

# **Konfiguraation hallinnan tietokanta**

## **Configuration Management DataBase (CMDB)**

Juri Seppälä

Opinnäytetyö

Huhtikuu 2017

Tieto- ja viestintäteknikan ala

Insinööri (AMK), Tietotekniikan koulutusohjelma

Tietoverkkotekniikka

Tekijä(t) Seppälä, Juri	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Huhtikuu 2017
	Sivumäärä 83	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: x
Työn nimi <b>Konfiguraation hallinnan tietokanta</b> Configuration management database (CMDB)		
Tutkinto-ohjelma Insinööri (AMK), tietotekniikan (tietoverkkotekniikan) koulutusohjelma		
Työn ohjaaja(t) Antti Häkkinen, Mika Rantonen		
Toimeksiantaja(t) Protacn Solutions Oy		
Tiivistelmä <p>Työssä konfiguroitiin ja otettiin käyttöön konfiguraation hallinnan järjestelmä iTop. Konfiguraation hallinnan järjestelmää ylläpidetään alustavasti synkronoimalla tietoa muista järjestelmistä, muun muassa ValueFrame. ValueFrame-järjestelmästä synkronoidaan palvelimet, projektit, osaprojektit ja asiakkaat konfiguraation hallinnan järjestelmään.</p> <p>Konfiguraation hallinnan rinnalle otettiin käyttöön IP-osoitteiden hallintajärjestelmä phpl-PAM. Tähän järjestelmään tuotiin alustavasti olemassa olevasta järjestelmästä palvelimien IP-osoitteet ja dokumenteista tuotiin VLAN-tiedot. Tästä järjestelmästä tehtiin synkronointi konfiguraation hallinnan järjestelmään, jossa synkronoidaan palvelimille IP-osoitteet ja VLAN-tiedot.</p> <p>Lisäksi työssä luotiin synkronointityökalu konfiguraation hallinnan järjestelmälle. Työkalun avulla synkronoidaan tietoa muista järjestelmistä.</p> <p>Lopuksi otettiin käyttöön konfiguraatioiden versionhallinnan varasto. Palvelinten konfiguraatiot tuodaan tähän varastoon, jolla pidetään yllä historiatietoja palvelinten konfiguraatiomuutoksista.</p>		
Avainsanat ( <a href="#">asiasanat</a> ) cmdb, itop, ipam, phpipam, synkronointi, repositorio, ip, vlan, konfiguraatio		
Muut tiedot		

Author(s) Seppälä, Juri	Type of publication Bachelor's thesis	Date April 2017 Language of publication: Finnish
	Number of pages 83	Permission for web publication: x
Title of publication Configuration management database (CMDB)		
Degree programme Degree Programme in Information Technology (Data Network Technology)		
Supervisor(s) Antti Häkkinen, Mika Rantonen		
Assigned by Protacon Solutions Oy		
Abstract <p>Configuration management system iTop was created. Configuration management system is maintained initially by synchronizing data from another systems e.g ValueFrame. From ValueFrame system servers, projects, subprojects and customers are synchronized to the configuration management system.</p> <p>Alongside the configuration system was created ip address management system phpIPAM. Server ip addresses were imported initially from the existing system and VLAN information from documents. From this system synchronization was made where server ip addresses and VLAN informations are synchronized to the configuration management system.</p> <p>Self-made synchronization tools were made to the configuration management system which is used for data synchronization from other systems.</p> <p>In addition, configuration version control repository was created. Servers configurations are transferred to this repository. Which maintains a history of servers configurations.</p>		
Keywords/tags ( <a href="#">subjects</a> ) cmdb, itop, ipam, phpipam, synchronization, repository, ip, vlan, configuration		
Miscellaneous		

## Sisältö

<b>Termit ja lyhenteet .....</b>	<b>4</b>
<b>1 Johdanto .....</b>	<b>7</b>
1.1 Toimeksiantajan esittely.....	7
1.2 Työn kuvaus.....	7
<b>2 Teoria.....</b>	<b>9</b>
2.1 REST .....	9
2.2 Ohjelmistoagentti.....	10
2.3 ETL .....	11
2.4 ValueFrame .....	12
2.5 Apache http-palvelin .....	13
2.6 Python .....	14
2.7 phpIPAM.....	14
2.8 Ansible .....	15
<b>3 CMDB.....</b>	<b>15</b>
3.1 Mitä CMDB on? .....	15
3.2 Configuration Item .....	16
3.3 CMDB:n hyödyt .....	16
3.4 CMDB:n haasteet.....	17
<b>4 CDMB-järjestelmältä vaadittavat ominaisuudet .....</b>	<b>18</b>
4.1 Työn tarkempi läpikäynti.....	18
4.2 CMDB:n toiminnallisuus .....	19
4.3 CMDB:n käytettävyys .....	19
<b>5 Tuotteet .....</b>	<b>20</b>
5.1 CMDBuild.....	21
5.2 ITop.....	23
5.3 Tuotteen valinta .....	25
<b>6 Työn toteutus .....</b>	<b>26</b>
6.1 Konfiguraation osien tuonti konfiguraation hallinnan tietokantaan .....	26
6.1.1 REST-rajapinnan hyödyntäminen .....	26
6.2 Konfiguraation hallinnan järjestelmä .....	27
6.2.1 iTop-konfiguraation hallinnan asennus.....	30
6.2.2 ITop-kustomisoidut moduulit .....	30
6.3 IPAM .....	32
6.3.1 phpIPAM:n asennus.....	32
6.4 Järjestelmien synkronointi .....	33
6.4.1 iTop ja ValueFrame .....	33

6.4.2	iTop ja phpIPAM .....	33
6.4.3	iTop ja vSphere .....	33
6.5	Konfiguraatioiden versionhallinta repositorio .....	34
6.5.1	Konfiguraatioiden versionhallinnan repositorian asennus .....	35
<b>7</b>	<b>Testaus</b> .....	<b>35</b>
7.1	CMDB.....	35
7.2	CMDB:n ja ValueFramen synkronointi .....	38
7.3	CMDB:n ja VMwaren synkronointi.....	38
7.4	CMDB:n ja phpIPAM:n synkronointi .....	38
7.5	PhpIPAM.....	38
7.6	Konfiguraatioiden versionhallinnan -repositorio .....	39
<b>8</b>	<b>Kokonaisuuden kuva</b> .....	<b>39</b>
<b>9</b>	<b>Pohdinta</b> .....	<b>41</b>
	<b>Lähteet</b> .....	<b>43</b>
	<b>Liitteet</b> .....	<b>45</b>
	Liite 1. ITop integraatioiden config.ini .....	45
	Liite 2. ITop synkronisoinnin vally2itoptool.py koodi .....	46
	Liite 3. ITop synkronisoinnin vally2itopsync.py koodi .....	53
	Liite 4. ITop synkronisoinnin phpipam2itoptool.py koodi .....	58
	Liite 5. ITop synkronisoinnin phpipam2itopsync.py koodi.....	62
	Liite 6. ITop asennus.....	66
	Liite 7. Osaprojekti moduulin datamodel.add_subproject_class .....	73
	Liite 8. PhpIPAM asennus.....	79

## Kuviot

Kuvio 1. Protacon Groupin virallinen logo (Media N.d.) .....	7
Kuvio 2. CMDBuild logo (CMDBuild press kit N.d.) .....	21
Kuvio 3. CMDBuild-käyttöliittymän demoversio. (CMDBuild demo: assets list N.d.)..	23
Kuvio 4. Combodo yrityksen logo (Who we are N.d.) .....	24
Kuvio 5. ITop-järjestelmän kansio- ja tiedostorakenne (iTop Customization 2016)....	28
Kuvio 6. iTop-moduulin luominen ohjatulla toiminnolla (iTop Customization 2016)..	29
Kuvio 7. iTop-käyttöliittymän näkymä .....	36
Kuvio 8. Hypervisor-komponentin vaikutusanalyysi graafisena esityksenä .....	37
Kuvio 9. Hypervisor-komponentin vaikutusanalyysi listamuodossa.....	37
Kuvio 10. Kokonaisuuden kuva .....	40
Kuvio 11. ITop ohjattu toiminto, edellytyksien tarkastus .....	68
Kuvio 12. ITop ohjattu toiminto, uusi asennus .....	68
Kuvio 13. ITop ohjattu toiminto, lisenssien hyväksyntä .....	68
Kuvio 14. ITop ohjattu toiminto, tietokantayhteys.....	69
Kuvio 15. ITop ohjattu toiminto, järjestelmänvalvoja tili.....	69
Kuvio 16. ITop ohjattu toiminto, lisävalinnat.....	70
Kuvio 17. ITop ohjattu toiminto, asennettavat iTop moduulit .....	70
Kuvio 18. ITop ohjattu toiminto, palvelunhallinta .....	71
Kuvio 19. ITop ohjattu toiminto, tiketinhallinta.....	71
Kuvio 20. ITop ohjattu toiminto, muutoksenhallinta.....	72
Kuvio 21. ITop ohjattu toiminto, lisä ITIL valinnat .....	72

## Taulukot

Taulukko 1. ITop-palvelin laitteiston vaatimukset (Installing iTop 2016) .....	25
--	----

## Termit ja lyhenteet

<b>Ajax</b>	Asynchronous JavaScript And XML. Sisältää tekniikoita verkkosivujen luontiin, jotka parantavat sivustojen käytettävyyden sulavuutta.
<b>API</b>	Ohjelmointirajapinta, jonka myötä ohjelmat voivat vaihtaa tietoja ja keskustella keskenään.
<b>Avoin lähdekoodi</b>	Eng. Open Source. Yleisesti saatavilla ja muokattavissa oleva ohjelman lähdekoodi.
<b>Backend</b>	Backend on tietotekniikan käsite, yleensä kattaa kolme osaa: palvelimen, sovelluksen ja tietokannan. Se osa, joka hakee, lukee, tallentaa ja käsittelee informaation.
<b>CI</b>	Configuration Items. Konfiguraation hallinnan objekti, joka voi olla muun muassa laite, sovellus, henkilö, dokumentti tai suunnitelma.
<b>CMDB</b>	Eng. Configuration Management Database. Konfiguraation hallinnan tietokanta.
<b>CMS</b>	Eng. Configuration Management System. Sisällönhallintajärjestelmä.
<b>cron</b>	Työkalu Linux-käyttöjärjestelmässä, jolla voidaan ajaa komentoja ajoitetusti.
<b>CSS3</b>	Ohjelmointikieli, joka kuvaa HTML-dokumenttien tyyliä.
<b>ERP</b>	Enterprise Resource Planning, toiminnanohjausjärjestelmä.
<b>ETL</b>	Extract-Transform-Load.
<b>FQDN</b>	Fully qualified domain name. Täydellinen koneen nimi. Sisältää kaksi osaa: koneen nimi ja toimialueen nimi esimerkiksi server.domain.com.
<b>Git</b>	Versionhallintatyökalu, jolla seurataan tiedostomuutoksia.
<b>HTML</b>	Hypertext Markup Language. Yleisesti käytetty kuvauskieli verkkosivujen tekemiseen.
<b>HTML5</b>	HTML-kuvauskielen uusi versio.
<b>HTTP</b>	Hypertext Transfer Protocol. Hypertekstin siirtoprotokolla, jota selaimet ja www-palvelimet käyttävät tiedosiirtoon.
<b>Hypervisor</b>	Sovellus, laitesovellus tai rauta, joka luo ja ajaa virtuaalikoneita. Koneita, jolla ajetaan yhtä tai useampaa virtuaalikonetta, kutsutaan isäntälaitteeksi ja sen virtuaalikoneita vieraslaitteiksi.

<b>IP</b>	Internet Protocol. IP huolehtii pakettien kuljettamisesta lähteestä kohteeseen.
<b>IPv4</b>	Internet Protocol version 4.
<b>IPv6</b>	Internet Protocol version 6.
<b>ITIL</b>	Information Technology Infrastructure Library.
<b>JavaScript</b>	Skriptikieli verkkosivujen dynaamisten toiminnallisuuksien lisäämiseen.
<b>jQuery</b>	JavaScript-ohjelmointikirjasto, yksinkertaistaa JavaScript ohjelmointia.
<b>JSON</b>	JavaScript Object Notation. Formaatti, jolla tieto saadaan ihmisen ja tietokoneen ymmärtämään muotoon.
<b>MySQL</b>	Avoimen lähdekoodin tietokantasovellus.
<b>NCSA</b>	National Center for Supercomputing Applications.
<b>Ohjelmistokehys</b>	Eng. software framework. Ohjelmistotuote, joka muodostaa kehyksen sen päälle rakennettavalle ohjelmistolle.
<b>Ohjelmoitikirjasto</b>	Sisältää valmista koodia, jonka käyttäminen helpottaa ohjelmointia.
<b>PHP</b>	Ohjelmointikieli, eng Hypertext Preprocessor.
<b>PSA</b>	Professional Services Automation, asiantuntijayrityksen toiminnanohjaus.
<b>Python</b>	Ohjelmointikieli.
<b>Query</b>	Tietotekniikassa query-sana viittaa tiedon kyselyyn tai hakuun tietokannoista ja järjestelmistä.
<b>REST</b>	Representational State Transfer. Ohjelmallinen rajapinta, jolla saadaan vietyä ja tuotua tietoa esim XML- tai JSON-muodossa.
<b>RESTful</b>	Järjestelmä, joka käyttää REST-arkkitehtuuria.
<b>RPC</b>	Remote Procedure Calls.
<b>SELinux</b>	Security-Enhanced Linux, Linux-ytimeen laajennus, joka tarjoaa lisäominaisuudet järjestelmän tietoturvaan.
<b>SOAP</b>	eng. Simple Object Access Protocol. XML-pohjainen viestintäprotokolla tiedon välitykseen koneiden välillä.
<b>SourceFroge</b>	Verkkosivusto, jonka kautta voidaan jakaa avoimenlähdekoodin projekteja.
<b>URI</b>	Uniform Resource Identifier, tietotekniikassa tarkoittaa merkkijonoa, jolla ilmaistaan tiedon sijainti. Tunnetuin



	URI-muoto on URL, joka tunnetaan yleisimmin verkko-osoitteena.
<b>Web</b>	World Wide Web.
<b>VLAN</b>	Virtual VLAN. Virtuaalinen lähiverkko.
<b>VMware</b>	Sovellus, jolla voidaan luoda ja hallita virtuaalikoneita.
<b>VRF</b>	Virtual Routing and Forwardin. Mahdollistaa usean reititystaulun käytön samanaikaisesti.
<b>WSDL</b>	Web Service Description Language.
<b>XML</b>	Extensible Markup Language. Formaatti, jolla tieto saadaan ihmisen ja tietokoneen ymmärtämään muotoon.
<b>Zip</b>	Tiedonpakkausmenetelmä, jonka tiedostopääte on .zip.

# 1 Johdanto

## 1.1 Toimeksiantajan esittely

Toimeksiantajana opinnäytetyölle toimi Protacón Solutions Oy, joka on osa Protacón Group -konsernia. Protacón Group on suomalainen ja kansainvälistyvä, vuonna 1990 perustettu teknologia-alan suunnittelu- ja palveluyritys, joka tuottaa prosessi-, auto- maatio- ja ICT-ratkaisuja projektointiin, tuotannon ja ylläpidon tarpeisiin. Palveluvali- koima sisältää esimerkiksi testaus- ja laadunvarmistusjärjestelmiä, ohjelmistotuottei- ta sekä teknologian avaimet käteen -ratkaisuja. (Me olemme Protacón N.d.)

Protacón Group Oy koostuu neljästä tulosvastuullisesta osakeyhtiöstä. Nämä osake- yhtiöt ovat nimeltään Protacón Group Oy, Protacón Analyzes Oy, Protacón Technolo- gies Oy ja Protacón Solutions Oy. Kuviossa 1 on Protacón Groupin virallinen logo.

Jyväskylässä päätoimipaikkaansa pitävällä suunnittelu- ja palvelukonsernilla on toi- mipisteitä yhdellätoista paikkakunnalla Suomessa ja yksi Kiinassa. Yrityksen liikevaih- to on noin 20 miljoonaa euroa ja henkilöstömäärä yli 250. (Protacón Group LinkedIn N.d.)



Kuvio 1. Protacón Groupin virallinen logo (Media N.d.)

## 1.2 Työn kuvaus

Opinnäytetyön aihe oli konfiguraation hallinnan tietokanta (CMDB). CMDB-lyhenne tarkoittaa termiä configuration management database. Työnantajan toimeksianto oli toteuttaa järjestelmä, johon tuodaan organisaation komponentteja, kuten palveli-

met, projektit, osaprojektit, asiakkaat ja tulevaisuudessa mahdollisesti muita, jotka kuuluvat jatkokehitykseen. Kyseisen järjestelmän tarkoituksena on koostaa tuotujen organisaation komponenteille palvelupuut. Palvelupuun avulla saadaan selville, esimerkiksi mihin kaikkeen palvelin vaikuttaa ja luotua tästä raportti. Tästä raportista voidaan helposti nähdä, mihin asiakkaisiin mahdollinen palvelukatkos vaikuttaa ja keiden projektipäälliköiden tulisi tietää asiasta.

Tarkempaa määrittystä sen suhteen miten tai mistä kyseisen järjestelmän tiedot tul-taisiin kokoamaan ei määritetty. Tällä hetkellä Protaconilla on eri järjestelmiä, joissa näitä tietoja pidetään yllä ja käytetään. Tulevan CMDB-järjestelmän tulisi kerätä ja ylläpitää tietoja automaattisesti.

Toimeksiannossa tuli esille, CMDB-järjestelmä voisi olla myös maksullinen, jos kulut pysyvät järkevänä ja sen ominaisuudet on perusteltu. Lähtökohtaisesti on löytää avoimen lähdekoodin toteutus.

Toisena osana tehtävänantoa olisi toteuttaa konfiguraatioiden versionhallinta. Tällä tarkoitetaan järjestelmää, johon tuodaan alustavasti palvelimien konfiguraatiot ja järjestelmä pitää historiatietoja konfiguraatiomuutoksista. Palvelimilla tarkastettaisiin konfiguraatiot päivittäin, jos on tapahtunut muutosta, viedään tieto tähän järjestelmää. Tämän avulla voitaisiin mahdollisessa palvelimen ongelmatilanteessa selvittää, johtuuko se konfiguraation tehdystä muutoksesta ja millainen konfiguraatio oli, kun se vielä toimi.

Kolmas osa tehtävänantoa on toteuttaa, joko CMDB-järjestelmään, jos siitä löytyy tämä ominaisuus valmiina tai sen rinnalle IP-osoitteiden hallinta (IPAM). Tällä järjestelmällä olisi tarkoitus hallita ja ylläpitää IP-osoitteita. IPAM-järjestelmästä tul-taisiin viemään IP-osoitteet CMDB-järjestelmässä niille kuuluville palvelimille.

## 2 Teoria

### 2.1 REST

REST-lyhenne tarkoittaa termiä REpresentational State Transfer. Se on arkkitehtuurimalli tietoverkon sovelluksille HTTP-protokollan yli ja se on suosittu verkkosovellusten kehityksessä. REST:n avulla sovellukset voivat keskustella keskenään. REST esiteltiin ensimmäistä kertaa vuonna 2000 Roy Fielding:n toimesta, hänen väitöskirjassaan. (What is REST API N.d.)

REST ei määrittele mitään sääntöjä, miten se tulisi toteuttaa alemmalla tasolla, vaan se ainoastaan asettaa korkean tason ohjeistukset ja jättää toteutuksen täytäntöönpanon sen tekijälle. (What is REST API N.d.)

Kuten millä tahansa muulla arkkitehtuurisella mallilla, REST:illä on myös sen omat kuusi ohjaavaa rajoitetta, jotka tulisi täyttää, jotta rajapintaa voidaan pitää oikeana REST -rajapintana. (What is REST API N.d.)

Nämä kuusi ohjaavaa rajoitetta ovat seuraavat:

1. Client-server (suom. asiakas-palvelin), asiakassovellus ja palvelin on erotettu toisistaan, niiden tulee olla riippumattomia toisistaan. Näin ne voivat kehittyä vaikuttamatta toisiinsa. Asiakassovelluksen tulee ainoastaan tietää resurssin (palvelimen) verkko-osoitteen (URI). (What is REST API N.d.)
2. Stateless, jokainen pyyntö asiakassovellukselta palvelimelle tulee sisältää kaikki tarvittava tieto, jolla pyyntö voidaan käsitellä. Palvelin ei säilytä mitään tietoa aikaisemmista asiakassovelluksen http-pyyntöistä. Palvelin käsittelee jokaisen pyynnön uutena ilman historiatietoja. (What is REST API N.d.)
3. Cacheable (suom. välimuistiin), välimuistin käyttö, joka ilmaistaan yksiselitteisesti, onko käytössä vai ei. Käyttöönotto voidaan ilmaista palvelimen tai asiakassovelluksen puolella. Välimuistin käyttäminen parantaa suorituskykyä asiakassovelluksen puolella ja vähentää palvelimen kuormaa. (What is REST API N.d.)

4. Uniform interface (suom. yhtenäinen käyttöliittymä), palvelinpuolella täytyy päättää API-rajapinta, joka tullaan paljastamaan verkkoon asiakasovelluksille. Järjestelmässä resursilla tulisi olla vain yksi URI, jonka tulisi tarjota tavat hakea siihen liittyviä tietoja. (What is REST API N.d.)
5. Layered system (suom. kerroksellinen järjestelmä), kerroksellinen järjestelmä mahdollistaa arkkitehtuurin muodostamisen hierarkisten tasojen kautta. Komponenttien käyttäytymistä rajoitetaan siten, että ne eivät "näe" välittömän kerroksen yli, jonka kanssa ne ovat vuorovaikutuksessa. (What is REST API N.d.)
6. Code on demand (optional), tämä rajoite on valinnainen. Yleensä lähetetään sen hetkinen tilanne tiedosta, joko XML- tai JSON-muodossa. Tarvittaessa voidaan palauttaa suoritettava koodi, tukemaan sovellusta, esimerkiksi jolla laajennetaan sovelluksen toiminnallisuutta. (What is REST API N.d.)

REST:ssä tiedon erottaminen on resurssi. Kaikki tieto, joka voidaan nimetä, on resurssi, kuten dokumentti tai kuva, väliaikainen palvelu ja joukko muita resursseja. REST käyttää resurssitunnistetta tunnistamaan tietyn resurssin olevan osallisena komponenttien vuorovaikutuksessa. (What is REST API N.d.)

REST:n olennainen osa on myös resurssin menetelmät, joita käytetään saadakseen haluttu toimenpide. Menetelmillä tarkoitetaan, halutaanko hakea, muuttaa, lisätä tai poistaa tietoa. Usein virheellisesti resurssimenetelmillä viitataan HTTP GET/PUT/POST/DELETE. Roy Fielding ei missään vaiheessa maininnut mitään suositusta mitä toimenpidettä tulisi käyttää missäkin tilanteessa. Jos halutaan käyttää HTTP POST-toimenpidettä tietojen päivittämiseen HTTP PUT-toimenpiteen sijaan, se on hyväksyttävää, kun sovelluksen rajapinta on RESTful. (What is REST API N.d.)

## 2.2 Ohjelmistoagentti

Ohjelmistoagenttia voidaan pitää itsenäisenä ongelmanratkaisijana ja päämäärätietoisena ohjelmallisena osana, joka pystyy tehokkaasti, ehkä jopa ennakoivasti toimi-

maan avoimessa ja dynaamisessa ympäristössä. Ohjelmistoagentti seuraa tilannetta ja toimii tilanteen mukaan sille määritetyllä tavalla saavuttaakseen sen tavoitteensa. (Leitão, Karnouskos 2015, luku 1.2)

Kehittyneemmän agentin ominaisuuksia on muun muassa:

- **autonomia:** itsenäinen toiminta ilman ulkopuolista vaikutusta.
- **reagointikyky ja tilannetietoisuus:** agentti seuraa ympäristönsä tilannetta ja toimii sen mukaan.
- **proaktiivisuus:** agentti ei toimi vain reakoiden vaan myös ennakoiden.
- **vuorovaikutus:** agentti kuuntelee ja keskustelee ulkopuolisen järjestelmän kanssa ja toimii sen mukaan. (Leitão, Karnouskos 2015, luku 1.2)

## 2.3 ETL

ETL-lyhenne tarkoittaa termiä Extract-Transform-Load (suom. kerää, siirrä ja syötä). ETL-termillä viitataan datan keräämiseen, datan siirtämiseen kohteeseen ja sen syöttämiseen tietokantaan. ETL sisältää laajaan määrään eri vaiheita, eikä vain kolmea tarkasti määriteltyä vaihetta. (Overview of Extraction, Transformation, and Loading N.d.)

ETL-menetelmät ja -vaiheet ovat olleet tunnettuja jo useiden vuosien ajan. Dataa täytyy siirtää sovellusten tai järjestelmien välillä, integroiden, jotta ainakin kaksi järjestelmää sisältää samat tiedot. (Overview of Extraction, Transformation, and Loading N.d.)

Keräysvaihe kattaa datan keräämisen lähdejärjestelmästä ja valmistele sen myöhempää käsittelyä varten. Keräysvaiheen päätarkoitus on kerätä kaikki tarvittava data lähteestä mahdollisimman pienillä resursseilla. Keräysvaihe tulisi toteuttaa siten, että se ei vaikuta negatiivisesti tuotannon suorituskykyyn. Keräysvaiheeseen on useimpia eri tapoja. Kolme esimerkkiä ovat:

- **päivitysilmoitus:** jos lähdejärjestelmä kykenee ilmoittamaan, että muutos tietueessa on tapahtunut ja kuvailemaan tietueen muutoksen, se olisi helppoin tapa.
- **inkrementaalinen:** vähittäin kasvava keräys, jotkin järjestelmät eivät voi lähettää ilmoitusta kerralla, mutta voivat tunnistaa mitkä tietueet ovat muuttuneet ja tarjoavat kyseisille tietueille keräyksen.
- **Täysi keräys:** jotkin järjestelmät eivät voi lähettää ilmoitusta muutoksista tai tunnistaa, mikä on muuttunut. Tällöin täysi keräys on ainoa toteutusvaihtoehto, jolla data kerätään järjestelmästä. (ETL (Extract-Transform-Load) N.d.)

Siirtovaihe kattaa vaiheet, joilla saadaan data siirrettyä lähteestä kohteeseen. Tähän vaiheeseen kuuluu datan käsittely sellaiseen muotoon, jotta sitä voidaan käyttää kohteessa. Siirtovaiheeseen kuuluu myös datan kerääminen useammasta lähteestä ja sen käsittely myöhempää käyttöä varten. (ETL (Extract-Transform-Load) N.d.)

Syöttövaihe kuvaa datan syöttämisen kohdejärjestelmään. Tämän vaiheen tulisi myös käyttää mahdollisimman vähän resursseja. (ETL (Extract-Transform-Load) N.d.)

CMDB-järjestelmän kanssa ETL-työkalu tulee olennaiseksi, jos sillä halutaan käyttää muiden järjestelmien sisältämää dataa. Samalla datan integroinnin vaiheet tulisi olla automaattisia.

## 2.4 ValueFrame

Protacon-konsernin käyttämä toiminnanohjausjärjestelmä on nimeltään Valueframe PSA Pro. Valueframe PSA Pro on toiminnanohjausjärjestelmä (ERP), joka tarjoaa työkalut asiantuntijayrityksien päivittäiseen käyttöön. Sen tarjoama ERP-järjestelmä koostuu yhteen ratkaisuun kaiken yrityksen johtamiseen tarvittavan keskeisen tiedon. ValueFrame auttaa näkemään reaaliaikaisesti ajasta ja paikasta riippumatta, mitä organisaatiossa tapahtuu sekä miksi ja miten toimintaa tulee suunnata. (Toiminnanohjausjärjestelmä N.d.)

ValueFrame PSA Pro kokoaa yhteen organisaation taloudellisen informaation, projektien- ja resurssienhallinnan, työajanseurannan sekä asiakkaisiin ja henkilöstöön liittyvät tiedot. Johdon raportoinnin avulla voidaan tehdä oikeita päätöksiä liiketoiminnan kasvattamiseksi ja kannattavuuden parantamiseksi. Liiketoiminta tehostuu yhdenmukaisten toimintatapojen ja sähköisten toimintaprosessien ansiosta. (Toiminnanohjausjärjestelmä N.d.)

ValueFrame-järjestelmällä organisaatiossa hallitaan toiminnanohjausta, laite-, projekti-, osaprojekti- ja asiakaskantaa.

ValueFrame-tietoihin päästään suoraan REST:n kautta käsiksi ilman erillistä ohjelmointia (ns. taulupohjainen REST), mutta monimutkaisemmissa tilanteissa ja erityisesti jos tietoja täytyy muokata, ohjelmointityö voi olla tarpeen. Tällöin yksittäisten tietokantataulujen sijaan rajapinta voidaan ohjelmoida (niin sanottu luokkapohjainen REST) taulumäärittysten sijaan. (ValueFrame lisäpalvelut REST N.d.)

ValueFramen REST-rajapinnasta löytyy valmiita malleja molempiin edellä mainittuihin tilanteisiin. Kutsuvan ohjelman näkökulmasta käyttö on molemmissa tilanteissa samanlaista: [https://url/rest/v2/{REST\\_resurssi}/](https://url/rest/v2/{REST_resurssi}/). (ValueFrame lisäpalvelut REST N.d.)

## 2.5 Apache http-palvelin

Apache on avoimeen lähdekoodiin perustuva HTTP-palvelinohjelma, johon useimmiten viitataan pelkästään Apache-nimellä. Apache on voittoa tavoittelemattoman Apache Software Foundation -organisaation tunnetuin tuote. Alun perin Apache suunniteltiin korvaamaan NCSA HTTP -palvelinohjelma ja tähän päivään mennessä se on kasvanut internetin suosituimmaksi HTTP-palvelinohjelmaksi. (Apache HTTP Server 2014.)



## 2.6 Python

Python on tehokas korkea tason monikäyttöinen ohjelmointikieli, joka on Guido van Rossum:n luoma. Pythonin kehitys alkoi 1980-luvun lopulla ja julkaistiin helmikuussa vuonna 1991. Sillä on laaja valikoima sovelluksia web-kehityksessä, kuten Django ja Bottle, tieteellisestä ja matemaattisesta tietojen käsittelystä (Orange, SymPy, NumPy) aina työpöytäympäristöjen graafisiin käyttöliittymiin (Pygame, Panda3D). (Learn Python Programming N.d.)

Python-ohjelmointikielen rakenne on selkeä ja koodin pituus on suhteellisen lyhyt. Pythonin kanssa on mukava työskennellä, koska se antaa ohjelmoijan keskittyä ongelmanratkaisun selvittämiseen sen sijaan, että ohjelmoijan tarvitsisi käyttää aikaa koodin rakenteen tarkasteluun. (Learn Python Programming N.d.)

## 2.7 phpIPAM

PhpIPAM on avoimen lähdekoodin web-pohjainen IP-osoitteiden hallintasovellus (IPAM, IP address management). Sen tavoite on tarjota kevyt, moderni ja hyödyllinen IP-osoitteiden hallinta-alusta. PhpIPAM on rakennettu käyttäen PHP-ohjelmointikieltä ja se hyödyntää MySQL-tietokantaa. Lisäksi se käyttää jQuery-kirjastoja, Ajaxia ja HTML5/CSS3-ominaisuuksia. (phpIPAM N.d.)

PhpIPAM-sovelluksen tärkeimpiä ominaisuuksia ovat muun muassa:

- IPv4- ja IPv6-osoitteiden hallinta
- aliverkkojen hallinta
- aliverkkojen vapaiden IP-osoitteiden ilmaisu
- käytettyjen IP-osoitteiden löytäminen
- laitteiden saatavuuden tarkastaminen
- REST-rajapinta
- VRF- ja VLAN-hallinta. (phpIPAM N.d.)

## 2.8 Ansible

Ansible on IT-automaatiotyökalu. Sen avulla voidaan konfiguroida järjestelmiä, asentaa sovelluksia ja orkestroida vaativia IT-toimenpiteitä, kuten ohjelmistokoodin testaus ja sovelluspäivitysten asennus. (About Ansible 2017.)

Ansible:n tehtävä on olla yksinkertainen ja helppokäyttöinen. Sillä on myös vahva painopiste tietoturvassa ja luotettavuudessa, sisältäen mahdollisimman vähän liikkuvia osia. Lisäksi Ansible mahdollistaa salatun SSH-yhteyden käytön tiedonsiirrossa ja sen käyttö ei edellytä ohjelmoinnin osaamista. (About Ansible 2017.)

Ansible:lla on työkalu nimeltään playbooks. Playbooks:t ovat Ansible:n konfiguraatio, käyttöönotto ja orkestrointikieli. Perustasolla niitä voidaan käyttää koneiden konfiguraatioiden hallintaan ja käyttöönottoon. (Playbooks 2017.)

Ansible playbooks -konfiguraatio tehdään YAML-formaatissa. YAML on merkintäkieli, joka on yleisesti käytetty konfiguraatioissa. YAML-formaatin konfiguraatiota on helppo ymmärtää ja kirjoittaa.

## 3 CMDB

### 3.1 Mitä CMDB on?

CMDB-lyhenne tarkoittaa termiä Configuration Management DataBase (suom. konfiguraation hallinnan tietokanta). CMDB on alun perin tullut ITIL:stä, joka tarjoaa koelman käytäntöjä IT-palveluiden hallintaan (Gordon 2015, liite H).

Konfiguraation hallinnan tietokanta on tietokanta, joka säilyttää konfiguraation osien (CI) informaation, joita käytetään muutoksen, julkaisun, vikatilanteen, palvelupyynnön, ongelman ja konfiguraation hallinnan prosesseissa. Konfiguraation hallinnan tietokanta on oleellinen osa ITIL:iin perustuvaa palvelunhallintaa. (Configuration Management Database N.d.)

Konfiguraation hallinnan tietokanta säilyttää oleellisen informaation konfiguraation osista (CI) (esimerkiksi palvelin, projekti tai asiakas), joiden hyödyntäminen tehostaa

työ tekemistä. Esimerkkinä konfiguraation osa sisältää sen tyyppin tai luokituksen, sen sisältämät attribuutit, jotka kuvaavat konfiguraation osan erityispiirteitä ja tiedot sen yhteyksistä muihin konfiguraation osiin. (Configuration Management Database N.d.)

Pohjimmillaan konfiguraation hallinnan tehtäviä ovat konfiguraation osien määrittely ja niiden sisällyttäminen konfiguraation hallinnan tietokantaan, konfiguraation osien hallinta ja ylläpitäminen, konfiguraation osien tilan ylläpitäminen ja tietokannan sisältämän tiedon varmentaminen, jotta sen sisältämä tieto on luotettavaa. Tämän lisäksi konfiguraation hallinnan tietokannan tehtäviin kuuluu erilaisten prosessien ohjaus. Esimerkkinä muutoksen hallinta, jonka yhteydessä on tarve tehdä muutos konfiguraation osaan. tällaisessa tilanteessa voidaan tarvita ensin muutoksen määrittely, toisen tahon tarkastus ja hyväksyntä.

### 3.2 Configuration Item

Tässä yhteydessä tietojärjestelmän komponentteja kutsutaan nimellä Configuration Item (CI), suomennettuna konfiguraation osa. Konfiguraation osa voi olla mikä tahansa IT-komponentti, mukaan lukien laitteistot, ohjelmistot, dokumentoinnit, henkilöstö, asiakkaat, projektit, sekä niiden yhdistelmät. (Quigley, Robertson 2015, luku 5.2.)

Konfiguraation hallinnan prosessien tavoite on hallita ja pitää yllä konfiguraation osia siten, että niiden sisältämät tiedot ovat hyödyllisiä ja luotettavia.

### 3.3 CMDB:n hyödyt

Organisaation kasvaessa sen resurssien määrä kasvaa ja tällöin on haastavampaa pysyä mukana organisaatiossa käytettävistä komponenteista ja niiden liitoksista. Niinpä käytetyt resurssit olisi hyvä dokumentoida.

Konfiguraation hallintaa voidaan itsessään käyttää rajapintana sille, että siihen lisätään käsin tarvittavat tiedot. Siihen voidaan integroida muita järjestelmiä ja tuoda tarvittava tieto muista organisaation käyttämistä järjestelmistä. Vaihtoehtoisesti näitä molempia voidaan hyödyntää rinnakkain. Konfiguraation hallinnan avulla voidaan

luoda raportteja riskianalyseistä, voidaan varautua ongelmatilanteeseen tai ongelmatilanteessa saadaan helposti poimittua oleellinen tieto sen ratkaisemiseen.

Riippuen konfiguraation hallinnan järjestelmästä, se voi kuulua paljon erilaisia ominaisuuksia. Konfiguraation osien hallinnan lisäksi useimmissa järjestelmissä on tarjolla ominaisuudet tiketin hallinnalle. Tiketin hallinnan avulla saadaan palveltua asiakkaita ja vietyä ongelmaa eteenpäin ja ratkaistua ne. Useimmista löytyy myös muutoksen hallinnan työkalut. Varsinkin suurissa organisaatioissa ei voida perusteetta tehdä muutoksia laitteisiin. Muutokselle tarvitaan perustelut, muutos tulee määritellä ja se tulee hyväksyttävä valtuutetulla henkilöllä. Näin voidaan seurata muutoksia ja säästyään suuremmilta ongelmilta.

### 3.4 CMDB:n haasteet

Konfiguraation hallinnan järjestelmän käsitteen tarkoituksesta ja sisällöstä on useita eri näkemyksiä. Useimmiten ollaan eri mieltä, että tulisiko siitä tarpeellinen osa organisaation IT-infrastruktuuria. Tänä päivänä nousee usein esille väite, että CMDB olisi kuollut. Syynä tähän on useimmiten mahdollisesti huonot konfiguraation hallinnan toteutukset. Esimerkki huonosta toteutuksesta on sellainen, joka sisältää liikaa turhaa tietoa ja sitä ei saada hyödynnettyä tarpeeksi hyvin. Sen ylläpitäminen on haastavaa ja sen sisältämään tietoon ei voi enää luottaa. Talainen CMDB-järjestelmä jää usein käyttämättä.

Alun perin CMDB käsitettiin yhtenä tietokantana, sisältäen tiedot kokoonpanon osista (CI) ja niiden suhteista toisiinsa. Toisaalta useimmat kokeneemmat alan ammattilaiset pitävät sitä loogisena tietokantana, joka voi koostua useammasta hajautetusta fyysisestä tietokannasta, useimmat pitävät sitä yhtenä fyysisenä tietokantana. Yhden fyysisen tietokannan malli keskisuurten ja suurten yritysten kohdalla on ongelmallinen. Tällaisessa tilanteessa rajoituksena toteutukselle muodostuvat ylläpidolliset ongelmat sekä suuret hankinta- ja investointikustannukset. (Shirley, Norfolk 2014, luku 5.2.)

Usein luotetaan liikaa palveluntarjoajien tarjoamiin työkaluihin, kuten auto-discovery, jonka avulla automatisoitaisiin tietokannan luontiprosessi. Auto-discovery

työkalun tarkoituksena olisi löytää automaattisesti kokoonpanon osat organisaation verkosta, kuten esimerkiksi IT-ympäristön laitteisto. Vaikka osa auto-discovery työkaluista voivat tarjota tehokkaat tavat koota tieto konfiguraation osista, ne eivät välttämättä kerää kaikkia tarvittavia tietoja, joita tarvittaisiin hyvään konfiguraatioiden hallintaan ja jotta sitä kyettäisiin hyödyntämään tehokkaasti. Tämän lisäksi näillä työkaluilla kerätään usein liikaa turhaa tietoa. Monet CMDB-toteutukset epäonnistuvat tämän vuoksi, kerätään liikaa turhaa tietoa ja konfiguraation hallinnan tietokannan ylläpitäminen ei toimi. Näin konfiguraation hallinnan tietokannan suunnitteluun ja toteuttamiseen käytetty raha ja aika on tuhlatu. (Shirley, Norfolk 2014, luku 5.2.)

## **4 CDDB-järjestelmältä vaadittavat ominaisuudet**

### **4.1 Työn tarkempi läpikäynti**

Työ tavoitteena on luoda käyttöliittymä, jonka avulla sen käyttäjä pystyy helposti koostamaan vikatilannetta koskevan vika-analyysin (eng. impact analysis). Esimerkkinä tästä voisi olla asiakaslistan koostaminen palvelimen vikaantuessa niistä asiakkaisista, joihin vika vaikuttaa. Tämän analyysin avulla voidaan helposti ilmoittaa tarvittaville tahoille palvelun poikkeustilanteesta.

CMDB-kantaan tallennetut laitteisto- projekti ja asiakastiedot on tarkoitus noutaa yrityksen olemassa olevasta järjestelmästä. Tietojen haku toteutetaan REST-rajapintaa hyödyntäen. Haetun tiedon lisääminen tehdään CMDB-järjestelmän työkalua käyttäen. Tiedot haetaan kerran päivässä ja päivitetään CMDB-tietokantaa automaattisesti. Tämän toiminnon toteuttaminen edellyttää oman ohjelmakoodin luontia.

CMDB-tietokantaan tallennettujen osien (palvelimet, projektit, asiakkaat) välille luodaan palvelupuu. Tällä tarkoitetaan osien välisiä liitoksia, jotka kuvaavat osien keskinäisiä vaikutussuhteita. Palvelupuun rakentuminen CMDB-järjestelmään tapahtuisi myös automaattisesti hyödyntäen yrityksen olemassa olevaa järjestelmää. Myös tämän toiminnon automatisointi edellyttää oman ohjelmakoodin luontia.

CMDB-järjestelmän rinnalle luodaan konfiguraatioiden versionhallinnan järjestelmä, jonka tulisi sisältää historiatiedot palvelimille tehdyistä konfiguraatiomuutoksista. Palvelimilta haetaan ensin alustavasti konfiguraatiot ja sen jälkeen niihin tehdyt muutokset. Palvelimille voidaan luoda oma agentti, joka tarvittaessa lähettää muuttuneen konfiguraatiotiedoston. Vaihtoehtoinen toteutusmalli olisi, että palvelimet lähettävät konfiguraatioiden muutostiedot tietyn ajan välein. Tämä järjestelmä integroidaan CMDB-järjestelmään.

CMDB-järjestelmän tulisi toimia mahdollisimman automaattisesti, koska sen käsin ylläpitäminen tulisi olemaan raskasta ja on todella todennäköistä, että sen sisältämä tieto ei olisi ajanmukaista ja luotettavaa.

## 4.2 CMDB:n toiminnallisuus

Tiedon lisäämiseen CMDB-tietokantaan on useimpia tapoja, joita ovat muun muassa organisaation verkon aktiivinen skannaaminen tai organisaation käyttämän järjestelmän REST-rajapinnan käyttäminen.

Organisaation verkon aktiivinen skannaaminen tulisi vaatia oman skannausjärjestelmän tai mahdollisesti agent-skannaussovelluksen asentamisen asiakaslaitteille.

REST-rajapinnalla tietoja saadaan tuotua yrityksen järjestelmästä yksinkertaisessa JSON-muodossa. CMDB-järjestelmä rakennetaan sopivaksi, jotta saadaan tuotua haluttu tieto.

## 4.3 CMDB:n käytettävyys

Tarvittaville tahoille mahdollistetaan pääsy CMDB-käyttöliittymään ja näin mahdollistetaan tarvittavien tietojen hakeminen. Tästä esimerkkinä voisi olla laitteen palvelupuun esittäminen ja laitteen vaikutussuhteiden määrittäminen esitetyn palvelupuun pohjalta.

Kyseisen käyttöliittymän tulisi myös mahdollisesti toimia keskipisteenä tarvittaville toiminnoille, joista päästään muihin palveluihin, kuten laitteiden konfiguraatioiden tarkasteluun ja IP-osoitteiden hallintaan (IPAM).

## 5 Tuotteet

CMDB-järjestelmän toteutusta varten verrattiin soveltuvia tuotteita, joilla konfiguraation hallinta voidaan toteuttaa ja jotka täyttävät työssä asetetut tavoitteet. Tuotteen avulla tuodaan ja ylläpidetään tietoa tarvittavista organisaation konfiguraation osista ja niiden riippuvuuksista toisiinsa. Toinen vaadittava ominaisuus oli konfiguraatioiden master-laitteen käyttöönotto. Tämä mahdollistaa laitteiden konfiguraatioiden muutoshistorian hallinnan. Työhön haettiin ensisijaisesti avoimen lähdekoodin ratkaisua (eng. open source), mutta otettiin myös huomioon maksulliset järjestelmät niissä tapauksissa, joissa kulut ovat järkevät ja perustellut.

Maksullisia CMDB-järjestelmiä tutkittaessa tuli huomattua, että ne kattavat todella paljon. Maksulliset järjestelmät sisältävät kokonaisuudessaan muun muassa tiketöintijärjestelmän, toiminnanohjauksen, muutoksen hallinnan, dokumentin hallinnan ja useita työkaluja, esimerkiksi verkon laitteiden tietojen automaattiseen hakemiseen ja keräämiseen. Tästä taas seuraa se, että järjestelmän kokonaisuuden hinta nousee liian korkeaksi tähän tarkoitukseen. Tämän vuoksi maksullisia järjestelmiä tutkiessa otettiin huomioon, ovatko ne modulaarisia, saisiko laiteinventaarion palvelupuu ominaisuudella omana osanaan.

Konfiguraatioiden hallinta on laaja käsite, maksullisiin konfiguraation hallinnan tietokannan järjestelmiin on koottu todella paljon ominaisuuksia, joiden ratkaisut korvaisivat kokoonpanollaan tällä hetkellä Protacoinin käyttämän järjestelmän ja tähän ei ole tällä hetkellä tarvetta. Myöskään maksullisia toteutuksia tutkiessa ei tullut eteen, että niiden ominaisuuksien pilkkominen olisi mahdollista ja että niiden osalta voitaisiin ottaa käyttöön vain haluttu toiminnallisuus. Tämän vuoksi päädyttiin avoimen lähdekoodin toteutukseen.

## 5.1 CMDBuild

CMDBuild-projekti alkoi vuonna 2005. Kuviossa 2 on esiteltyä CMDBuild virallinen logo. Projektin tarkoituksena oli toteuttaa konfiguraation hallinnan tietokantajärjestelmä (CMDB) noudattaen ITIL-ohjeistuksia ja pidemmällä tähtäimellä saada aikaiseksi täydellinen ja yhtenäinen ympäristö objektien hallinnan sovelluksien konfigurointiin. (The CMDBuild Project N.d.)



Kuvio 2. CMDBuild logo (CMDBuild press kit N.d.)

CMDBuild on avoimen lähdekoodin web-sovellusratkaisu, joka on suunniteltu ja kehitetty Tecnotecan puolesta IT-ympäristön johtamisen hallintaan (CMDBuild ready2use N.d.). CMDBuild tuotteella on latauksia yli 30000 vuodessa, se on käännetty 19 kielelle ja sitä kehittää yli 30 täyspäiväistä työntekijää. CMDBuild on toteutettu Java-ohjelmointi kielellä ja muita sen käyttämiä teknologioita ovat PostgreSQL, web, Ajax ja SOA. (CMDB press kit.) Sen kehitys on pysynyt tähän päivään mennessä aktiivisena ja uusin versio 2.4.2 julkaistiin 18 elokuuta 2016.

CMDBuild:sta on myös saatavilla "maksullinen" versio, joka kattaa tuen Tecnotecan puolesta oman ympäristön perustamiseen. Lisäksi sisältää advanced connector -lisämoduulin, joka perustuu konfiguroitavaan ohjelmistokehykseen Groovy-skriptauskielellä, jolla voidaan kommunikoida etäjärjestelmien kanssa.

CMDBuild tarjoaa sen mitä ITIL kutsuu CMS:ksi (Configuration Management System). Kokonaisuudessa paketti koostuu seuraavista osista:

- CMDB (data ja palvelupuut)
- prosessit (Service Desk, workflows, muutoksenhallinta, resurssien elämänsykli, yms.)



- raportit
- työpöydän
- monipuolisten rajapintojen hallinta
- autentikaatio
- profilointi järjestelmät
- automaattiset hallinta mekanismit
- yhteistoimivuuden mekaniikat
- dokumentaatiot (CMDB press kit.)

CMDBuild tukee SOAP- ja REST-protokollia, joiden avulla voidaan muun muassa lisätä ja muokata tietokannan sisältöä ja lisäksi toteuttaa liitoksia esimerkiksi puhelinsovellusten kanssa.

CMDBuild-sovelluksen vaatimuksia ovat:

- asennus Linux järjestelmälle (suositeltu) tai Windows palvelimelle
- PostgreSQL tietokanta 9.0 tai uudempi (9.3 suositeltu)
- PostGIS 2.0 (vaihtoehtoinen)
- JDK 1.6 tai uudempi
- Apache Tomcat 6.0 tai uudempi (8.0 suositeltu)
- Alfresco 3.4 liitännäisten hallintaan (vaihtoehtoinen)
- Liferay Portal 6.0 (vaihtoehtoinen, itsepalvelu portaaliin)
- OCS Inventory (vaihtoehtoinen, laitekannan keruuseen). (CMDBuild Tecnical Manual 2016.)

Laitteiston puolen vaatimuksia ovat:

- nykysukupolven 4 ytiminen prosessori

- vähintään 4 GB keskusmuistia (6 GB tai 8 GB suositus)
- vähintään 100 GB tallennustilaa (CMDBuild Tecnical Manual 2016.)

Kuviossa 3 on esimerkki CMDBuild-järjestelmän käyttöliittymästä. Esimerkkikuvassa vasemmalla näkyy navigointikenttä, jossa kansioon lajiteltuna löytyvät konfiguraation osat, kuten palvelimet, tulostimet ja sijainnit. Jokainen yksittäinen osa sisältää sille olennaiset tiedot. Osille voi määrittää liitoksia muihin konfiguraation hallinnan järjestelmän tietokannassa oleviin konfiguraation osiin. Lisäksi konfiguraation osiin voi liittää dokumentteja.

The screenshot shows the CMDBuild web application interface. On the left is a navigation menu with categories like Dashboard, Basic archives, Purchases, Locations, Assets, and Asset (with sub-items like Computer, PC, Notebook, Server, Monitor, Printer, NetworkDevice, Rack, UPS, License, Report, Workflow). The main area displays a 'List - Asset' table with columns: Subclass, Code, Description, Serialnumber, Supplier, Brand, Model, Assignee, and Technical ri. Below the table are tabs for Card, Detail, Notes, Relations, History, and Attachments. The 'Detail' tab is active, showing information for asset 'MON0001': Description: Acer - AL1716, Serialnumber, Room, Assignee: Taylor William, and Workplace. At the bottom, there are tabs for General data, Administrative data, and Technical data, along with Save and Cancel buttons.

Subclass	Code	Description	Serialnumber	Supplier	Brand	Model	Assignee	Technical ri
Monitor	MON0001	Acer - AL1716			Acer	AL1716	Taylor ...	
Monitor	MON0002	Acer - B243WCydr	PRT576		Acer	B243WCydr	Miller U...	
PC	PC0001	Acer - Netbook ...	43434		Acer	D250	Williams...	Smith J...
Monitor	MON0003	Acer - V193HQb			Acer	V193HQb	Wilson ...	
Printer	PRT0001	Canon - IX5000	YT687		Canon	IX5000	Wilson ...	
Printer	PRT0002	Epson - ELP 6200L	RTD575		Epson	ELP 6200L	Williams...	

Kuvio 3. CMDBuild-käyttöliittymän demoversio. (CMDBuild demo: assets list N.d.)

## 5.2 ITop

ITop on Combodo nimisen yrityksen luoma avoimen lähdekoodin ITSM ratkaisu. Combodo perustettiin vuonna 2010 ja se tarjoaa ITSM-palveluita. Tällä hetkellä tarjoaa kolmea eri tuotetta: iTop, TeemIP ja ITSM designer. Tällä Combodo yrityksellä työskentelee yhdeksän täyspäiväistä henkilöä. Kuviossa 4 on esiteltyä virallinen Combodo logo. (Who we are N.d.)



Kuvio 4. Combodo yrityksen logo (Who we are N.d.)

ITop tulee sanoista IT Operational Portal, joka on avoimen lähdekoodin sovellus IT-ympäristön hallintaa. ITop-ominaisuuksia ovat muun muassa IT-ympäristön dokumentointi ja sen osien keskinäiset liitokset (palvelimet, sovellukset, verkkolaitteet, virtuaalikoneet, henkilöt, sijainnit jne.), ongelmien hallinta, käyttäjäpyyntöjen hallinta ja huoltoikkunoiden hallinta. (What is iTop 2016)

Muita sen ominaisuuksia ovat muun muassa tikettien hallinta, help desk, ongelman hallinta, muutoksen hallinta, palvelu katalogi ja automaattinen vaikutusanalyysin luominen konfiguraation kokoonpanon osien kanssa. Lisäksi tuote tukee dataintegroatioita ulkopuolisten lähteiden kanssa (sovellukset, tiedostot, tietokannat). Informaatiota voidaan lisätä, muokata ja poistaa määritettyjen asetusten mukaan. Dataa voidaan käsitellä ETL-tuotteella (Talend, Pentaho jne.) tai skriptillä. ITop-ominaisuuksista löytyy myös datan käsittely REST/JSON API -rajapinnan kautta. (What is iTop 2016)

ITSM designer -työkalulla, joka tulee maksullisen palvelun mukana, voidaan muokata iTop:n toimintaa. Graafisella käyttöliittymällä voidaan lisätä uusia konfiguraation kokoonpanon osia.

ITop on kirjoitettu PHP-ohjelmointikielellä. Palvelinalustan minimivaatimukset on esitetty taulukossa 1. ITop-järjestelmän minimaaliset konfiguraation vaatimukset ovat:

- **selain:** IE 9+, Firefox 3+, Safari 5+, Chrome
- **käyttöjärjestelmä:** Windows, Linux (Debian, Red Hat, FreeBSD), Solaris
- **web-palvelinsovellus:** Microsoft:n IIS, Apache tai nginx ja web-palvelinsovellus on konfiguroitu käyttämään PHP-tiedostoja, minimi PHP versio 5.3.6 tai uudempi parempi

- **Tietokantapalvelin:** MySQL versio 5.0 tai uudempi. (What is iTop 2016)

Taulukko 1. ITop-palvelin laitteiston vaatimukset (Installing iTop 2016)

Käyttöjärjestelmä	Resurssi	Minimi	Suositus
Linux	Levy	5 GB	20 GB
	Muisti	1 GB	2 GB
	Proessori	1 GHz (1 ydin)	2 GHz (2 ydin)
Windows	Levy	5 GB	20 GB
	Muisti	1 GB	2 GB
	Proessori	1 GHz (1 ydin)	2 GHz (2 ydin)

ITop-järjestelmässä osia joihin tallennetaan konfiguraation osia, kuten esimerkiksi palvelin, dokumentti tai projekti, kutsutaan luokiksi (class). ITop-järjestelmässä olemassa olevien luokkien ja uusien luominen tehdään luomalla XML-tiedosto, jossa määritellään luokkien muutokset tai uuden luokan määrittelyt. Kun XML-tiedostoon on määritetty, miten luokan tulisi rakentua, XML kootaan PHP-luokaksi sovelluksen asennuksessa. Aikaisemmin mainitulla maksullisella ITSM designer -työkalulla tämä voidaan tehdä graafisesti, mutta tämä onnistuu myös käsin muokkaamalla tekstieditorilla XML-tiedostoa.

### 5.3 Tuotteen valinta

Tuotteiden vertailun pohjalta päädyttiin valitsemaan konfiguraation hallinnan tuotteeksi iTop. Tässä tapauksessa tuli todettua, että konfiguraatioiden osien synkronointi konfiguraation hallintaan on yksinkertaisempi iTop-järjestelmän avulla. ITop-käyttöliittymä näyttää paremmalta ja selkeämmältä. Tämän lisäksi testauksen pohjalta iTop on yksinkertaisempi ja kevyempi käyttää kuin CMDBuild-järjestelmä. Lisäksi konfiguraatioiden osien liitosten toteuttaminen on helpompaa iTop-järjestelmällä.

## 6 Työn toteutus

### 6.1 Konfiguraation osien tuonti konfiguraation hallinnan tietokantaan

Konfiguraation osien tuonnin kanssa piti ottaa huomioon tuotteen itsenäinen toiminta. Riippuvuuksia toisiin osiin täytyy muodostaa muun muassa laitteiden, projektien, osaprojektien ja asiakkaiden kesken. Laitteiden tuonnin kanssa mahdollisia tapoja on useampia ja yksi potentiaalinen vaihtoehto oli verkon skannaaminen ja Protaconin käyttämän järjestelmän REST-rajapinnan hyödyntäminen.

Tässä työssä päädyttiin Protaconin käyttämän ValueFrame-järjestelmän REST-rajapinnan hyödyntämiseen, koska se on helppo toteuttaa ja sitä käyttämällä saadaan myös muodostettua riippuvuussuhteet esimerkiksi laitteiden ja projektien kesken. Tämän lisäksi kyseinen järjestelmä on tällä hetkellä se järjestelmä, jossa näiden konfiguraatioiden osien tietoja säilytetään ja pidetään yllä.

Protaconin käyttämässä ValueFrame-järjestelmässä on REST-ominaisuus, mutta tähän tarkoitukseen tarvittaville konfiguraation osille ei ollut ennestään ominaisuutta. Järjestelmään täytyi lisätä ominaisuudet, jotta saataisiin haettua REST-rajapintaa käyttäen laitteet, projektit, osaprojektit ja asiakkaat.

#### 6.1.1 REST-rajapinnan hyödyntäminen

Protaconin järjestelmää käyttäen määritettiin mitä tietoja halutaan tuoda konfiguraatioiden hallinnan tietokantaan. Laitteista halutaan tuoda seuraavat tiedot: laitteisiin liittyvät osaprojektit, projektien tiedot ja asiakkaat. Näistä konfiguraatioiden osien tiedoista selvitettiin mitä on mahdollista tuoda ja mitä halutaan tuoda. Tuotavia tietoja ovat esimerkiksi nimi, käyttötarkoitus, tekniset tiedot, vastuulliset henkilöt, riippuvaiset osaprojektit jne.

Tätä varten määritettiin jokaisesta laitteesta omat listat mitä tietoja halutaan. Protaconin ValueFrame-järjestelmään täytyi lisätä resurssit. Tähän tarvittiin ValueFrame-

henkilöstöä, joiden kanssa suunniteltiin mitä tarvitaan. Tämän pohjalta ValueFrame-tiimin henkilö määritteli ValueFrame järjestelmään tarvittaville resursseille taulupohjaiset REST-rajapinnat. Tämän jälkeen REST-rajapintaa käyttäen saadaan haettua ajantasaiset tiedot yksinkertaisessa JSON-muodossa.

## 6.2 Konfiguraation hallinnan järjestelmä

Järjestelmä, jolla konfiguraation hallinnan tietokannan järjestelmä tehtiin, oli iTop. ITop-järjestelmään tuodaan konfiguraation osien tiedot ja niitä pidetään yllä hyödyntäen Protaconin järjestelmän REST-rajapintaa.

Jotta tarvittavat tiedot saadaan synkronoitua Protaconin ValueFrame ja iTop järjestelmien välillä, tulee käyttää ETL-toteutusta (Extract-Transform-Load). Tähän tarkoitukseen ei ole valmista ratkaisua. Ensin haetaan data Protaconin REST-rajapinnasta, sitten data täytyy muokata sellaiseksi, että se voidaan syöttää iTop-järjestelmään.

Tässä tapauksessa päädyttiin tekemään oma python ohjelmointikielellä kirjoitettu skripti. Python-skripti ajetaan määritetyin ajoin ja saadaan tuotua tieto Protaconin ValueFrame-järjestelmästä iTop-järjestelmään ja pidettyä sen sisältämä tieto ajan tasalla.

Protacon ValueFrame järjestelmässä kullakin laitteella, osaprojektilla, projektilla, asiakkaalla on oma id-tietue. Tätä tietuetta tullaan käyttämään ITop-konfiguraation hallinnan järjestelmässä liitoksien muodostamisessa konfiguraation osien väleillä. Esimerkkinä tästä on, palvelimeen liittyy jokin osaprojekti, osaprojektiin projekti ja projektin asiakas.

ITop-järjestelmässä konfiguraation osia kutsutaan luokiksi (class), joihin mm. palvelinten tietoja tuodaan ja käytetään. Alustavasti palvelimien osalta ne eivät tukeneet ValueFrame:sta tuotavaa tietoa tai miten konfiguraation osien väleille tulisi muodostaa liitokset. Tämän lisäksi iTop-luokista ei löydy luokkia projekteille, osaprojekteille ja asiakkaille. Tätä varten täytyi muokata iTop-järjestelmää, muokata olemassa olevia luokkia ja luoda uusia. ITop-luokkien muokkaaminen ja luominen tehdään luomalla laajennus moduuli ja muokkaamalla moduulin XML-tiedostoja.

Kuviossa 5 esitetty iTop kansio ja tiedosto rakenne.

- conf, sisältää konfiguraatitiedostot.
- data, sisältää sovelluksen luoman tiedon.
- datamodel, sisältää meta data model määrittymiset, eli miten iTop-järjestelmän luokat tulee muodostua ja rakentua.
- env-xxx, nämä kansiot sisältävät muodostetun ympäristön rakenteen, jotka muodostetaan aina, kun sovellus asennetaan.
- extensions, tämä kansio on se sijainti, johon tullaan lisäämään kaikki moduulit, joilla muokataan iTop-järjestelmän toimivuutta, esimerkiksi tänne lisätään laajennukset, luokkien muokkaukset ja lisäykset.
- log, sisältää sovelluksen lokitiedostot, kuten setup.log ja error.log

### iTop 2.0 files layout

▶ iTop	
▶ addons	User-rights management add-ons
▶ application	Generic UI (PHP classes)
▼ conf	Configuration files
▶ production	Created by the setup
▶ toolkit	Created by the toolkit
▶ core	ORM engine
▶ css	Generic UI (Cascading Style Sheets)
▶ data	Local data generated by the application
▼ datamodels	
▶ 1.x	Data model for upgrading a 1.x version
▶ 2.x	Data model for 2.x versions
▶ dictionaries	Generic UI (Dictionaries)
▶ documentation	Online help for the setup
▶ env-production	"Compiled" data model, generated by the setup, do not edit
▶ env-toolkit	"Compiled" data model, generated by the toolkit, do not edit
▶ extensions	Copy your iTop extension modules in this folder
▶ images	Generic UI (Images)
▶ js	Generic UI (Javascript files)
▶ lib	Third party PHP libraries
▶ log	Log files
▶ pages	Generic UI pages
▶ portal	Portal user interface
▶ setup	Setup program
▶ synchro	Data synchronization
▶ toolkit	Toolkit for modifying the data model
▶ webservice	Import/Export and SOAP web services

Kuvio 5. ITop-järjestelmän kansio- ja tiedostorakenne (iTop Customization 2016)

Omien kustomoitujen laajennusmoduulien luominen onnistuu ajamalla aluksi iTop-järjestelmän oma ohjattu toiminto, jolla määritellään moduulin nimi, jolla se näkyy

asennuksessa, versionumero, kategoria ja tarvittavat riippuvuudet olemassa oleviin luokkiin, jos on tarve lainata niistä ominaisuuksia (katso kuvio 6).

Label	Value	Remarks
Module name	sample-add-class	Names starting with <code>itop-</code> and <code>combodo-</code> are reserved for use by Combodo. It is recommended not to put spaces or accentuated characters in the name of the module. Two modules with the same name cannot co-exist in the same iTop instance.
Module Label	Add Class Sample	This label will be displayed in the setup wizard. Localized characters and spaces are allowed
Module Version	1.0.0	The convention is to use a 3 digits numbering scheme: X.Y.Z
Category	business	Modules that provide modifications to the data model should be in the category 'business'
Dependencies	itop-config-mgmt/2.0.0,itop-endusers-devices/2.0.0	Our customization module depends on the modules: iTop Configuration Management (version 2.0.0) in which the Physical Device class is defined and iTop End User Devices (version 2.0.0) in which the "End User Devices" menu is defined

Kuvio 6. iTop-moduulin luominen ohjattulla toiminnolla (iTop Customization 2016)

Ohjattu toiminto luo uuden alustavan moduulin, kansion, joka on nimetty moduulin nimellä ja sisältää neljä alustavaa tiedostoa:

- datamodel.sample-add-class.xml
- module.sample-add-class.php
- en.dict.sample-add-class.php
- model.sample-add-class.php

Tässä esimerkissä tiedostoon `datamodel.sample-add-class.xml` tehdään määrytykset, miten luokka rakentuu, millä nimellä luodaan tietokantaan, mitä miten sitä käytetään, mitä kenttiä sisältää esim. nimi, selite, vastuuhenkilö ja miten muodostetaan riippuvuuksia toisiin konfiguraation osiin. Jotta saadaan uusi luokka näytettyä oikein käyttöliittymässä tiedostoon `en.dict.sample-add-class.php` tehdään määrytykset millä nimellä mikäkin kenttä näytetään käyttöliittymässä.

iTop-järjestelmän luokkia joita muokattiin, olivat palvelimet (Server), isäntälaitteet (Hypervisor) ja vieraslaitteet (Virtual Machine). Uudet luokat jotka luotiin olivat projekti (Project), osaprojekti (Subproject), asiakas (Customer). Jokaiselle muokattavalle ja luotavalle luokalle luotiin uusi moduuli ja muokattiin niiden tiedostoja halutulla tavalla.



### 6.2.1 iTop-konfiguraation hallinnan asennus

iTop-järjestelmä asennettiin Protaconin VMware-virtuaaliympäristöön. Koneelle määritettiin laiteresursseiksi: prosessori kahdella ytimellä, 2 GB RAM-muistia ja alustavasti 20 GB tallennustilaa, jota voidaan tarvittaessa lisätä helposti. ITop asennettiin linux Centos 7 -käyttöjärjestelmälle.

Palvelimelle, jolle asennettiin iTop, tehtiin alustavat Protaconin Linux-järjestelmän perusasennukset sisältäen muun muassa LDAP-palvelun konfiguroinnin, valvontaan liittämisen jne.

Palvelimelle asennettiin Apache web-palvelin, jolla saadaan tarjottua iTop:n palvelut. ITop on kirjoitettu PHP-koodilla ja tätä varten asennettiin PHP ja sen moduulit iTop ajamiseen. Tietokantapalvelimeksi asennettiin MariaDB, jossa iTop:n käyttämät tiedot säilytetään.

Koska palvelin asennettiin Centos 7 Linux-käyttöjärjestelmälle, käytössä on SELinux ja se haluttiin pitää käytössä. ITop-järjestelmän asennuksen kaikki vaiheet löytyvät liitteestä 6.

### 6.2.2 ITop-kustomisoidut moduulit

Kartoituksessa, jossa käytiin läpi mistä ja mitä tietoja halutaan tuoda ja sisällyttää konfiguraation hallinnan järjestelmässä, tuli esiin, että iTop-järjestelmää on tarve muokata. Tätä varten muokattiin iTop:ssa valmiina olevia luokkia (CI), kuten palvelimet, Hypervisor ja virtuaalipalvelimet. Niille oli tarve lisätä kentät tarvituille tiedoille ja muokata asetusta, jolla riippuvuudet rakentuvat muiden konfiguraation osien kesken. ITop-järjestelmään täytyi lisätä uutta kolme uutta luokkaa, koska vastaavia ei ennestään löytynyt. Nämä luokat olivat projektit, osaprojektit ja asiakkaat. Näille määritettiin kokonaan uudestaan mitä tietoja ne sisältävät ja miten riippuvuudet muodostuvat.

Palvelimien, virtualisoinnin isäntä- ja vieraslaitteiden välisiä liitoksia täytyi muokata. Esimerkkinä iTop-järjestelmässä ei ollut mahdollista määrittää palvelinta varmuuskopiointi palvelimeksi ja että tämä palvelin vaikuttaisi palvelimiin joita se varmentaa.

Lisäksi riippuvuuksia määritettiin siten, että palvelin tai virtuaalikone voi olla varmuuskopiointipalvelin ja siten voivat vaikuttaa palvelimiin tai virtuaalikoneisiin. Palvelimiin ja virtuaalikoneisiin voi liittyä osaprojekti ja se on silloin riippuvainen niistä.

Muita liitoksia konfiguraation osien välillä ovat muun muassa:

- osaprojekti vaikuttaa projektiin
- projekti vaikuttaa asiakkaaseen
- projektit ja osaprojektit vaikuttavat projektipäällikköihin
- asiakkaat vaikuttavat asiakasvastaaviin

Otetaan esimerkki lisäystä osaprojektin moduulin 2 tiedostoista datamodel.add\_subproject\_class ja en.dict.add\_subproject\_class, jotka konfiguroitiin ja niiden sisältö on esitetty liitteissä 7 ja 0. Seuraavana on poimittu datamodel-tiedostosta alkuosa, josta voidaan nähdä, että luokan nimi on Subproject ja se määritellään uutena. Tässä on tärkeää huomata \_delta="define" -määritys. Muokatessa olemassa olevaa luokkaa määritetäisiin \_delta="redefine". Toinen kohta on <parent>FunctionalCI</parent>, jolla voidaan määrittää riippuvuussuhteet niiden osien kesken kaikkien parent-tunnisteiden kesken, joiden sisään on määritetty FunctionalCI.

```
<?xml version="1.0" encoding="UTF-8"?>
<itop_design xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0">
  <classes>
    <class id="Subproject" _delta="define">
      <parent>FunctionalCI</parent>
    .
    .
    .
  </class>
</classes>
</itop_design>
```

Osiossa `<reconciliation>... </reconciliation>` määritellään millä tiedoilla kyseiselle luokalle(CI) voidaan muodostaa riippuvuuksia. Osiossa `<fields>...</fields>` määritellään kaikki kentät, joita luokka tulee sisältämään. Niille määritellään muun muassa tiedon tyyppi, nimi tietokannassa, onko tieto pakollinen jne. Osiossa `<details>...</details>` määritellään näytettävät kentät iTop:n käyttöliittymässä, jotta niitä voidaan käyttää. Osiossa `<search>...</search>` määritetään millä tiedoilla voidaan kyseisestä luokasta hakea tietoja. Osiossa `<list>...</list>`, määritetään näytettävät tietueet, kun listataan esimerkiksi palvelimet. Osiossa `<relations>...</relations>` määritetään vaikutus toisiin luokkiin, esimerkiksi palvelimessa on tietuekenttä, joka hakee osaprojektit ja siihen on valittu yksi osaprojekteista, niin palvelin vaikuttaa siihen. Tiedostossa `en.dict.add_subproject_class` (katso liite 0) määritellään, miten saadaan kaikki esitellyt kentät ihmisystävällisessä muodossa.

### 6.3 IPAM

Konfiguraation hallinnalle oli tarkoitus saada IP-osoitteiden hallinta. Joko konfiguraation hallinnasta itsessään oleva tai sen rinnalle rakennettava. Tässä päädyttiin rakentamaan konfiguraation hallinnan rinnalle IPAM ja sovellukseksi valittiin tuote nimeltään phpIPAM.

Palvelinten IP-osoitteiden CMDB-järjestelmään palvelimille vientiä varten, phpIPAM IP-osoitteilla täytyy olla tieto, joka ilmaisee millä palvelimella se on. PhpIPAM:ssa on mahdollista lisätä muun muassa IP-osoitteille omia kenttiä ja tässä tapauksessa tarvitaan normaali tekstikenttä. Alustavasti Protaconin järjestelmästä tuotiin kaikkien palvelinten IP-osoite tiedot siten, että phpIPAM:iin saatiin jokaiselle tarvitulle IP-osoitteelle myös palvelimen nimi, jolla se on käytössä.

#### 6.3.1 phpIPAM:n asennus

PhpIPAM asennettiin samalle palvelimelle kuin iTop, asennuksen vaiheet löytyvät liitteestä 8. PhpIPAM on nimensä mukaan kirjoitettu PHP-ohjelmointikielelle. PhpIPAM:lle ei tarvinnut erikseen asentaa web- tai tietokantapalvelinta, koska ne asen-

nettiin iTop:n asennuksen yhteydessä. PHP:n toimintaa varten tuli asentaa muutama moduuli, joita tarvitaan phpIPAM:n kanssa

## 6.4 Järjestelmien synkronointi

### 6.4.1 iTop ja ValueFrame

Tieto tullaan synkronoimaan ValueFrame-järjestelmästä iTop-järjestelmään itse tehdyllä Python-skriptillä (katso liitteet 1, 2 ja 0). Skripti ajastetaan ajettavaksi säännöllisin väliajoin cron-työkalulla ja näin saadaan tuotua halutut tiedot Protacoinin ValueFrame-järjestelmästä. Kun tietoa tuodaan ValueFrame:sta, tieto lisätään iTop:n työkalulla jokaiselle synkronoitavalle konfiguraation osalle niiden erillisiin synkronointitauluihin. Tämän jälkeen kyseinen työkalu vertaa, jokaisen konfiguraation osan kohdalla synkronointitaulua sen varsinaiseen tietokantatauluun. Tässä prosessissa käydään läpi, että lisääkö, muokataanko vai poistetaanko tietoa.

### 6.4.2 iTop ja phpIPAM

Python-ohjelmointikielellä phpIPAM- ja iTop-palveluiden väliset synkronoinnin skriptit (katso liitteet 1, 4 ja 5) ajetaan cron:lla säännöllisin ajoin ValueFrame- ja iTop-palveluiden synkronoinnin jälkeen. Jos IP-osoitteilla on nimitieto mille palvelimelle se kuuluu, voidaan PhpIPAM:sta viedä iTop:iin palvelinten IP-osoitteet. Tämän lisäksi PhpIPAM:ssa viedään kaikki VLAN:t iTop:iin. Jos phpIPAM:ssa IP-osoitteen aliverkko on liitetty johonkin VLAN:iin, tällä tiedolla iTop:ssa näille IP-osoitteille lisätään tieto, mihin VLAN:iin ne kuuluvat.

### 6.4.3 iTop ja vSphere

iTop:lla löytyi valmiina työkalu vSphere Data Collector, jolla saadaan kerättyä tietoja vSphere-palvelimesta. Tämä tarkoittaa sitä, että saadaan tuotua isäntälaitteet ja niillä olevat virtuaalikoneet. Isäntälaitteesta saadaan kerättyä muun muassa nimi, fyysisen

palvelimen valmistaja ja laitteen merkki, prosessoriydinten lukumäärä, muistin määrä, käyttöjärjestelmä ja tila. Virtuaalikoneilta voidaan kerätä selite, hallinta IP-osoite, prosessoriydinten luku, muistin määrä ja tieto missä isäntälaitteessa se on. Tällä työkalulla saadaan myös tuotua tieto, jos virtuaali host:t ovat klusterissa. Tässä tapauksessa rajattiin, että tämä työkalu vain päivittää iTop:ssa olevien palvelimien tietoja, eikä lisää uusia.

## 6.5 Konfiguraatioiden versionhallinta repositorio

Konfiguraatioiden versionhallinnan repositoriolla tarkoitetaan varastoa, johon tuodaan alustavasti palvelimista konfiguraatiot. Tämän jälkeen esimerkkinä kerran päivässä tehdään palvelimilla konfiguraatioille tarkistus, onko tapahtunut muutosta. Jos palvelimella on konfiguraatioissa tapahtunut muutos, tästä muutoksesta viedään tieto tähän varastoon. Tällä järjestelmällä on käyttöliittymä, josta voi selata, mitä konfiguraatioita palvelimilta löytyy ja mitä muutoksia on tapahtunut ja milloin.

Konfiguraation versionhallinnan repositorio tehtiin Gogs-nimisellä avoimen lähdekoodin tuotteella. Gogs on Git-repositorio, jolla voi hallita Git-repositorioita selaimella.

Tärkeä osa konfiguraatioiden repositoriassa on tietoturva. Jokainen palvelin lähettää tiedot omista konfiguraatioista tähän repositoriioon SSH-yhteydellä. Jokaisella palvelimella saa olla oikeus vain ajaa määritettyjä git komentoja repositorio palvelimelle ja jokaisella palvelimella saa olla pääsy vain omiin konfiguraatioihin. Tämä saadaan aikaiseksi käyttämällä SSH-avainpareja. Jokaisella palvelimella luodaan uniikit SSH-avainparit, Gogs-palvelimella luodaan repositorio palvelimen konfiguraatioille, luodaan käyttäjä, jolla käyttöoikeus tähän repositoriioon ja käyttäjälle lisätään luodun SSH-avainparin julkinen avain. Tällä saadaan aikaiseksi, että palvelimilla on oikeus käyttää vain Git-komentoja repositoriopalvelimelle ja käyttöoikeus vain omaan repositoriioon.

Kun koneille on varattu repositorio, tarvitaan toimenpiteet, joilla koneilla tehdyt muutokset viedään repositoriioon. Tähän käytetään etckeeper-nimistä avoimen läh-

dekoodin työkalua. Etckeeper käyttää muun muassa Git-työkalua ja sillä voi automatisoida muutosten viennin repositorioon.

### 6.5.1 Konfiguraatioiden versionhallinnan repositorion asennus

Gogs asennettiin samalle palvelimelle, jossa sijaitsee CMDB- ja phpIPAM-järjestelmät. Gogs-repositoriolle luotiin omat sivut ja Apache:lle luotiin virtualhost-konfiguraatio, jolla ohjataan sivusto Gogs-koodeihin.

Gogs-repositorioon tullaan tekemään palvelimille käyttäjä ja repositorio Ansible playbooks -työkalun kanssa. Tätä varten tehtiin playbooks-konfiguraatio. Tällä konfiguraatiolla määritetään koneilla ajettavaksi komentoja, joilla luodaan Gogs-repositoriota varten SSH-avainpari, Gogs API -rajapintaa käyttäen luodaan repositorioon koneen nimellä käyttäjä ja yksityinen repositorio. Käyttäjälle lisätään Gogs:iin luodun SSH-avainparin julkinen avain ja käyttäjälle määritetään oikeudet sille tarkoitettuun repositorioon. Luodaan SSH-konfiguraatio, kun kone yrittää ottaa SSH-yhteyttä repositorion verkko-osoitteeseen, se käyttää tähän tarkoitukseen luotua SSH-avainta. Saatiin muodostettua jokaiselle koneelle repositorioon oma repositorio, johon sillä on vain oikeus.

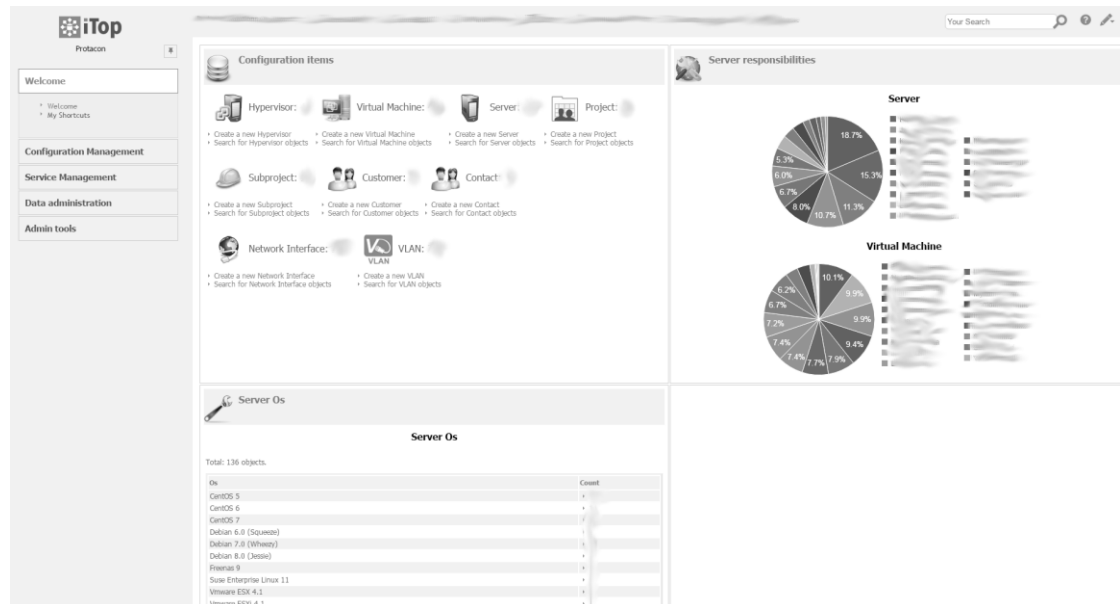
Ansible playbooks:lla tehtiin myös konfiguraatio, jolla koneilla suoritetaan komennot, joilla koneille asennetaan ja konfiguroidaan etckeeper. Etckeeper konfiguroidaan seuraamaan koko /etc-polun sisältöä. Alustavasti jokainen kone lähettää koko /etc-polun sisällön. Tämän jälkeen koneilla tehdään joka päivä tarkistus muutoksista ja tarvittaessa lähetetään tieto repositorioon.

## 7 Testaus

### 7.1 CMDB

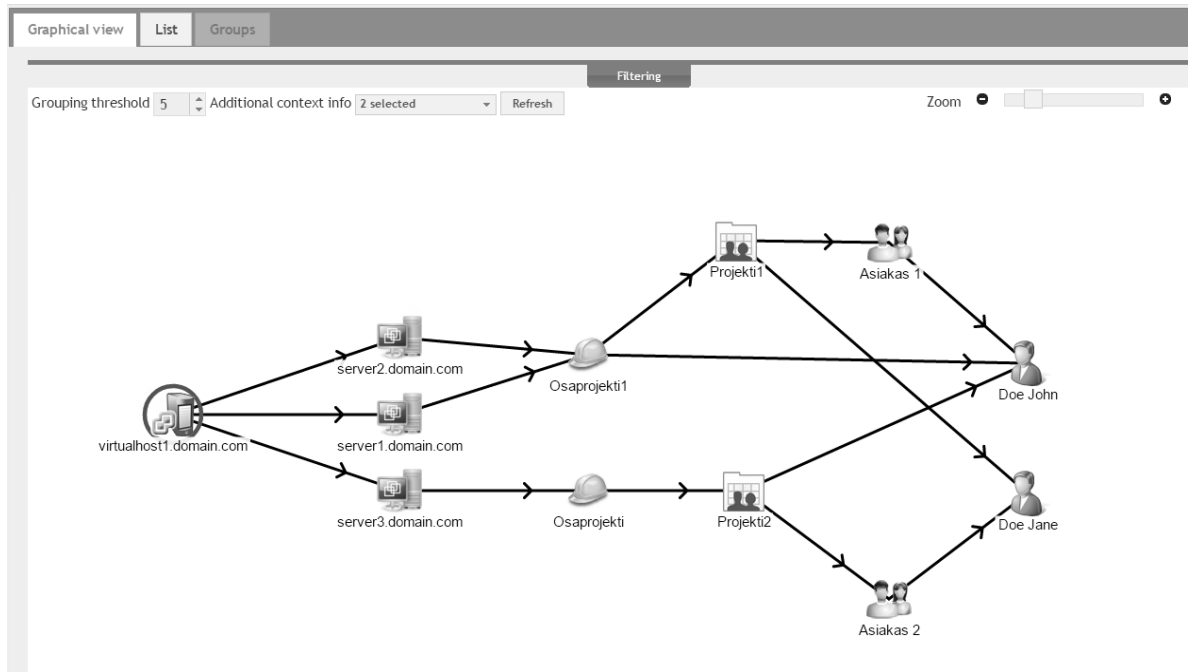
Tähän mennessä iTop-konfiguraation hallinnan järjestelmää on konfiguroitu ja muokattu olemassa olevia konfiguraation osia: palvelimet, hypervisor ja virtuaalikoneet.

Lisäksi on määritetty iTop-järjestelmään uudet luokat halutuille konfiguraation osille, eli projektit, osaprojektit ja asiakkaat. Kuviossa 7 on näkymä iTop-käyttöliittymästä, johon on kerätty olennaisimpia konfiguraation osia, joihin muun muassa tullaan synkronoimaan tietoa muista järjestelmistä (kuvioista on sotkettu lukumäärät ja nimet).



Kuvio 7. iTop-käyttöliittymän näkymä

Kuviossa 8 on esimerkki vaikutusanalyysin (eng. impact analysis) graafisesta kuvasta (kuvion konfiguraation osien nimet ovat keksittyjä). Kyseisessä kuvassa on analyysi hypervisor-komponentista ja mihin kaikkiin muihin konfiguraation osiin se vaikuttaa. Kuvassa vasemmalta alkaen on hypervisor, sillä olevat virtuaalikoneet, virtuaalikoneisiin liittyvät osaprojektit, osaprojekteihin liittyvät projektit sekä projekteihin ja asiakasliittymään liittyvät henkilöt. Analyysissä on mahdollista valita mitä konfiguraation osia halutaan nähdä. Esimerkkinä voitaisiin valita näytettäväksi ainoastaan virtuaalikoneet ja asiakkaat. Vaikutusanalyysistä on mahdollista tulostaa kuva PDF-muodossa. Impact analyysistä on mahdollista tulostaa kuva PDF-muotoon.



Kuvio 8. Hypervisor-komponentin vaikutusanalyysi graafisena esityksenä

Kuviossa 9 on esitys samasta vaikutusanalyysistä listamuodossa (katso kuvio 8). Listassa kaikki vaikutusanalyysin konfiguraation osat on jaoteltu. Listan saa tulostettua esimerkiksi PDF-muodossa. Tulostettaessa konfiguraation osia, on mahdollista, tulostaa kaikki niiden sisältämät tiedot ja myös valita vai haluamat.

Graphical view | List | Groups

Filtering

Grouping threshold 5 Additional context info 2 selected Refresh Zoom

**Elements impacted by virtualhost1.domain.com**

2 object(s) of class Customer found.

Total: 2 objects. [New...](#) [Other Actions](#)

Customer	Customer status	Customer manager
Asiakas 1	Aktiivinen asiakas	
Asiakas 2	Aktiivinen asiakas	

1 object(s) of class Hypervisor found.

Total: 1 objects. [Modify...](#) [New...](#) [Other Actions](#)

Hypervisor	Organization	Server	Farm	Business criticality	Move to production date
virtualhost1.domain.com	Protacon	virtualhost1.domain.com	undefined	low	

2 object(s) of class Person found.

Total: 2 objects. [New...](#) [Other Actions](#)

Person	First Name	Organization	Status	Location	Email	Phone
Doe Jane	Doe	Protacon	Active	undefined		
Doe John	Doe	Protacon	Active	undefined		

Kuvio 9. Hypervisor-komponentin vaikutusanalyysi listamuodossa



## 7.2 CMDB:n ja ValueFramen synkronointi

Synkronointi-skripti toi ensimmäisellä ajolla alustavasti tiedot ValueFrame-järjestelmästä ja loi konfiguraation osat iTop-järjestelmään. Tämän jälkeen joka kerta kun cron-työkalulla ajetaan synkronointiskripti, se ajaa synkronointiskriptin, se vertaa uutta tietoa iTop:ssa olevaan ja tuodun tiedon perusteella lisää uusia konfiguraation osia tai poistaa niitä ja muokkaa olemassa olevien konfiguraation osien sisältämiä tietoja.

## 7.3 CMDB:n ja VMwaren synkronointi

iTop:n valmiilla työkalulla saatiin tuotua tiedot isäntälaitteille ja virtuaalikoneille. Virtuaalikoneille saatiin myös tuotua ajantasainen tieto siitä, millä isäntälaitteella se sijaitsee. Lisäksi vSphere:stä saatiin tuotua klusteritiedot ja näiden tietojen avulla iTop:iin luotua klusterit ja liitettyä niihin kuuluvat isäntälaitteet.

## 7.4 CMDB:n ja phpIPAM:n synkronointi

Synkronointiskripti toi ensimmäisellä ajolla alustavasti IP-osoitteet palvelimille ja kaikki VLAN-tiedot phpIPAM:sta. Tämän jälkeen joka kerta kun cron-työkalulla ajetaan synkronointiskripti, se vertaa uutta tietoa iTop:ssa olevaan ja tuodun tiedon perusteella lisää uusia verkkorajapintoja iTop:ssa oleville palvelimille tai poistaa niitä ja tarvittaessa muokkaa niiden tietoja. Lisäksi liitettyä IP-osoitteisiin VLAN, jos sen aliverkko oli phpIPAM:ssa määritetty johonkin VLAN:iin.

## 7.5 PhpIPAM

PhpIPAM:iin tuotiin alustavasti kaikkien aliverkkojen tiedot, jotka löytyivät Protacconin järjestelmästä. Lisäksi tuotiin kaikki VLAN-tiedot olemassa olevasta dokumentista ja ne liitettiin oikeisiin aliverkkoihin. Protacconin järjestelmästä tuotiin myös alusta-

vasti kaikkien palvelimien IP-osoitetiedot. Jokaisessa IP-osoitteessa on rajapinnan nimi ja palvelimen nimi, jolle se on määritetty.

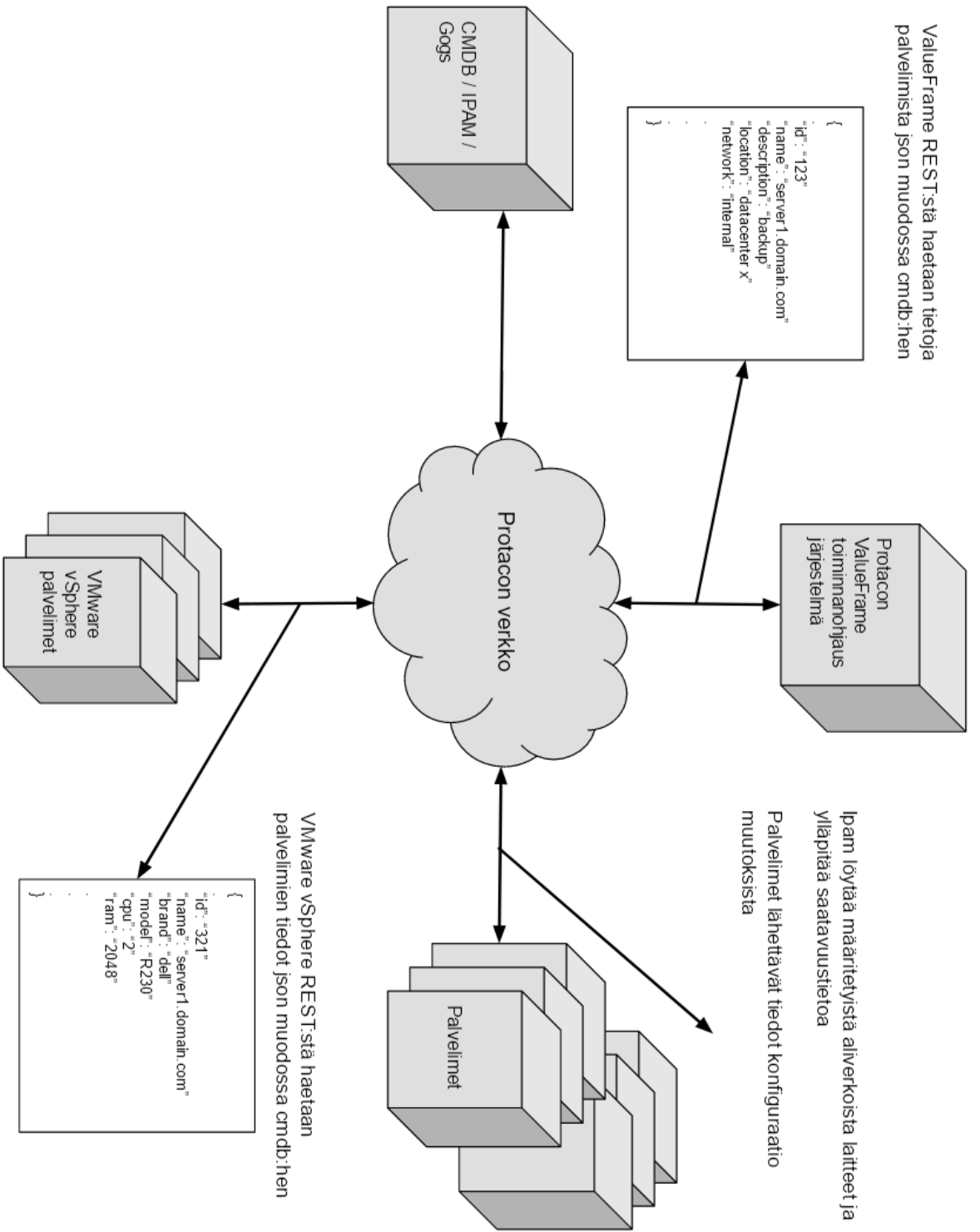
## 7.6 Konfiguraatioiden versionhallinnan -repositorio

Ansible playbooks -työkalulla saatiin luotua koneille Gogs-repositorioon omat repositoriot ja koneiden käyttöoikeudet rajoitettua niiden omiin. Koneet lähettivät SSH-yheydellä koko /etc-polun sisällön. Joka päivä tehdään tarkistus, onko tapahtunut muutos ja muutokset lähetetään koneelle määritettyyn repositorioon.

## 8 Kokonaisuuden kuva

Kuviossa 10 on esitetty kuva kokonaisuudesta. Kuviosta näkee, että CMDB, IPAM ja Gogs-repositorio on asennettu samalle palvelimelle. Tällä palvelimella haetaan Protaconin ValueFrame-järjestelmästä sen REST-rajapintaa käyttäen kaikkien palvelimien tiedot ja niihin liittyvät projektit, osaprojektit, asiakkaat ja henkilöstö. CMDB:ssä oleville palvelimille haetaan VMware-palvelimesta sen API-rajapintaa käyttäen palvelimilta virtuaalikoneiden tietoja, kuten prosessoriydinten määrä ja muistin määrä.

IPAM-järjestelmään tiettyihin aliverkkoihin on määritetty uusien koneiden löytäminen ja koneiden saatavuustiedon ylläpitäminen. IPAM varmistaa aliverkon IP-osoitteiden aktiivisuuden ICMP-protokollaa käyttäen. Palvelimet, joille on tehty konfiguraation versionhallinnan asennus, lähettävät tiedot konfiguraatioiden muutoksista Gogs-repositorioon niille tarkoitettuihin repositorioihin.



Kuvio 10. Kokonaisuuden kuva

## 9 Pohdinta

Konfiguraation hallinnan järjestelmistä ei varsinaisesti ollut aikaisemmin kokemusta. Opintojen aikana tuli käytyä CMDB tarkoitus lyhyesti ja lisäksi testattiin avoimen lähdekoodin CMDB-järjestelmiä tiketöinnin osalta.

Opinnäytetyössä selvitettiin mitä konfiguraation hallinnan järjestelmiä mitä löytyy, mukaan lukien ilmaiset avoimen lähdekoodin tuotteet sekä maksulliset tuotteet. Maksullisten osalta tuli todettua, että, kustannukset kasvavat liian korkeaksi, joten päädyttiin ilmaiseen avoimen lähdekoodin ratkaisuun.

Alustavasti tietoa piti tuoda Protaconin käyttämästä ValueFrame-järjestelmästä. Valitusta Valitusta konfiguraation hallinnan tuotteesta (iTop) löytyi valmiina työkalu, jolla tietoa saadaan lisättyä. Täytyi vain ensin kerätä tieto muista järjestelmistä ja tuoda se kyseiselle palvelimelle. Tässä päädyttiin ohjelmoimaan itse työkalu tiedon synkronointiin, jossa hyödynnettiin iTop:n työkalua.

Varsinaista kokemusta ohjelmoinnista ei ennestään ollut, mutta internetin dokumenttien avulla saatiin luotua työkalu, jolla saadaan REST-rajapintoja hyödyntäen synkronoitua tietoa muista järjestelmistä. Ohjelmoidut koodit muuttuivat matkan varrella useampaan otteeseen, kun oppi enemmän. Jatkokehityskohteina tarvetta olisi synkronointikoodin rakenteen optimoinnille ja uusien konfiguraation osien tuomiselle.

PhpIPAM-tuotteen IP-osoitehallinnan pystyttäminen oli yksinkertainen. Sille tuotiin Protaconin järjestelmästä kerralla kaikkien palvelinten IP-osoitteet. Aikaisemmin luodulla synkronoinnin työkalulla saadaan helposti vietyä IP-osoitteiden tiedot konfiguraation hallintaan. Protaconilla pysytään vielä olemassa olevassa järjestelmässä palvelimien IP-osoitteiden hallinnassajoten jatkokehityskohteena tällä osa-alueella olisi luoda synkronointi Protaconin ValueFrame-järjestelmän ja phpIPAM:n välille.

Konfiguraatioiden keräämisen kanssa tuli hieman ongelmia, koska tuli ajateltua asiaa liian vaikeasti. Tuli huomattua miten helppoa se on luoda Git-versionhallinnan työkaluilla.

Matkan varrella tuli paljon asioita ja tuli tehtyä virheitä. Näitä virheitä osaa nyt jatkossa välttää. Opinnäytetyötä tehdessä konfiguraation hallinnalle tuli tarve uusille ominaisuuksille, joita voisi toteuttaa. Lopputulosta katsoessa voidaan todeta, että täyttää työlle asetetut tavoitteet täyttyvät.

ITop CMDB -järjestelmä on tehty siten, että se ei on tehdy siten, että ei ole riippuvainen ValueFramen tiedosta. Tällä tarkoitetaan sitä, että jos ValueFrame-järjestelmästä halutaan luopua ja käyttää ainoastaan CMDB:tä, ei pitäisi tulla ongelmia.

Jatkokehityskohteenä IPAM- ja ValueFrame-järjestelmien välille voisi luoda synkronointisuhteen. Koska ValueFrame-järjestelmä pidetään käytössä ja laitteiden IP-osoitetietoja ylläpidetään ValueFramessa, tarvitaan synkronointisuhde, jotta saadaan pidettyä IPAM:ssa IP-osoitteissa tieto, että millä palvelimella se on, luotettavana.

Tällä hetkellä CMDB:een tuodaan vain Protaconin palvelimet ValueFramesta. Jatkokehityksenä tähän voisi tuoda myös muita laitteita, kuten kytkimiä ja asiakkaiden omia laitteita.

## Lähteet

- About Ansible. 2017. Ansible documentation verkkosivut. Viitattu 23.4.2017.  
<http://docs.ansible.com/ansible/index.html>
- Apache HTTP Server. 2014. Apache Software Foundation github verkkosivut. Viitattu 6.7.2017. <https://github.com/apache/httpd>
- CMDBuild demo: assets list. N.d. CMDBuild:n verkkosivut. Viitattu 26.2.2017.  
[http://www.cmdbuild.org/en/documentazione/screenshot/02\\_cmdbuild-demo-lista-asset/view](http://www.cmdbuild.org/en/documentazione/screenshot/02_cmdbuild-demo-lista-asset/view)
- CMDBuild press kit. N.d. CMDBuild:n verkkosivut. Viitattu 13.11.2016.  
<http://www.cmdbuild.org/en/diffusione/press-kit>
- CMDBuild ready2use. N.d. CMDBuild:n verkkosivut. Viitattu 13.11.2016.  
<http://www.cmdbuild.org/en/prodotti/ready2use>
- CMDBuild Technical Manual. 2016. CMDBuild:n verkkosivut. Viitattu 20.11.2016.  
<http://www.cmdbuild.org/file/manuali/technical-manual-in-english>
- Configuration Management Database. N.d. IBM Knowledge Center verkkosivut. Viitattu 30.10.2016.  
[http://www.ibm.com/support/knowledgecenter/SSWT9A\\_7.5.3/com.ibm.sccd.doc/overview/c\\_cmdb.html](http://www.ibm.com/support/knowledgecenter/SSWT9A_7.5.3/com.ibm.sccd.doc/overview/c_cmdb.html)
- Elkstein, M. N.d. What is REST?. Viitattu 20.11.2016. <http://rest.elkstein.org/>
- ETL (Extract-Transform-Load). N.d. Dataintegrationin verkkosivut. Viitattu 15.1.2017.  
<http://www.dataintegration.info/etl>
- Fielding, R.T. 2000. Architectural Styles and the Design of Network-based Software Architectures. Väitöskirja kalifornian yliopisto, DOCTOR OF PHILOSOPHY in Information and Computer Science. Viitattu 6.11.2016.  
[https://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)
- Gordon, A. 2015. Official (ISC)2 Guide to the CISSP CBK. 4. p. CRC Press. Books24x7 versio. Viitattu 6.11.2016.  
<http://library.books24x7.com.ezproxy.jamk.fi:2048/toc.aspx?bookid=112151>
- Installing iTop. 2016. iTop:n verkkosivut. Viitattu 5.3.2017.  
[https://wiki.openitop.org/doku.php?id=2\\_3\\_0:admin:installing\\_itop](https://wiki.openitop.org/doku.php?id=2_3_0:admin:installing_itop)
- iTop Customization. 2016. iTop:n verkkosivut. Viitattu 19.3.2017.  
[https://wiki.openitop.org/doku.php?id=2\\_3\\_0:customization:datamodel](https://wiki.openitop.org/doku.php?id=2_3_0:customization:datamodel)
- Learn Python Programming. N.d. Programiz:n verkkosivut. 19.2.2017.  
<https://www.programiz.com/python-programming#what-is-python>
- Leitão, P., Karnouskos, S. 2015. Industrial Agents: Emerging Applications of Software Agents in Industry. Elsevier Science and Technology Books, Inc. Books24x7 versio. Viitattu 27.11.2016.  
<http://library.books24x7.com.ezproxy.jamk.fi:2048/toc.aspx?bookid=82833>

- Me olemme Protacon. N.d. Protaconin verkkosivut. Viitattu 23.10.2016.  
<https://www.protacon.com/me-olemme-protacon/>
- Media. N.d. Protaconin verkkosivut. Viitattu 30.10.2016.  
<https://www.protacon.com/media/>
- Overview of Extraction, Transformation, and Loading. N.d. Database Data Warehousing Guide. Oraclen verkkosivut. Viitattu 15.1.2017.  
[http://docs.oracle.com/cd/B19306\\_01/server.102/b14223/etlover.htm#CEGJGJBD](http://docs.oracle.com/cd/B19306_01/server.102/b14223/etlover.htm#CEGJGJBD)
- phpIPAM. N.d. phpIPAM verkkosivut. 6.3.2017. <https://phpipam.net/>
- Playbooks. 2017. Ansible documentation verkkosivut. Viitattu 23.4.2017.  
<http://docs.ansible.com/ansible/playbooks.html>
- Protacon Group LinkedIn. N.d. LinkedIn -sivusto. Viitattu 23.10.2016.  
<https://www.linkedin.com/company/protacon-group>
- Quigley, J.M., Robertson, K.L. 2015. Configuration Management: Theory, Practice, and Application. Auerbach Publications. Books24x7 versio. Viitattu 6.11.2016.  
<http://library.books24x7.com.ezproxy.jamk.fi:2048/toc.aspx?bookid=74137>
- REST. N.d. ValueFramen verkkosivut. Viitattu 6.11.2016.  
<http://www.valueframe.fi/help/lisapalvelut/rest/>
- Shirley, L., Norfolk, D. 2014. Configuration Management: Expert Guidance for IT Service Managers and Practitioners. BCS. Books24x7 versio. Viitattu 6.11.2016.  
<http://library.books24x7.com.ezproxy.jamk.fi:2048/toc.aspx?bookid=62002>
- The CMDBuild Project. N.d. CMDBuild:n verkkosivut. viitattu 13.11.2016.  
<http://www.cmdbuild.org/en/progetto/descrizione>
- Toiminnanohjausjärjestelmä. N.d. ValueFramen verkkosivut. Viitattu 5.2.2017.  
<http://www.valueframe.fi/tuotteet/toiminnanohjausjarjestelma/>
- ValueFrame lisäpalvelut REST. N.d. ValueFramen verkkosivut. Viitattu 19.2.2017.  
<http://www.valueframe.fi/help/lisapalvelut/rest/>
- What is iTop. 2016. iTop:n verkkosivut. Viitattu 27.11.2016.  
<https://wiki.openitop.org/doku.php?id=start>
- What is REST API. N.d. Restfulapi verkkosivut. Viitattu 12.3.2017.  
<http://restfulapi.net/>
- Who we are. N.d. iTop:n verkkosivut. Viitattu 27.11.2016.  
<http://www.combodo.com/who-are-we>

## Liitteet

### Liite 1. ITop integraatioiden config.ini

```
[vfrest]
#Values:
# resource1: name of the REST-resource1
# resource2: name of the REST-resource2
# urlbase: direction where the rest-call is sent
# servername: name of the actual server that will "answer" the rest-call
# transferkey: customer based transfer key
resource1 = resource
resource2 = resource
urlbase = url
servername = servername
transferkey = key

[itop]
#Values:
# itop_username: itop user which is used in synchronization
# itop_password: itop user password
itop_username = username
itop_password = password

[phpipam]
#Values:
# username: phpipam login username
# password: phpipam login password
# server: direction where the rest-call is sent
# app_id: the app id to access
username = username
password = password
server = url
app_id = id
```



## Liite 2. ITop synkronisoinnin vallu2itoptool.py koodi

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

# vallu2itoptool.py
# Fetchrest luokka, noudetaan palvelimet valueframen kannasta palvelimet ja
backupliitokset.
# ITop synkronointi työkalun vuoksi REST:stä saadut tiedot kirjoitetaan csv muotoon.
#

from __future__ import print_function
from ConfigParser import SafeConfigParser
from tempfile import mkstemp
import os
import sys
import requests
import time
import hashlib
import json
import pandas
import csv

""" Get configuration values """
config = SafeConfigParser()
configFilePath = 'path'
config.read(configFilePath)

class fetchrest:

    def __init__(self):
        self.timestamp = int(time.time())

    """ Printing to stderr """
    def error(self,*args):
        print("ERROR: ", *args, file=sys.stderr)

    """ Creates temporary file and returns the path, path is stored in a global, add suf-
fix if needed
Attributes:
    suff: if neede add suffix end of the file
    """
    def make_tempfile(self,suff=''):
        fd, tmpfile = mkstemp(suffix=suff)
        os.close(fd)
        return tmpfile
```

```

""" Generates REST hash
Attributes:
    resource: resource name
"""
def genresthash(self, resource):
    # Update timestamp
    self.timestamp = int(time.time())
    # Create hash
    hash = hashlib.md5(str(self.timestamp) + "/" + resource + "/" + config.get('rest',
'transferkey')).hexdigest()
    return hash

""" Fetch json from REST and returns response
Attributes:
    resource: resource name
"""
def fetchrest(self,resource):
    header = {}
    header["X-VF-REST-USER"] = config.get('vfrest', 'servername')
    header["X-VF-REST-HASH"] = self.genresthash(resource)
    header["X-VF-REST-TIMESTAMP"] = str(self.timestamp)
    header["X-VF-REST-REAL-JSON-OUTPUT"] = "1"
    try:
        response = requests.get(config.get('vfrest', 'urlbase') + resource, head-
ers=header, verify=True)
    except (requests.exceptions.HTTPError, requests.exceptions.SSLError, re-
quests.exceptions.Timeout, requests.exceptions.ConnectionError) as e:
        self.error("Failed to fetch REST data: " + str(e) + ". Reason: " + response.text)
        sys.exit(1)
    try:
        return response.json()
    except ValueError as e:
        self.error("Failed to parse json: " + str(e))
        sys.exit(2)

""" Saves json data to file and returns jsonfile path
Attributes:
    resource: resource name
"""
def jsontofile(self,resource):
    jsonfile = self.make_tempfile('.json')
    input_data = self.fetchrest(resource)
    with open(jsonfile, "w") as f:
        json.dump(input_data, f)
    return jsonfile

""" Convert json to csv file and return csv file temporary file path """
def createitservercsv(self):

```

```

jsonfile = self.jsonToFile(config.get('vfrest', 'resource1'))
csvfile = self.make_tempfile('.csv')
df = pandas.read_json(jsonfile)
# Trasforms json data to csv file and determine columns
df.to_csv(csvfile, index=False, encoding="utf-8", columns=['column1','column2','column3'])
# Remove json file
try:
    os.remove(jsonfile)
except OSError as e:
    self.error("Failed to remove json file" + str(e))
    sys.exit(3)
return csvfile

""" Convert json to csv file and return csv file temporary file path """
def createBackupInkcsv(self):
    jsonfile = self.jsonToFile(config.get('vfrest', 'resource2'))
    csvfile = self.make_tempfile('.csv')
    df = pandas.read_json(jsonfile)
    # Trasforms json data to csv file and determine columns
    df.to_csv(csvfile, index=False, encoding="utf-8", columns=['column1','column2','column3'])
    # Remove json file
    try:
        os.remove(jsonfile)
    except OSError as e:
        self.error("Failed to remove json file" + str(e))
        sys.exit(4)
    return csvfile

""" Writes initial csv file to new file
Attributes:
    old: file to read
    new: file to write
"""
def readcsv(self,old,new):
    df = pandas.read_csv(old)
    df.to_csv(new, index=False, quoting=csv.QUOTE_ALL)

""" Default steps
Attributes:
    csvfile: file to modify
    dropna: column name, drop row if empty cell in column
    dropdup: column name, drop duplicate rows
    orgcol: organisation name, create org_id column, iTop needs org_id example
'Protacon'
    prikey: column values to use, create primary_key column, needed in iTop synch,
use example id column

```

```

copyfrom: values copy from, use with copyto, create new column with copied
values
copyto: values to copy, use with copyfrom, create new column with copied val-
ues,
"""
def de-
faults-
teps(self, csvfile, dropna=None, dropdup=None, orgcol=None, prikey=None, dupcolumn
=None, copyfrom=None, copyto=None):
    df = pandas.read_csv(csvfile)
    try:
        if dropna:
            df.dropna(subset=[dropna], inplace=True)
        if dropdup:
            df = df.drop_duplicates(subset=[dropdup])
        if orgcol:
            df.insert(0, 'org_id', orgcol)
        if prikey:
            df['primary_key'] = df[prikey]
        if copyfrom and copyto:
            df[copyto] = df[copyfrom]
    except KeyError as e:
        fetchrest.error('Defaultsteps failed, check that value ' + str(e) + ' is correct in '
+ sys._getframe(1).f_code.co_name)
        sys.exit(5)
    df.to_csv(csvfile, index=False, quoting=csv.QUOTE_ALL)

""" Creates url column
csvfile: file to modify
colname: name for column
urltmpl: url template to use
urlid: id value to use with urltmpl
Example addurl(tmpfile, 'column', 'url', 'column')
"""
def addurl(self, csvfile, colname, urltmpl, urlid):
    df = pandas.read_csv(csvfile)
    try:
        df[colname] = urltmpl + df[urlid].astype(str)
    except KeyError as e:
        fetchrest.error('Addurl failed, check that value ' + str(e) + ' is correct in ' +
sys._getframe(1).f_code.co_name)
        sys.exit(5)
    df.to_csv(csvfile, index=False, quoting=csv.QUOTE_ALL)

""" Renames columns
csvfile: file to modify
kwargs: renames column name, using keyword arguments
Example renamecolumn(csvfile, old1='new1', old2='new2')
"""

```

```

def renamecolumn(self, csvfile, **kwargs):
    df = pandas.read_csv(csvfile)
    for col in kwargs:
        old = '%s' % (col)
        new = '%s' % (kwargs[col])
        df = df.rename(columns={old: new})
    df.to_csv(csvfile, index=False, quoting=csv.QUOTE_ALL)

""" Filter rows
    csvfile: file to modify
    kwargs: filters rows containing given value using keyword arguments
Example rowfilter(csvfile, column='1')
"""

def rowfilter(self, csvfile, **kwargs):
    df = pandas.read_csv(csvfile)
    for col in kwargs:
        col = '%s' % (col)
        value = '%s' % (kwargs[col])
        filter = df[col] == int(value)
        df[filter].to_csv(csvfile, index=False, quoting=csv.QUOTE_ALL)

""" Merge two columns
Attributes:
    csvfile: file to modify
    newcol: new column name
    fcol: firts column to merge
    scol second column to merge
    separator: separator if neened, as default nothing ""
"""

def mergecolumns(self, csvfile, newcol, fcol, scol, separator=""):
    df = pandas.read_csv(csvfile)
    df[newcol] = df.apply(lambda row: '{0} {1}
{2}'.format(row[fcol], separator, row[scol]), axis=1)
    df.to_csv(csvfile, index=False, quoting=csv.QUOTE_ALL)

""" Dertermine columns to keep
Attributes:
    csvfile: file to modify
    col: columns to keep, add as list, example keepcolumns([col1,col2,col3])
"""

def keepcolumns(self, csvfile, col):
    df = pandas.read_csv(csvfile)
    try:
        df = df[col]
    except KeyError as e:
        fetchrest.error('Assembling the columns failed in ' +
sys._getframe(1).f_code.co_name + ': ' + str(e))
        sys.exit(5)

```

```
df.to_csv(csvfile, index=False, quoting=csv.QUOTE_ALL)
```

```
""" Synchronizes csvfile to iTop with iTop php script
```

```
Attributes:
```

```
    csvfile: file to modify
```

```
    sourceid: synchrho data id, can be found in iTop Admin tools -> Synchronization
```

```
Data Sources.
```

```
"""
```

```
def synctoitop(self, csvfile, sourceid):
```

```
    os.system('php -q path/itop/synchro/synchro_import.php --auth_user=' + config.get('itop', 'itop_username') + ' --auth_pwd=' + config.get('itop', 'itop_password') + ' --data_source_id=' + str(sourceid) + ' --separator=', --csvfile=' + csvfile)
```

```
""" Convert all numbers to int
```

```
Some reason modifactions covert numbers to float
```

```
Attributes:
```

```
    csvfile: file to modify
```

```
    column: column values to int
```

```
"""
```

```
def convertnumtoint(self, csvfile, *column):
```

```
    df = pandas.read_csv(csvfile)
```

```
    for i in column:
```

```
        df[i] = df[i].astype(int)
```

```
    df.to_csv(csvfile, index=False, quoting=csv.QUOTE_ALL)
```

```
""" Synchronize backup links to iTop
```

```
Attributes:
```

```
    csvfile: file to modify
```

```
    servers: servers csv file path
```

```
    bulinks: backup link csv file path
```

```
    buserver_id: iTop class id which backs up, example server_id, virtualmachine_id, hypervisor_id
```

```
"""
```

```
def preparebulnk(self, tmpfile, servers, bulinks, buserver_id):
```

```
    df1 = pandas.read_csv(bulinks)
```

```
    df2 = pandas.read_csv(servers)
```

```
    # Match servername for column and column to column
```

```
    match1 = pandas.merge(left=df1, right=df2, how = 'left', left_on = 'column', right_on = 'column')
```

```
    match1 = match1.rename(columns={'column': buserver_id})
```

```
    # Match servername for serverid and column to column
```

```
    match2 = pandas.merge(left=df1, right=df3, how = 'left', left_on = 'column', right_on = 'column')
```

```
    match2 = match2.rename(columns={'column': 'column'})
```

```
    # Merge match1 and match2
```

```
    matched = pandas.merge(left=match2, right=match2, left_on = 'id', right_on = 'id')
```

```
    # Add primary_key column, needed in iTop synchronization
```

```
    matched['column'] = matched['id']
```

```

# Determine required columns
keep_col = ['column',busserver_id,'primary_key']
matched = matched[keep_col]
# Drop rows if empty cell in column
matched = matched.dropna(subset=['column',busserver_id])
# Write to csv file
matched.to_csv(tmpfile, index=False, quoting=csv.QUOTE_ALL)

""" Split names to last and first
Attributes:
    csvfile: file to modify
    oldcol: original column
    lastname: lastname column name
    firstname: firstname column name
"""

def splitname(self,csvfile, oldcol, lastname, firstname):
    df = pandas.read_csv(csvfile)
    df[lastname], df[firstname] = zip(*df[oldcol].str.split().tolist())
    df.to_csv(csvfile, index=False, quoting=csv.QUOTE_ALL)

""" Swap lastname and firstname position
Attributes:
    col: a column value which is swapped
    namecol: new column name
"""

def swapname(self,csvfile, col, namecol):
    df = pandas.read_csv(csvfile)
    df['lastname'], df['firstname'] = zip(*df[col].str.split().tolist())
    df[namecol] = df['firstname'] + ' ' + df['lastname']
    df.to_csv(csvfile, index=False, quoting=csv.QUOTE_ALL)

""" Removes file
Attributes:
    files: file to remove
"""

def removefile(self,*files):
    for file in files:
        os.remove(file)

```

## Liite 3. ITop synkronisoinnin vallu2itopsync.py koodi

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

# vallu2itopsync.py
# Vallu2itoptool työkalulla haetaan REST rajapinnasta data, joka on kirjoitettu csv
muotoon.
# Tuodu data muunnetaan iTop synkronointi formaattiin, että ne vastaavat iTop
synkronointi
# tauluja. Synkronointiin käytetään Itop:n omaa työkalua, jossa csv tiedostoa käyte-
tään.
#

from ConfigParser import SafeConfigParser
from vallu2itoptool import fetchrest
from tempfile import mkstemp
import os
import sys
import csv
import pandas
import time

""" Get configuration values """
config = SafeConfigParser()
configFilePath = 'path'
config.read(configFilePath)

""" Instantiate the fetchrest class """
fetchrest = fetchrest()

""" Get initial data from REST in csv format """
csv = fetchrest.createcsv()
csv = fetchrest.createbackuplnkcsv()

""" Synchronize persons to iTop """
def synchro_person():
    tmpfile = fetchrest.make_tempfile(suff='.csv')
    df = pandas.read_csv(csv)
    try:
        accountmanager = df[['columns']].dropna()
        accountmanager = accountmanager.rename(columns={'columns'})
        projectmanager = df[['columns']].dropna()
        projectmanager = projectmanager.rename(columns={'columns'})
        subprojectmanager = df[['columns']].dropna()
        subprojectmanager = subprojectmanager.rename(columns={'columns'})
```



```

        person = pandas.concat(['columns'], ignore_index=True)
        person.to_csv(tmpfile, index=False, quoting=csv.QUOTE_ALL)
    fetch-
rest.defaultsteps(tmpfile,dropdup='columns',orgcol='columns',prikey='columns')
    fetchrest.splitname(tmpfile, 'columns', 'columns', 'columns')
    keep_col = ['columns']
    fetchrest.keepcolumns(tmpfile,keep_col)
    fetchrest.convertnumtoint(tmpfile, 'columns', 'columns')
    fetchrest.synctoitop(tmpfile,'1')
finally:
    fetchrest.removefile(tmpfile)

""" Synchronize customers to iTop """
def synchro_customer():
    tmpfile = fetchrest.make_tempfile(suff='.csv')
    try:
        fetchrest.readcsv(csv,tmpfile)
        fetchrest.defaultsteps(tmpfile,'columns')
        fetchrest.addurl(tmpfile,'columns','url','columns')
        fetchrest.renamecolumn(tmpfile,accountName='columns')
        keep_col = ['columns']
        fetchrest.keepcolumns(tmpfile,keep_col)
        fetchrest.synctoitop(tmpfile,'2')
    finally:
        fetchrest.removefile(tmpfile)

""" Synchronize projects to iTop """
def synchro_project():
    tmpfile = fetchrest.make_tempfile(suff='.csv')
    try:
        fetchrest.readcsv(csv,tmpfile)
    fetch-
rest.defaultsteps(tmpfile,dropna='columns',dropdup='columns',orgcol='columns',prikey='columns')
        fetchrest.addurl(tmpfile,'columns','url','columns')
        fetchrest.renamecolumn(tmpfile,'columns'='columns', 'columns'='columns')
        fetchrest.mergecolumns(tmpfile,'columns','columns','columns',separator='-')
        keep_col = ['columns']
        fetchrest.keepcolumns(tmpfile,keep_col)
        fetchrest.synctoitop(tmpfile,'4')
    finally:
        fetchrest.removefile(tmpfile)

""" Synchronize subprojects to iTop """
def synchro_subproject():
    tmpfile = fetchrest.make_tempfile(suff='.csv')
    try:
        fetchrest.readcsv(csv,tmpfile)

```

```

    fetch-
rest.defaultsteps(tmpfile,dropna='columns',dropdup='columns',orgcol='columns',prikey='columns')
    fetchrest.addurl(tmpfile,'columns','url','columns')
    fetchrest.renamecolumn(tmpfile,accountName='columns', subproject-
Name='columns', projectName='columns')
    fetchrest.mergecolumns(tmpfile,'columns','columns','columns',separator='-')
    fetchrest.mergecolumns(tmpfile,'columns','columns','columns',separator='-')
    keep_col = ['columns']
    fetchrest.keepcolumns(tmpfile,keep_col)
    fetchrest.synctoitop(tmpfile,'2')
finally:
    fetchrest.removefile(tmpfile)

""" Synchronize server locations to iTop """
def synchro_location():
    tmpfile = fetchrest.make_tempfile(suff='.csv')
    try:
        fetchrest.readcsv(csv,tmpfile)
        fetch-
rest.defaultsteps(tmpfile,dropna='column',dropdup='column',orgcol='column',prikey='column')
        fetchrest.renamecolumn(tmpfile,locationName='name')
        keep_col = ['columns']
        fetchrest.keepcolumns(tmpfile,column)
        fetchrest.synctoitop(tmpfile,'48')
    finally:
        fetchrest.removefile(tmpfile)

""" Synchronize servers to iTop """
def synchro_server():
    tmpfile = fetchrest.make_tempfile(suff='.csv')
    try:
        fetchrest.readcsv(csv,tmpfile)
        fetchrest.defaultsteps(tmpfile,orgcol='column',prikey='column')
        fetchrest.rowfilter(tmpfile,column='0')
        fetchrest.addurl(tmpfile,'column','url','column')
        fetchrest.renamecolumn(tmpfile,column='column',column='column', column='column')
        keep_col = ['columns']
        fetchrest.keepcolumns(tmpfile,keep_col)
        fetchrest.synctoitop(tmpfile,'6')
    finally:
        fetchrest.removefile(tmpfile)

""" Synchronize hypervisors to iTop """
def synchro_hypervisor():
    tmpfile = fetchrest.make_tempfile(suff='.csv')
    try:

```

```

fetchrest.readcsv(csv,tmpfile)
fetchrest.defaultsteps(tmpfile,orgcol='columns',prikey= column)
fetchrest.rowfilter(tmpfile, column ='1')
fetchrest.addurl(tmpfile, column,'url', column)
fetchrest.renamecolumn(tmpfile,column='name')
df = pandas.read_csv(tmpfile)
df['server_id'] = df['name']
df.to_csv(tmpfile, index=False, quoting=csv.QUOTE_ALL)
keep_col = ['name', column_id', column_key','org_id']
fetchrest.keepcolumns(tmpfile,keep_col)
fetchrest.synctoitop(tmpfile,'7')
finally:
    fetchrest.removefile(tmpfile)

""" Synchronize virtual machines to iTop """
def synchro_virtualmachine():
    tmpfile = fetchrest.make_tempfile(suff='.csv')
    try:
        fetchrest.readcsv(csv,tmpfile)
        fetchrest.rowfilter(tmpfile,'columns'='1')
        fetchrest.defaultsteps(tmpfile,orgcol='columns',prikey='columns')
        fetchrest.addurl(tmpfile,'columns','url','columns')
        fetchrest.renamecolumn(tmpfile,'columns'='columns','columns'='columns', 'columns'='columns')
        keep_col = ['columns']
        fetchrest.keepcolumns(tmpfile,keep_col)
        fetchrest.synctoitop(tmpfile,'8')
    finally:
        fetchrest.removefile(tmpfile)

""" Synchronize backup links when server is backup server """
def synchro_server_bulnk():
    tmpfile = fetchrest.make_tempfile(suff='.csv')
    try:
        fetchrest.preparebulnk(tmpfile,csv,csv,"columns")
        fetchrest.synctoitop(tmpfile,'9')
    finally:
        fetchrest.removefile(tmpfile)

""" Synchronize backup links when virtual machine is backup server """
def synchro_vm_bulnk():
    tmpfile = fetchrest.make_tempfile(suff='.csv')
    try:
        fetchrest.preparebulnk(tmpfile,csv,csv,'columns')
        fetchrest.synctoitop(tmpfile,'10')
    finally:
        fetchrest.removefile(tmpfile)

""" Synchronize contacts links to customers, projects and subprojects """

```

```

def synchro_contact_functionalci_Ink():
    tmpfile = fetchrest.make_tempfile(suff='.csv')
    df = pandas.read_csv(csv)
    try:
        customer = df[['columns']].dropna()
        customer = customer.rename(columns={'columns'})
        project = df[['columns']].dropna()
        project = project.rename(columns={'columns'})
        subproject = df[['columns']].dropna()
        subproject = subproject.rename(columns={'columns'})
        links = pandas.concat(['columns','columns','columns'], ignore_index=True)
        links.to_csv(tmpfile, index=False, quoting=csv.QUOTE_ALL)
        fetchrest.swapname(tmpfile, 'columns', 'columns')
        keep_col = ['columns','columns','columns']
        fetchrest.keepcolumns(tmpfile,keep_col)
        fetchrest.defaultsteps(tmpfile,dropdup='columns')
        fetchrest.synctoitop(tmpfile,'11')
    finally:
        fetchrest.removefile(tmpfile)

if __name__ == '__main__':
    """ Run synchronization functions, right order is needed so that the dependencies
    between resources
    are correctly formed in iTop. e.g person -> customer -> project -> subproject ->
    server -> hypervisor
    -> virtual machine -> links
    """
    try:
        synchro_person()
        synchro_customer()
        synchro_project()
        synchro_subproject()
        synchro_location()
        synchro_server()
        synchro_hypervisor()
        synchro_virtualmachine()
        synchro_server_bulnk()
        synchro_vm_bulnk()
        synchro_contact_functionalci_Ink()
        print ('Itop synchronization completed successfully')
    finally:
        # Remove csv and csv files
        fetchrest.removefile(csv,csv)

```

## Liite 4. ITop synkronisoinnin phpipam2itoptool.py koodi

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

# phpipam2itoptool.py
# phpIPAM luokka. Käyttää phpIPAM:n REST rajapintaa tiedon hakuun phpIPAM:sta.

from __future__ import print_function
from tempfile import mkstemp
from requests.auth import HTTPBasicAuth
from ConfigParser import SafeConfigParser
import requests
import json
import sys
import os
import pandas

class phpIPAM:
    """
    Attributes:
        server: the base server location
        app_id: the app ID to access
        username: login username
        password: login password
    """
    def __init__(self, server, app_id, username, password):
        self.server = server
        self.app_id = app_id
        self.username = username
        self.password = password
        self.base = '%s/api/%s/' %(self.server,self.app_id)

    """ Error printing"""
    def error(self, *args):
        print("ERROR: ", *args, file=sys.stderr)

    """ Removes files
    Attributes:
        files = path/to/the/file
    """
    def removefile(self, *files):
        for file in files:
            os.remove(file)

    """ Creates temporary file, returns path
```

Attributes:

```
suff: suffix # Add file suffix if needed
"""
```

```
def make_tempfile(self,suff=""):
    fd, tmpfile = mkstemp(suffix=suff)
    os.close(fd)
    return tmpfile
```

""" Authentication """

""" Login to phpIPAM and get a token """

```
def login(self):
    response = requests.post(self.base + 'user/', auth=HTTPBasicAuth(self.username,
self.password))
    if response.status_code != 200:
        self.error("phpipam.login failure: %s " %(response.text))
        sys.exit(1)
    # now we have a token
    self.token = response.json()['data']['token']
    self.headers = {'token': self.token}
    return self.headers
```

""" Get the id of a specific section by using section name

Attributes:

```
section = The name of the spesific section
"""
```

```
def get_section_id(self, section):
    response = requests.get(self.base + 'sections/%s/?links=false' %(section),
headers=self.headers)
    sections_get_id = json.loads(response.text)
    if response.status_code != 200:
        self.error("phpipam.get_sections_id failure: %s " %(response.text))
        sys.exit(1)
    if not sections_get_id['success']:
        self.error("phpipam.get_sections_id failure: %s " %(response.text))
        sys.exit(1)
    return sections_get_id['data']['data']
```

""" Get the subnets for a specific section

Attributes:

```
section_id: section id
"""
```

```
def get_subnets(self, section_id):
    response = requests.get(self.base + 'sections/%s/subnets/?links=false' %(section_id), headers=self.headers)
    section_get_subnets = json.loads(response.text)
    if response.status_code != 200:
        self.error("phpipam.get_subnets failure: %s " %(response.text))
        sys.exit(1)
```

```

    return section_get_subnets['data']

    """ Get the addresses for a specific subnet
    Attributes:
        subnet: subnet id
    """
    def get_addresses(self, subnet):
        response = requests.get(self.base + '/subnets/%s/addresses/?link=false' %(sub-
net), headers=self.headers)
        return response.json()

    """ Parse subnet ids in a section
    Attributes:
        section_id: section id
    """
    def parse_subnetids(self, section_id):
        subnets = self.get_subnets(section_id)
        ids = []
        for i in subnets:
            id = i['id']
            ids.append(id)
        return ids

    """ Get specific vlan
    Attributes:
        vlanid: phpipam vlan id
    """
    def get_vlan(self, vlanid):
        response = requests.get(self.base + '/vlan/%s/?link=false' %(vlanid),
headers=self.headers)
        if response.status_code != 200:
            self.error("phpipam.get_vlan failure: %s " %(response.text))
            sys.exit(1)
        return response.json()

    """ Get netmask from specific subnet
    Attributes:
        subnetid: phpipam subnet id
    """
    def get_subnet_netmask(self, subnetid):
        response = requests.get(self.base + '/subnets/%s/?link=false' %(column),
headers=self.headers)
        vlan_get_id = json.loads(response.text)
        if response.status_code != 200:
            self.error("phpipam.get_subnet_netmask failure: %s " %(response.text))
            sys.exit(1)
        return vlan_get_id['data']['column']['column']

    """ Get all vlans """

```

```

def get_all_vlans(self):
    response = requests.get(self.base + '/vlan/?link=false', headers=self.headers)
    return response.json()

""" Get all vlans in csv format """
def get_all_vlans_csv(self):
    jsonfile = self.make_tempfile('.json')
    csvfile = self.make_tempfile('.csv')
    try:
        input_data = self.get_all_vlans()
        input_data = input_data['data']
        with open(jsonfile, 'w') as f:
            json.dump(input_data, f)
        df = pandas.read_json(jsonfile)
        df.to_csv(csvfile, index=False, encoding="utf-8", columns=['column', 'column', 'column', 'column', 'column', 'column', 'column', 'column'])
    finally:
        self.removefile(jsonfile)
    return csvfile

""" Get specific subnet
Attributes:
    subnetid: phpipam subnet id
"""
def get_subnet(self, subnetid):
    response = requests.get(self.base + '/subnets/%s/?links=false' %(subnetid),
headers=self.headers)
    return response.json()

```



## Liite 5. ITop synkronisoinnin phpipam2itopsync.py koodi

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

# phpipam2itopsync.py
# phpipam2itoptool työkalun avulla heataan laitteiden IP-osoitteet ja kaikki vlan:t.
# Tuotu data muunnetaan iTop synkronoinnin formaattiin. IP-osoitteet synkronoi-
# daan
# iTop:iin laitteille ja vlan liitokset IP-osoitteille.

from phpipam2itoptool import phpIPAM
from vallu2itoptool import fetchrest
from ConfigParser import SafeConfigParser
import json
import sys
import os
import pandas
import csv

""" Get configuration values """
config = SafeConfigParser()
configFilePath = 'path'
config.read(configFilePath)

""" Instantiate the fetchrest class """
fetchrest = fetchrest()

"""
Attributes:
    server: direction where the rest-call is sent
    app_id: the app ID to access
    username: login username
    password: login password
"""
server = config.get('phpipam', 'server')
app_id = config.get('phpipam', 'app_id')
username = config.get('phpipam', 'username')
password = config.get('phpipam', 'password')

# Create phpipam instance
ipam_session = phpIPAM(server, app_id, username, password)

# Initiate the phpIPAM session
ipam_session.login()
```

""" Get all the addresses from IPv4 section with a 'VF server name' value in csv format """

```
def get_ipv4section_addresses():
    jsonfile = ipam_session.make_tempfile(suff='.json')
    csvfile = ipam_session.make_tempfile(suff='.csv')
    try:
        section_id = ipam_session.get_section_id("id")
        address_ids = []
        address_ips = []
        address_netmasks = []
        address_ifnames = []
        address_hostnames = []
        address_devicenames = []
        address_subnetids = []
        addresses = []
        subnet_ids = ipam_session.parse_subnetids(section_id)
        for i in subnet_ids:
            data = ipam_session.get_addresses(i)
            if data['code'] != 200:
                continue
            netmask = ipam_session.get_subnet_netmask(i)
            for i in data['data']:
                address_ids.append(i['id'])
                address_ips.append(i['id'])
                address_ifnames.append(i['id'])
                address_netmasks.append(id)
                address_hostnames.append(i['id'])
                address_devicenames.append(i['id name'])
                address_subnetids.append(i['id'])
            for i in range(len(address_ids)):
                addresses.append({'id': address_ids[i], 'id': address_ips[i], 'id':
address_ifnames[i], 'id': address_netmasks[i], 'id': address_devicenames[i], 'id':
address_hostnames[i], 'id': address_subnetids[i]})
                with open(jsonfile, mode='a') as f:
                    json.dump(addresses, f)
                df = pandas.read_json(jsonfile)
                df.dropna(subset=['id'], inplace=True)
                df.to_csv(csvfile, index=False, encoding='utf-8', quoting=csv.QUOTE_ALL, co-
lumnns=['columns'])
    finally:
        ipam_session.removefile(jsonfile)
    return csvfile
```

""" Create list ip / vlan links in csv format """

```
def vlan_ip_lnk_synchro_itop():
    jsonfile = ipam_session.make_tempfile(suff='.json')
    csvfile = ipam_session.make_tempfile(suff='.csv')
    try:
        section_id = ipam_session.get_section_id("id")
```

```

vlan_links = []
vlanlnk_ips = []
vlanlnk_vids = []
vlanlnk_names = []
vlanlnk_ipadd_ids = []
subnet_ids = ipam_session.parse_subnetids(section_id)
for i in subnet_ids:
    data = ipam_session.get_addresses(i)
    if data['code'] != 200:
        continue
    vlanlnk_subnet_vid = ipam_session.get_subnet(i)
    vlanlnk_subnet_vid = vlanlnk_subnet_vid['id']['id']
    if vlanlnk_subnet_vid in ['0', 'null', None]:
        continue
    vlan_id = ipam_session.get_vlan(vlanlnk_subnet_vid)
    ipam_vlan_id = vlan_id['id']['id']
    ipam_vlan_name = vlan_id['id']['id']
    for i in data['data']:
        vlanlnk_vids.append(id)
        vlanlnk_names.append(id)
        vlanlnk_ips.append(i['id'])
        vlanlnk_ipadd_ids.append(i['id'])
    for i in range(len(vlanlnk_ips)):
        vlan_links.append({'id': vlanlnk_ips[i], 'id': vlanlnk_vids[i], 'id': vlan-
lnk_names[i], 'id': vlanlnk_ipadd_ids[i]})
    with open(jsonfile, mode='a') as f:
        json.dump(vlan_links, f)
    df = pandas.read_json(jsonfile)
    df['id'] = df.apply(lambda row: '{0} - {1}'.format(row['id'],row['id']), axis=1)
    df.to_csv(csvfile, index=False, encoding='utf-8', quoting=csv.QUOTE_ALL, co-
lumnns=['columns'])
    os.system('php -q path --auth_user=' + config.get('itop', 'itop_username') + ' --
auth_pwd=' + config.get('itop', 'itop_password') + ' --data_source_id=15 --
separator=, --csvfile=' + csvfile)
    finally:
        ipam_session.removefile(jsonfile,csvfile)
    return csvfile

""" Synchronize interfaces to itop
Attributes:
    itopdeviceattr: iTop device attribute which the address is added. e.g connectable-
ci_id to server and virtualmachine_id to virtualmachine.
    datasourceid: iTop synchronization data source id
"""

def ipaddressesses_synchro_itop(itopdeviceattr,datasourceid):
    try:
        # Get ipv4 section addresses from phpipam
        csvfile = get_ipv4section_addresses()
        df = pandas.read_csv(csvfile)

```

```

df['id'] = df['id']
df['id'] = df['id']
df = df.rename(columns={'columns'})
df['id'] = 'url' + df['id'].astype(str) + '/address-details/' + df['id'].astype(str) + '/'
keep_col = ['columns']
df = df[keep_col]
df.to_csv(csvfile, index=False, quoting=csv.QUOTE_ALL)
os.system('php -q path --auth_user=' + config.get('itop', 'itop_username') + ' --
auth_pwd=' + config.get('itop', 'itop_password') + ' --data_source_id=' + datasourceid
+ ' --separator=', --csvfile=' + csvfile)
finally:
    ipam_session.removefile(csvfile)

```

""" Synchronize all vlans to iTop """

```

def vlans_synchro_itop():
    csvfile = ipam_session.make_tempfile(suff='.csv')
    try:
        vlans_csv = ipam_session.get_all_vlans_csv()
        fetchrest.readcsv(vlans_csv, csvfile)
        fetchrest.defaultsteps(csvfile, orgcol='id', prikey='id')
        fetchrest.mergecolumns(csvfile, 'id', 'number', 'id', separator='-')
        keep_col = ['primary_key', 'id', 'id', 'description']
        fetchrest.keepcolumns(csvfile, keep_col)
        os.system('php -q path --auth_user=' + config.get('itop', 'itop_username') + ' --
auth_pwd=' + config.get('itop', 'itop_password') + ' --data_source_id= --separator=', --
csvfile=' + csvfile)
    finally:
        ipam_session.removefile(vlans_csv, csvfile)

```

```

if __name__ == '__main__':
    # Synchronize physical interfaces
    ipaddressesses_synchro_itop('id', '')
    # Synchronize logical interfaces
    ipaddressesses_synchro_itop('id', '')
    vlans_synchro_itop()
    vlan_ip_lnk_synchro_itop()

```

## Liite 6. ITop asennus

ITop-järjestelmän sovellusvaatimuksia ovat PHP versio 5.3.6 tai uudempi tai MySQL versio 5.0 tai uudempi, lisäksi web-palvelin, Apache, IIS tai mikä tahansa PHP 5 tukeva web-palvelin.

ITop järjestelmä asennettiin Centos 7 Linux -käyttöjärjestelmälle. Web-palvelimeksi asennettiin Apache ja tietokannaksi MariaDB.

Apache asennettiin seuraavalla komennolla:

- `yum install httpd`

Apache:lle tehtiin SSL virtualhost -konfiguraatio, joka käyttää Protacoinin omaa sertifikaattia. Apache virtualhost -konfiguraatiossa määritetään sivun ohjaus HTTPS-yhteydelle, käytettävä sertifikaatti ja iTop-polun sijainti.

PHP ja sen iTop järjestelmän kanssa tarvittavat moduulit asennettiin komennolla:

- `yum install php php-mcrypt php-xml php-cli php-soap php-ldap graphviz php-gd php-pear php-devel httpd-devel pcre-devel gcc make`

ITop käytettävyyden parantamiseksi asennettiin Alternate PHP Cache (APC), joka on avoimenlähdekoodin välimuisti PHP koodille. APC:lla nopeutetaan PHP sovelluksia käyttäen välimuistia, joten ei tarvitse rakentaa sivuston resursseja joka sivun päivityksellä. Aikaisemmassa komennossa asennettiin työkalu pecl, jolla APC asennettiin komennolla:

- `pecl install apc`

Tietokantasovellus MariaDB asennettiin komennolla:

- `yum install mariadb mariadb-server php-mysqld`

Apache- ja MariaDB-palvelut otettiin käynnistyksessä käyttöön ja käynnistettiin komennolla:

- `systemctl enable httpd`
- `systemctl start httpd`

- `systemctl enable mariadb`
- `systemctl start mariadb`

MariaDB-tietokannan tietoturva parannettiin ajamalla sen oma komento, jolla asetetaan root-käyttäjälle salasana, poistetaan root-käyttäjältä mahdollisuus etäyhteydelle, poistetaan anonymous-user käyttäjät ja lopuksi poistetaan testi tietokanta (johon kaikilla käyttäjillä on oikeus). Tämä tehtiin komennolla:

- `mysql_secure_installation`

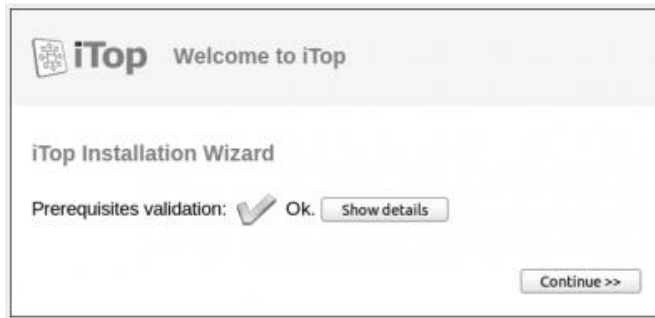
ITop järjestelmän koodit ladattiin iTop:n SourceForge verkkosivulta

<https://sourceforge.net/projects/itop/files/itop>. ITop:sta asennettiin sen hetkinen uusin vakaa versio 2.3.3, joka on tullut 22.12.2016. ITop:n ladattu pakattu paketti purettiin sen tulevaan polkuun `/polku/cmdb/html`.

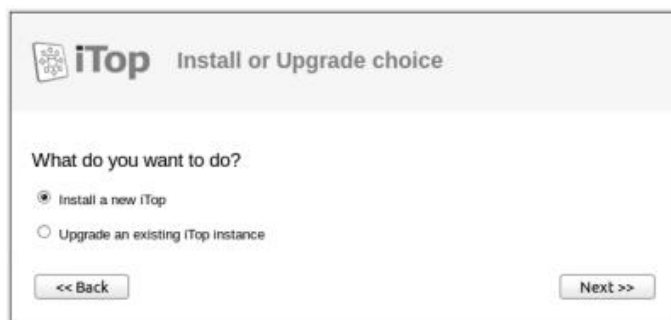
Koska käyttöjärjestelmänä on Centos 7, tarkoittaa, että käytössä on SELinux. Se halutaan pitää päällä tietoturvasyistä ja tämä vaatii SELinux määrittämiä. Haluttiin, että Apache web-palvelin voi ottaa verkkoyhteyden tietokantaan ja tätä varten ajettiin komento:

- `setsebool -P httpd_can_network_connect_db on`

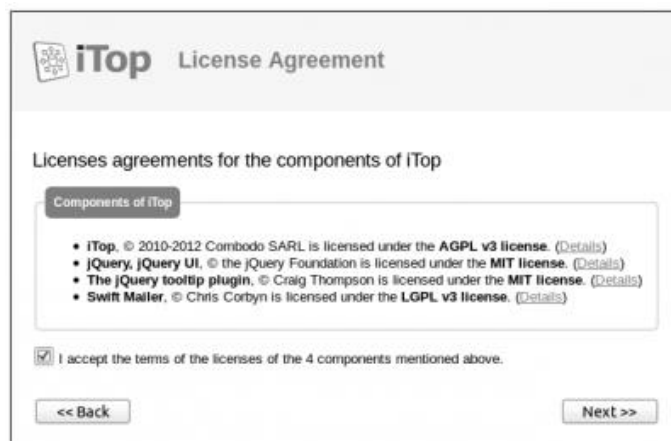
ITop-koodit ovat niille tarkoitettussa polussa, Apache-, MariaDB-, PHP- ja SELinux-konfiguroinnit ovat kunnossa, aloitettiin iTop asennus. Avattiin selaimella palvelimen sivu, joka osoittaa iTop polkuun <https://cmdb.url/itop/>. Sivuston auetessa tuli esiin iTop oma asennus työkalu, ohjattu toiminto. Ohjattu toiminto tarkastaa aluksi, että iTop-järjestelmän asennuksen edellytykset löytyvät (katso kuvio 11). Jos kaikki edellytykset löytyvät valitaan asennettavaksi uusi iTop asennus (katso kuvio 12) ja hyväksytään lisenssi sopimus (katso kuvio 13).



Kuvio 11. ITop ohjattu toiminto, edellytyksien tarkastus



Kuvio 12. ITop ohjattu toiminto, uusi asennus



Kuvio 13. ITop ohjattu toiminto, lisenssien hyväksyntä

Konfiguroitiin tietokantayhteys, serverin nimi, tietokannan tunnistautumisen tiedot ja luotiin uusi tietokanta iTop järjestelmälle (katso kuvio 14).

**iTop Database Configuration**

Configuration of the database connection:

**Database Server Connection**

Server Name:  E.g. "localhost", "dbserver.mycompany.com" or "192.142.10.23"

Login:  The account must have the following privileges on the database: SELECT, INSERT, UPDATE, DELETE, DROP, CREATE, ALTER, CREATE VIEW, SUPER, TRIGGER

Password:

Database server connection Ok.

**Database**

Create a new database:

Use the existing database:

Use a prefix for the tables:

<< Back Next >>

Kuvio 14. ITop ohjattu toiminto, tietokantayhteys

Määritellään iTop järjestelmänvalvojan tili (katso kuvio 15).

**iTop Administrator Account**

Definition of the Administrator Account

**Administrator Account**

Login:

Password:

Confirm password:

Language:

<< Back Next >>

Kuvio 15. ITop ohjattu toiminto, järjestelmänvalvoja tili

Valittiin kieli iTop asennukselle, iTop-järjestelmän url-osoite, johon se on asennettu ja asennettavaksi tuotanto versio puhtaalla tietokannalla (katso kuvio 16).



**iTop Miscellaneous Parameters**

**Additional parameters**

**Default Language**  
 Default Language:

**Application URL**  
 URL:   
 Change the value above if the end-users will be accessing the application by another path due to a specific configuration of the web server.

**Sample Data**

I am installing a **demo or test** instance, populate the database with some demo data.

I am installing a **production** instance, create an empty database to start from.

Kuvio 16. ITop ohjattu toiminto, lisävalinnat

Configuration Management options kohdassa valittiin asennettavaksi kaikki 5 moduulia että saadaan kaikki moduulit asennettua (katso kuvio 17).

**iTop Configuration Management options**

The options below allow you to configure the type of elements that are to be managed inside iTop.

- Configuration Management Core**  
 All the base objects that are mandatory in the iTop CMDB: Organizations, Locations, Teams, Persons, etc.
- Data Center Devices**  
 Manage Data Center devices such as Racks, Enclosures, PDUs, etc.
- End-User Devices**  
 Manage devices related to end-users: PCs, Phones, Tablets, etc.
- Storage Devices**  
 Manage storage devices such as NAS, SAN Switches, Tape Libraries and Tapes, etc.
- Virtualization**  
 Manage Hypervisors, Virtual Machines and Farms.

Kuvio 17. ITop ohjattu toiminto, asennettavat iTop moduulit

Configuring Service Management kohdassa valittiin Service Management for Enterprises, koska tarve on vain yhdelle infrastruktuurille (katso kuvio 18).

**iTop Service Management options**

Select the choice that best describes the relationships between the services and the IT infrastructure in your IT environment.

- Service Management for Enterprises**  
Select this option if the IT delivers services based on a shared infrastructure. For example if different organizations within your company subscribe to services (like Mail and Print services) delivered by a single shared backend.
- Service Management for Service Providers**  
Select this option if the IT manages the infrastructure of independent customers. This is the most flexible model, since the services can be delivered with a mix of shared and customer specific infrastructure devices.

<< Back Next >>

Kuvio 18. ITop ohjattu toiminto, palvelunhallinta

Tickets Management options kohdassa määritettiin, että ei asenneta tiketin hallintaa, sille ei ole tarvetta (katso kuvio 19). Configuring Change Management, valittiin myös ei muutoksenhallintaa, ei ole tarvetta (katso kuvio 20). Ei valittu lisäominaisuuksia ITIL-tiketinhallinnalle (katso kuvio 21).

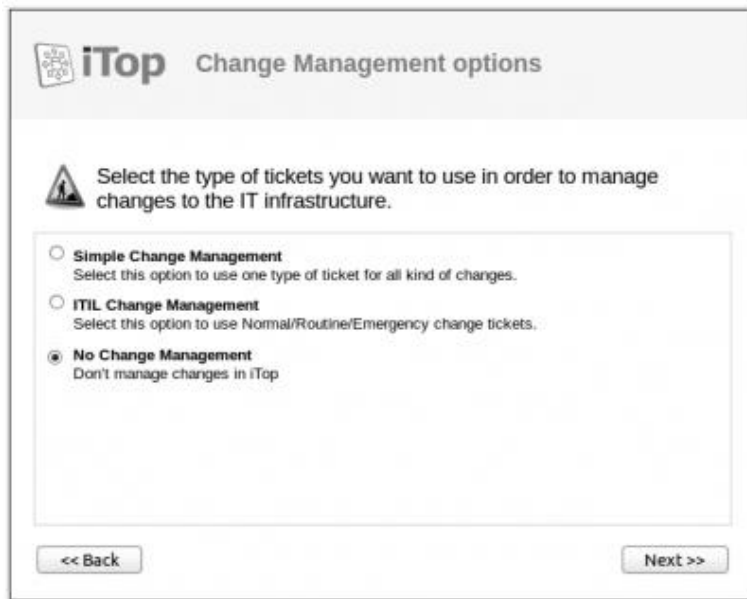
**iTop Tickets Management options**

Select the type of tickets you want to use in order to respond to user requests and incidents.

- Simple Ticket Management**  
Select this option to use one single type of tickets for all kind of requests.
- ITIL Compliant Tickets Management**  
Select this option to have different types of ticket for managing user requests and incidents. Each type of ticket has a specific life cycle and specific fields
  - User Request Management  
Manage User Request tickets in iTop
  - Incident Management  
Manage Incidents tickets in iTop
- No Tickets Management**  
Don't manage incidents or user requests in iTop

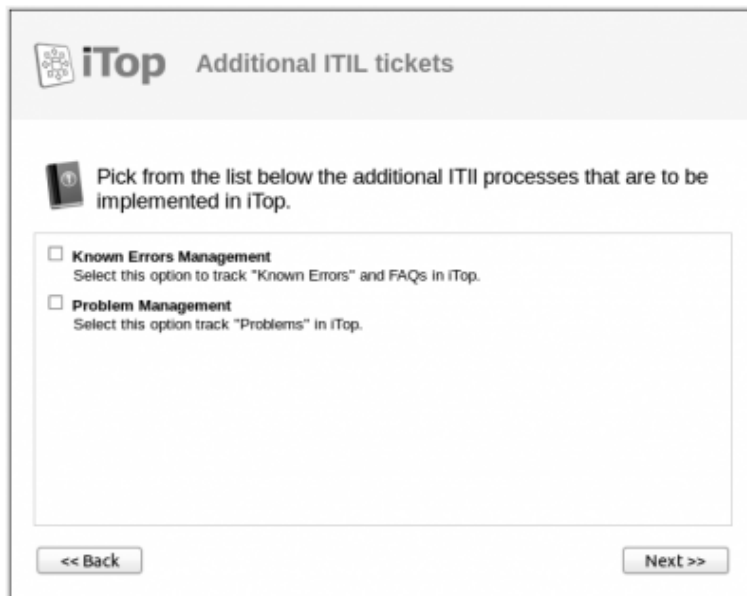
<< Back Next >>

Kuvio 19. ITop ohjattu toiminto, tiketinhallinta



The screenshot shows the 'iTop Change Management options' configuration window. At the top left is the iTop logo. The title is 'Change Management options'. Below the title is a warning icon and the instruction: 'Select the type of tickets you want to use in order to manage changes to the IT infrastructure.' There are three radio button options: 'Simple Change Management' (with subtext 'Select this option to use one type of ticket for all kind of changes.'), 'ITIL Change Management' (with subtext 'Select this option to use Normal/Routine/Emergency change tickets.'), and 'No Change Management' (with subtext 'Don't manage changes in iTop'). The 'No Change Management' option is selected. At the bottom are '<< Back' and 'Next >>' buttons.

Kuvio 20. ITop ohjattu toiminto, muutoksenhallinta



The screenshot shows the 'iTop Additional ITIL tickets' configuration window. At the top left is the iTop logo. The title is 'Additional ITIL tickets'. Below the title is a document icon and the instruction: 'Pick from the list below the additional ITIL processes that are to be implemented in iTop.' There are two checkbox options: 'Known Errors Management' (with subtext 'Select this option to track "Known Errors" and FAQs in iTop.') and 'Problem Management' (with subtext 'Select this option track "Problems" in iTop.'). Both options are unchecked. At the bottom are '<< Back' and 'Next >>' buttons.

Kuvio 21. ITop ohjattu toiminto, lisää ITIL valinnat

Viimeisenä valittiin kaikki asennettavat itse tehdyt moduulit, jotka mm. muokkaavat olemassa olevia luokkia ja lisää uusia ja aloitetaan iTop-järjestelmän asennus. Onnistuneen asennuksen jälkeen siirryttiin iTop:iin.

## Liite 7. Osaprojekti moduulin datamodel.add\_subproject\_class

```

<?xml version="1.0" encoding="UTF-8"?>
<itop_design xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0">
  <classes>
    <class id="Subproject" _delta="define">
      <parent>FunctionalCI</parent>
      <properties>
        <category>bizmodel,searchable</category>
        <abstract>>false</abstract>
        <key_type>autoincrement</key_type>
        <db_table>subproject</db_table>
        <db_key_field>id</db_key_field>
        <db_final_class_field/>
        <naming>
          <attributes>
            <attribute id="name"/>
          </attributes>
        </naming>
        <display_template/>
        <icon>images/business-process.png</icon>
        <reconciliation>
          <attributes>
            <attribute id="name"/>
            <attribute id="org_id"/>
            <attribute id="organization_name"/>
            <attribute id="id"/>
          </attributes>
        </reconciliation>
      </properties>
      <fields>
        <field id="id" xsi:type="AttributeExternalKey">
          <sql>id</sql>
          <target_class>class</target_class>
          <is_null_allowed>>true</is_null_allowed>
          <on_target_delete>DEL_MANUAL</on_target_delete>
        </field>
        <field id="id" xsi:type="AttributeExternalField">
          <extkey_attcode>id</extkey_attcode>
          <target_attcode>name</target_attcode>
        </field>
        <field id="id" xsi:type="AttributeExternalKey">
          <sql>id</sql>
          <target_class>class</target_class>
          <is_null_allowed>>true</is_null_allowed>
          <on_target_delete>DEL_MANUAL</on_target_delete>
      </fields>
    </class>
  </classes>
</itop_design>

```



```

</field>
<field id="id" xsi:type="AttributeURL">
  <sql>id</sql>
  <default_value/>
  <is_null_allowed>true</is_null_allowed>
  <target>_blank</target>
</field>
</fields>
<presentation>
<details>
  <items>
    <item id="col:col1">
      <rank>10</rank>
      <items>
        <item id="fieldset:Subproject:vfinfo">
          <rank>10</rank>
          <items>
            <item id="id">
              <rank>10</rank>
            </item>
            <item id="id">
              <rank>20</rank>
            </item>
            <item id="name">
              <rank>30</rank>
            </item>
            <item id="id">
              <rank>40</rank>
            </item>
            <item id="id">
              <rank>50</rank>
            </item>
            <item id="id">
              <rank>60</rank>
            </item>
            <item id="id">
              <rank>70</rank>
            </item>
            <item id="id">
              <rank>80</rank>
            </item>
            <item id="id">
              <rank>90</rank>
            </item>
            <item id="id">
              <rank>100</rank>
            </item>
            <item id="id">
              <rank>110</rank>
            </item>
          </items>
        </item>
      </items>
    </item>
  </items>
</details>
</presentation>

```

```
</item>
  <item id="org_id">
    <rank>120</rank>
  </item>
</items>
</item>
</items>
</item>
</items>
</details>
<search>
  <items>
    <item id="name">
      <rank>10</rank>
    </item>
    <item id="org_id">
      <rank>20</rank>
    </item>
    <item id="id">
      <rank>30</rank>
    </item>
    <item id="id">
      <rank>40</rank>
    </item>
    <item id="id">
      <rank>50</rank>
    </item>
  </items>
</search>
<list>
  <items>
    <item id="id">
      <rank>10</rank>
    </item>
    <item id="id">
      <rank>20</rank>
    </item>
    <item id="id">
      <rank>30</rank>
    </item>
    <item id="id">
      <rank>40</rank>
    </item>
    <item id="id">
      <rank>50</rank>
    </item>
  </items>
</list>
</presentation>
```

```

<methods>
  <method id="GetRelationQueries">
    <comment>/**
* Placeholder for backward compatibility (iTop &lt;= 2.1.0)
* in case an extension attempts to redefine this function...
*/</comment>
    <static>true</static>
    <access>public</access>
    <type>Overload-DBObject</type>
    <code><![CDATA[ public static function GetRelationQue-
ries($sRelCode){return parent::GetRelationQueries($sRelCode);} ]]></code>
  </method>
</methods>
<relations>
  <relation id="impacts">
    <neighbours>
      <neighbour id="id">
        <attribute>id</attribute>
      </neighbour>
      <neighbour id="id">
        <attribute>id</attribute>
      </neighbour>
    </neighbours>
  </relation>
</relations>
</class>
</classes>
<menus>
</menus>
</itop_design>

```



Osaprojekti moduulin en.dict.add\_subproject\_class

```
<?php
/**
 * Localized data
 *
 * @copyright Copyright (C) 2013 XXXXX
 * @license http://opensource.org/licenses/AGPL-3.0
 */

Dict::Add('EN US', 'English', 'English', array(
    'Class:Subproject' => 'Subproject',
    'Class:Subproject/Attribute:project' => 'Project',
    'Class:Subproject/Attribute:subproject' => 'Subproject id',
    'Class:Subproject/Attribute:name' => 'Subproject name',
    'Class:Subproject/Attribute:projectstatus' => 'Project status',
    'Class:Subproject/Attribute:personnel' => 'Subproject manager',
    'Class:Subproject/Attribute:customer' => 'Customer',
    'Class:Subproject/Attribute:datestartend' => 'Project valid',
    'Class:Subproject/Attribute:subprojectstatus' => 'Subproject status',
    'Class:Subproject/Attribute:subprojectstartend' => 'Subproject valid',
    'Class:Subproject/Attribute:url' => 'Protacon url',
    'Class:Subproject/Attribute:subprojectstatus' => 'Subproject status',
    'Class:Subproject/Attribute:projectmanager' => 'Project manager',
));
// Add translation for Fieldsets

Dict::Add('EN US', 'English', 'English', array(
    'Subproject:vfinfo' => 'VF information',
));
?>
```

## Liite 8. PhpIPAM asennus

PhpIPAM-asennusohje Centos 7 -käyttöjärjestelmälle. PhpIPAM edellytyksiä ovat Apache-palvelin, php phpIPAM:a tukevilla moduuleilla ja tietokanta. Huomiona, tässä työssä ei tarvinnut phpIPAM:n kanssa erikseen asentaa Apache-palvelinta tai tietokantaa, koska se asennettiin iTop:n rinnalle samalle palvelimelle, josta nämä löytyivät jo. Tässä ohjeessa käydään kuitenkin nyt kaikki vaiheet, jotka tarvitaan phpIPAM:n asennukseen.

Tarvittavat edellytykset saadaan asennettua komennolla:

- `yum install httpd mariadb-server php php-cli php-gd php-common php-ldap php-pdo php-pear php-snmp php-xml php-mysql git`

Määritellään Apache konfiguraatioon palvelimen nimi, käytetään omaa FQDN.

Apache-konfiguraatioon täytyy määrittää, että sallitaan `mod_rewrite`. Tämä saadaan aikaiseksi Apache-konfiguraatiossa `/etc/httpd/conf/httpd.conf`:

```
<Directory "/var/www/html">
    Options FollowSymLinks
    AllowOverride all
    Order allow,deny
    Allow from all
</Directory>
```

Käynnistetään Apache, MariaDB ja otetaan käyttöön myös koneen käynnistyessä:

- `systemctl start httpd`
- `systemctl enable httpd`
- `systemctl start mariadb`
- `systemctl enable mariadb`

Asetetaan SELinux kontekstit kuntoon phpIPAM:lle

- `semanage fcontext -a -t httpd_sys_content_t '/path/to/ipam/html(/.*)?'`

- `semanage fcontext -a -t httpd_sys_rw_content_t '/path/to/ipam/html/app/admin/import-export/upload(/.*)?'`
- `semanage fcontext -a -t httpd_sys_rw_content_t '/path/to/ipam/html/app/subnets/import-subnet/upload(/.*)?'`
- `semanage fcontext -a -t httpd_sys_rw_content_t '/path/to/ipam/html/css/1.3/images/logo(/.*)?'`
- `restorecon -rv /path/to/ipam/html`

Ladataan phpIPAM tiedostot sen github sivustolta sille tarkoitettuun polkuun:

- `cd /path/to/ipam/html`
- `git clone https://github.com/phpipam/phpipam.git .`
- `git checkout 1.2`

Konfiguroidaan tietokanta yhteys, ensin tarvitsee kopioida esimerkki konfiguraatio tiedosto sen tiedostopolussa `config.php` nimelle, jota phpIPAM käyttää:

- `cd confing cp config.dist.php config.php`

Tietokantayhteydelle määritellään käyttäjätunnus ja salasana, älä käytä root-tunnusta. PhpIPAM:lle määritetty käyttäjä käyttää tätä myöhempisiin yhteyksiin.

Seuraavaksi avataan selaimella palvelimen osoitteeseen `https://ipam_osoite/` ja sivustolla avautuu phpIPAM:n asennuksen ohjattu toiminto, jossa määritetään root:n tietokanta tunnukset, luodaan uusi tietokanta phpIPAM:lle ja viimeistellään asennus.