioonihalduse andmebaasi loomise ja rakendamise mudelit, analüüsida teenusemudeli vastavust avastatud kitsaskohtadele, luua IT-organisatsiooni jaoks konfiguratsioonihalduse andmebaasi andmemudel ja pakkuda välja lähenemisviisi ning pilootprojekti sisend selle teostamiseks.

Töö käigus analüüsis autor olemasolevat teenusemudelit ja tuvastas selle kitsaskohad. Autor tuvastas teenusemudeli dokumentatsiooni pealiskaudsuse ja hajutatuse erinevate keskkondade vahel, mis omakorda põhjustab teenuste kirjeldamise kvaliteedi ebaühtlust erinevate valdkondade lõikes. Teenuste objektides on kirjeldatud andmeid, mis autori hinnangul ei kuulu sinna ega kohaldu kõikidele teenuse aluseks olevatele tehnilistele teenustele või komponentidele. Kirjeldamata on ka sama süsteemi erinevate keskkondade teenustingimused.

Tehniliste teenuste analüüsimisel tuvastas autor, et kirjeldatud teenuste tüübid ja ulatused varieeruvad ja sisu sõltub paljuski teenust kirjeldanud isiku tõlgendusest. Lisaks ei võimalda ebaühtlane lähenemine määrata selgeid *Operational Level Agreements-e* (OLA-sid) üksikute komponentide lõikes. Nende puudumisel ei ole võimalik tagada äriteenusele kehtestatud SLA-sid. See raskendab vastutuse ja teenusnõuete täpset määratlemist ning täitmist.

Olemasolevate CI-de analüüsist lähtus, et CI tüüpideks oli realiseeritud ainult kolm alamklassi „Virtuaalne andmebaas“, „Server“ ja „Server Chefist“, mis ei ole piisav mõjuanalüüsi teostamiseks. Andmete kvaliteet oli väga kõikumine, osad atribuudid olid kaotanud oma relevantsuse seoses organisatsioonis toimunud muudatustega ja kokkuvõttes saab järeldada, et kogu olemasolev CMDB on asutuses hüljatud projekt.

Töös püstitatud eesmärgi täitmiseks kaardistas autor asutuses kasutusel oleva Zabbix monitooringusüsteemi komponendid ja analüüsis nende komponentide sobivust CI tüüpideks. Järgnevalt koostas kokkuvõtte Atlassiani IT Assets mooduli parimatest praktikatest, soovitustest ja võimalikest kitsaskohtadest. Seejärel sõnastas autor CMDB loomise raamistiku peatükis põhilised teesid, millest olulisim on alustada väikeste

sammudega ja minimaalsete andmetega, õppida kogemusest enne järgmiste iteratsioonidega edasi liikumist ja panna paika andmete omanikud ning vastutajad.

Töö tulemusena töötas autor välja uue teenusmodeli arhitektuuri, mis võimaldab paremini kirjeldada teenustingimusi süsteemi, platvormi või tehnilise teenuse lõikes. Äriteenuse objekt (SRV) kajastab teenuslepingut ja on siduv komponent teenustingimuste, aktiivse tööaja, reageerimisnõuete ja kontaktide pärandamisel järgmistele kihtidele. Tehnilise teenuse (TEH) olemus muutub objektiks, mis kirjeldab äriteenust toetavaid infosüsteeme, platvorme või keskkondasid. Teenusmodeli arhitektuuri lisandub tehnilise teenuse teenuspakkumuse (TP) komponent, mis kirjeldab süsteemis osalevate erinevate platvormide või komponentide teenustasemeid ja haldusvastutust.

Pilootprojekti teostamine ei olnud antud töö osaks. Autor on välja toonud pilootprojekti teostamiseks soovitusel luua kaks eraldi objektiskeemi IT-komponentide CI-dena kirjeldamiseks. Samuti on autor välja toonud loodud CI-de atribuutide standardloetelu koos kirjeldustega, kasutatavad seoste ja CI-de tüübid koos atribuutidega. Viimase punktina käsitleb autor soovitusi CMDB haldusrollideks ja dokumenteerimise põhimõtteid.

Antud töö tulemusi saab rakendada ka teistes sarnastes IT-organisatsioonides, mis pakuvad IT-teenuseid välistele klientidele teenuslepete alusel, praktiliseks lähenemiseks CMDB loomisel või arendamisel. Edasised tegevused uuritavas asutuses keskenduvad CMDB integreerimisele teiste tugisüsteemidega ning CMDB laiendamisele koos andmete automatiseeritud uuendamise meetoditega, et tõhustada veelgi haldusprotsesse ja teenuse juhtimise kvaliteeti.

Kasutatud kirjandus

- [1] ServiceNow, „Plan your successful CMDB deployment,“ 2021. [Võrgumaterjal]. Available: <https://www.servicenow.com/content/dam/servicenow-assets/public/en-us/doc-type/success/playbook/cmdb-deployment-plan.pdf>. [Kasutatud 08 11 2024].
- [2] M. Noormaa, Interviewee, *EITS koolitus*. 2020. [Intervjuu]. Available: <https://www.youtube.com/watch?v=0e-rDN-ntj4&t=28s>. [Kasutatud 30 11 2024].
- [3] Atlassian, „IT asset and service configuration management,“ Atlassian, 2023. [Võrgumaterjal]. Available: <https://www.atlassian.com/whitepapers/it-asset-and-service-configuration-management-in-jira-service-management>. [Kasutatud 03 03 2024].
- [4] „Teenuste portfelli ja IKT haldusprotsessid - lühitutvustus,“ 07 10 2024. [Asutusesiseks kasutamiseks dokument]. [Kasutatud 20 10 2024].
- [5] „IKT mõisted,“ 20 10 2023. [Asutusesiseks kasutamiseks dokument]. [Kasutatud 30 11 2024].
- [6] „Teenuste portfelli ja kataloogi juhtimine,“ 03 09 2024. [Asutusesiseks kasutamiseks dokument]. [Kasutatud 20 10 2024].
- [7] „Teenuste üldtingimused,“ 07 10 2024. [Asutusesiseks kasutamiseks dokument]. [Kasutatud 27 10 2024].
- [8] „Tehnilised teenused,“ 04 11 2024. [Asutusesiseks kasutamiseks dokument]. [Kasutatud 30 11 2024].
- [9] „Teenuse funktsioonid,“ 24 10 2022. [Asutusesiseks kasutamiseks dokument]. [Kasutatud 30 11 2024].
- [10] Axelos, „ITIL® 4 Service Configuration Management Official Practice Guide,“ 06 09 2024. [Võrgumaterjal] Available: <https://www.axelos.com/resource-hub/practice/readers-manual-itil-4-practice-guide>. [Kasutatud 18 12 2024].
- [11] S. White, „How to Modernize the Configuration Management Database,“ Gartner, 03 02 2020. [Võrgumaterjal]. Available: Pole avalikult kättesaadav. [Kasutatud 02 11 2024].
- [12] Atlassian, „Building an object schema for IT Assets Management (ITAM),“ [Võrgumaterjal]. Available: <https://support.atlassian.com/jira-service-management-cloud/docs/building-an-object-schema-for-it-assets-management-itam/>. [Kasutatud 28 11 2024].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

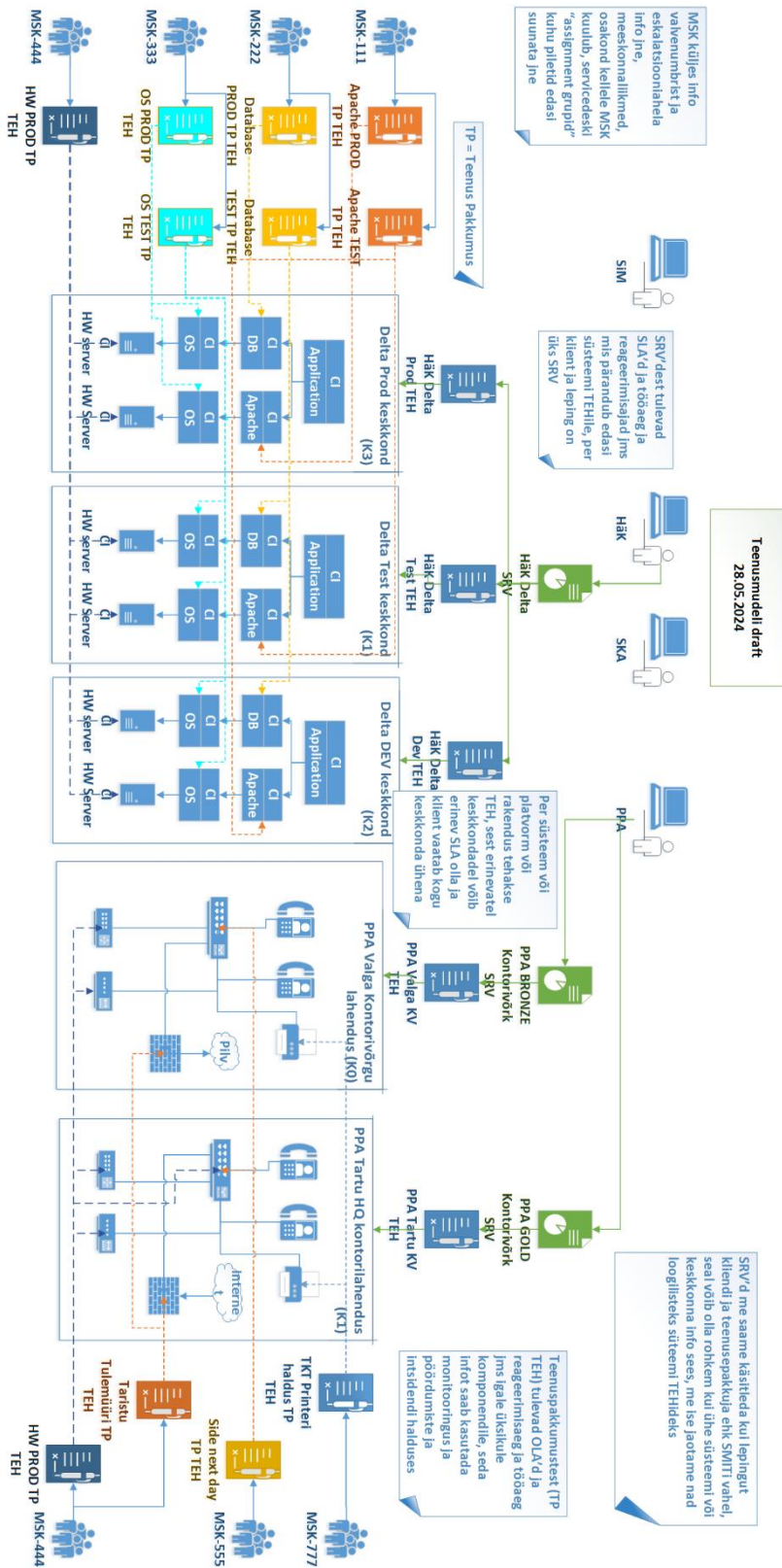
Mina, Tarvo Kesküla

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „IT organisatsiooni teenusemudeli ja CMDB analüüs“, mille juhendajad on Siim Vene ja Pirgit Kahro.
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

03.01.2025

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

Lisa 2 – Teenusumudeli näidis joonis



Joonis 3. Teenusumudeli näidis