

## Palvelimien ja sovellusten provisioinnin automatisointi

Joni Korhonen

Opinnäytetyö  
Tietojenkäsittelyn  
koulutusohjelma  
2016



<b>Tekijä(t)</b> Joni Korhonen	
<b>Koulutusohjelma</b> Tietojenkäsittelyn koulutusohjelma	
<b>Opinnäytetyön otsikko</b> Palvelimien ja sovellusten provisioinnin automatisointi	<b>Sivu- ja liitesivumäärä</b> 31
<b>Opinnäytetyön otsikko englanniksi</b> Automated provisioning of servers and services	
<p>Tämän opinnäytetyön tarkoituksena oli edistää asiakkaan automatisoinnin kehitystyötä. Lähtökohtana oli selvittää, missä vaiheessa automatisoinnin kehitystyö oli ja suunnitella sekä toteuttaa seuraavat automatisoidut vaiheet. Automatisoitujen vaiheiden lisäksi opinnäytetyössä pohditaan, miten automatisoinnin kehitystyötä voidaan viedä vielä entistä pidemmälle.</p> <p>Teoriaosuudessa käydään läpi käytettyjä tekniikoita ja teknologioita sekä esitellään yleisellä tasolla asiakkaan tietojärjestelmäarkkitehtuuria.</p> <p>Automatisoinnin kehitystyö kohdistui virtuaalikoneiden provisiointiin, sovelluksien asentamiseen ja palveluiden tarjoamiseen välityspalvelimen avulla. Empiirisessä osuudessa kuvataan, kuinka toteutettiin ”virtuaalikoneiden provisiointi itsepalveluna” -toiminnallisuus sekä automatisoitu välityspalvelimen konfigurointi. Lisäksi pohditaan, miten automatisoituja vaiheita voidaan hyödyntää, kun automatisoinnin kehitystyötä viedään eteenpäin ja kuinka voidaan toteuttaa virtuaalikoneiden ajastettu uudelleenprovisiointi kerran viikossa.</p> <p>Tuloksena syntyi kaksi uutta automatisoitua vaihetta, joiden avulla säästetään aikaa, kun uusia palveluita kehitetään. Lisäksi automatisoitujen vaiheiden myötä kehitystyötä voidaan jatkaa pidemmälle.</p> <p>Lopuksi esitellään jatkokehitysehdotuksena ratkaisu, kuinka automatisoinnin kehitystyötä voidaan viedä vielä entistä pidemmälle. Jatkokehitysehdotus liittyy tuotteeseen, jonka avulla voidaan hallita prosesseja. Tuotteen avulla automatisoinnin kehitystyöhön liittyvät vaiheet voidaan yhdistää yhdeksi prosessiksi ja mahdollistaa virtuaalikoneiden ajastettu uudelleenprovisiointi kerran viikossa.</p>	
<b>Asiasanat</b> Automatisointi, prosessinhallinta, virtuaalikone	

## Sisällys

1	Johdanto .....	1
1.1	Lyhenteet ja termit .....	1
2	Virtualisoidun palvelinympäristön automatisointi .....	3
2.1	Virtualisointi .....	3
2.2	Pilvipalvelut .....	4
2.2.1	Pilvipalveluiden tarjoaminen .....	5
2.2.2	Pilvipalvelumallit .....	6
2.3	Virtualisoidun palvelinympäristön hallinta .....	7
2.3.1	Virtualisointialusta .....	8
2.3.2	VMware vSphere .....	8
2.3.3	Pilvipalveluiden hallinta .....	9
2.4	Sovelluksien ja palveluiden hallinta .....	10
2.4.1	Konfiguraatioiden hallinta .....	11
2.4.2	Palveluiden tarjoaminen .....	11
2.5	IT-infrastruktuurin automatisointi .....	13
2.5.1	IT-operaatioiden hallinta .....	13
2.5.2	ServiceNow .....	13
3	Asiakkaan virtualisoitu palvelinympäristö .....	18
4	Automatisoinnin kehitystyö .....	19
4.1	Nykyisen tilanteen kartoitus .....	19
4.2	Virtuaalikoneiden provisiointi itsepalveluna .....	20
4.3	Automatisoitu kuormantasaus .....	21
4.4	Virtuaalikoneiden orkestroitu uudelleenprovisiointi viikoittain .....	22
5	Tulokset .....	24
5.1	Hyödyt kehittäjille .....	24
5.2	Hyödyt loppukäyttäjille .....	24
6	Jatkokehitysehdotus .....	26
7	Pohdinta .....	28
	Lähteet .....	29

# 1 Johdanto

Asiakkaan IT-palveluille ollaan toteuttamassa automatisoinnin kehitystyötä, tämä opinnäytetyöprojekti on osa tätä kokonaisuutta. Automatisoinnin kehitystyön tarkoituksena on tehostaa palveluiden kehitystä. Tavoitteena on muun muassa nopeuttaa kehitysprosessia, säästää kehittäjien aikaa ja parantaa tietoturvaa automatisoinnin avulla.

Palveluiden kehitysprosessi koostuu käytännössä kolmesta vaiheesta, virtuaalikoneen provisioinnista, sovelluksen asentamisesta virtuaalikoneelle ja välityspalvelimen konfiguroimisesta. Prosessin vaiheista ainoastaan sovelluksen asentaminen virtuaalikoneelle on automatisoitu. Opinnäytetyön tarkoituksena on suunnitella ja toteuttaa virtuaalikoneen automatisoitu provisiointi sekä välityspalvelimen automatisoitu konfigurointi. Lisäksi toteutusvaiheen jälkeen on tarkoitus pohtia, miten voidaan toteuttaa virtuaalikoneiden ajastettu uudelleenprovisiointi kerran viikossa. Lopuksi esitellään jatkokehitysehdotuksen muodossa ratkaisu, kuinka tämä voidaan toteuttaa.

Seuraavassa luvussa eli teoriaosuudessa esitellään projektin kannalta olennaisia tekniikoita ja teknologioita. Luvussa käydään yleisellä tasolla läpi, mitä tarkoitetaan virtualisoinnilla ja mitä ovat pilvipalvelut, sekä esitellään virtualisoidun palvelinympäristön hallintatyökalut, joita käytetään tässä ympäristössä. Lisäksi käydään läpi sovelluksien ja palveluiden hallintatyökaluja, joiden avulla voidaan automatisoida välityspalvelimen konfigurointi. Teoriaosuuden lopuksi esitellään jatkokehitysehdotukseen liittyvä tuote. Teoriaosuuden jälkeen kuvataan yleisellä tasolla asiakkaan tietojärjestelmäarkkitehtuuri ja mihin osuuteen automatisoinnin kehitystyö siinä kohdistuu.

Empiirisessä osuudessa kuvataan, kuinka virtuaalikoneen automatisoitu provisiointi sekä välityspalvelimen automatisoitu konfigurointi toteutettiin. Lisäksi kuvataan, mitä virtuaalikoneiden ajastettu uudelleenprovisiointi kerran viikossa edellyttäisi. Lopuksi esitellään jatkokehitysehdotuksena ratkaisu, kuinka se voidaan toteuttaa.

## 1.1 Lyhenteet ja termit

Isäntäkone (Host)

Kone jolla virtuaalikone ohjelmisto pyörii.

Virtuaalikone (Virtual Machine)

Ohjelmisto joka toimii samaan tapaan kuin fyysinen kone.

Instanssi (instance/Cloud Instance)

Virtuaalikone joka perustuu ennalta määrättyyn kuvaan ja joka on skaalattavissa muun muassa CPU:n, muistin ja verkkoliitännän osalta.

Tarjoaja (Provider)

Kone joka hallitsee useita virtuaalikoneita jotka sijaitsevat usealla isäntäkoneella.

Agentti

Ohjelma joka toimii agenttina käyttäjälle tai toiselle ohjelmalle. Se toteuttaa sille ohjelmoituja työtehtäviä.

Domain Name System (DNS)

Palvelu joka muuntaa verkkotunnuksia IP-osoitteiksi.

Dynamic Host Configuration Protocol (DHCP)

Protokolla joka asettaa dynaamisia IP-osoitteita verkossa oleville laitteille.

Asiakasohjelma

Ohjelma joka käyttää toista ohjelmaa

## 2 Virtualisoidun palvelinympäristön automatisointi

Tässä luvussa kuvataan teknologioita ja menetelmiä, jotka liittyvät virtualisoituun palvelinympäristöön, sovelluksien ja palveluiden hallintaan sekä automatisointiin. Luvussa käydään läpi muun muassa, mitä on virtualisointi ja mitä ovat pilvipalvelut sekä esitellään virtualisointituotteet jotka ovat asiakkaalla jo ennestään käytössä. Lisäksi käydään läpi välineitä, joiden avulla sovelluksien konfigurointi ja palveluiden tarjoaminen voidaan automatisoida. Lopuksi esitellään jatkokehitysehdotukseen liittyvä tuote, jonka avulla prosessin eri vaiheet voidaan nivoa yhteen ja tehdä koko prosessista automatisoitu. Luvut 2.1 – 2.4 liittyvät opinnäytetyön toiminnalliseen osuuteen, luku 2.5 käsittelee jatkokehitysehdotukseen liittyvää ratkaisua.

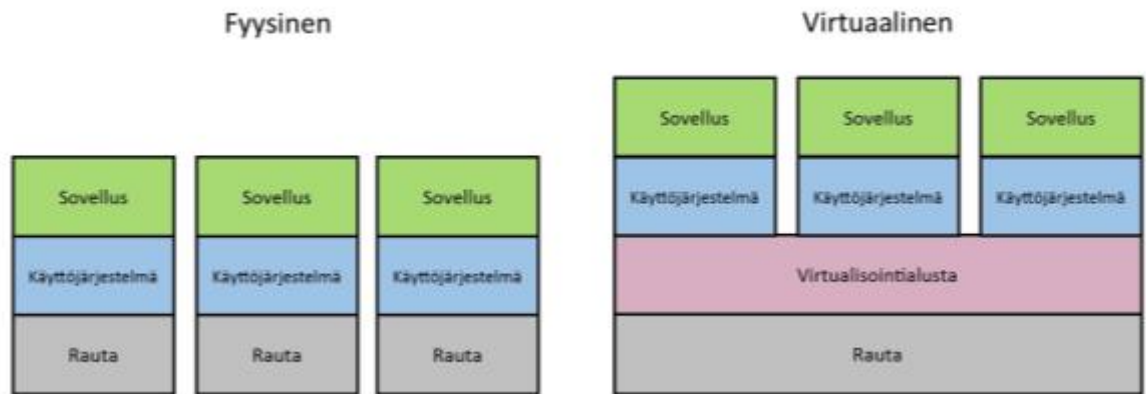
### 2.1 Virtualisointi

Virtualisointi on tekniikka, jonka avulla fyysinen resurssi voidaan jakaa useiksi loogisiksi resursseiksi, jotka toimivat kuten fyysiset resurssit. Virtualisointitekniikkaa voidaan soveltaa monessa eri asiassa, esimerkiksi palvelin-, muistivarasto- ja verkkovirtualisoinnissa. Tässä opinnäytetyössä keskitytään palvelinvirtualisointiin. Tässä yhteydessä virtualisoinnilla tarkoitetaan virtuaalikoneen luomista, joka emuloi täysin toiminnaltaan fyysistä tietokonetta käyttöjärjestelmää myöten. Virtualisointiin tarkoitettujen ohjelmistojen avulla yhdellä fyysisellä palvelimella voidaan ajaa useita virtuaalikoneita.

Virtualisoidun palvelinarkkitehtuurin hyödyt verrattuna fyysiseen palvelinarkkitehtuuriin ovat:

- konesalin pienemmät käyttökulut
- lyhyet häiriöajat
- tehokkuuden ja responsiivisuuden lisääntyminen
- nopeammat ja yksinkertaisemmat provisioinnit
- skaalautuvuus.

Fyysisen ja virtuaalisen palvelinarkkitehtuurin eroja voidaan helpoiten kuvata kuvion avulla. Kuvitellaan yksinkertainen palvelu, joka koostuu kolmesta sovelluksesta. Kuviossa 1 on kuvattu miltä palvelun palvelinarkkitehtuuri voisi näyttää fyysisenä sekä virtuaalisena.



Kuvio 1. Virtuaalinen ja fyysinen palvelinarkkitehtuuri

Virtuaalisen ja fyysisen palvelinarkkitehtuurin suurin ero on fyysisen palvelimen resurssien käyttö. Fyysisessä palvelinarkkitehtuurissa jokainen sovellus ja käyttöjärjestelmä edellyttää oman fyysisen palvelimen. Fyysisen palvelimen kaikki resurssit eivät luultavasti ole kuitenkaan koko ajan hyötykäytössä. Virtualisoinnin avulla on tarkoitus saada kaikki fyysisen palvelimen resurssit hyötykäyttöön. Virtualisointialustan tarkoituksena on jakaa fyysisen palvelimen resursseja sen päällä pyöriville käyttöjärjestelmille ja sovelluksille. Virtualisoinnin avulla palvelun käyttökuluja voidaan pienentää, koska usean fyysisen palvelimen sijaan voidaan käyttää yhtä tehokkaampaa palvelinta. (Sajan 2016.)

## 2.2 Pilvipalvelut

Pilvipalvelu ja virtualisointi sekoitetaan käsitteinä usein keskenään. Niiden määritelmät eroavat kuitenkin huomattavan paljon toisistaan. Virtualisointi on teknologia, ja pilvipalveluilla taas tarkoitetaan palveluita, jotka hyödyntävät virtualisointitekniikkaa. Täysin virtualisoitu infrastruktuuri ei vielä tarkoita pilvipalveluita. (Martin 2013).

Martin (2013) mukaan pilvipalveluiden määritelmään kuuluu muun muassa, että

- palvelut ovat tarjolla, kun tarvitaan (on demand)
- palveluita on mahdollista käyttää usealle eri päätelaitteella
- palvelut ovat skaalautuvia liikenteen kuormituksen mukaan
- palveluiden tilaa monitoroidaan ja mitataan.

Pilvipalveluita ovat käytännössä internetin välityksellä käyttäjälle tarjotut erilaiset palvelut ja ohjelmat. Niitä käytetään pääosin selaimen kautta, ja ne ovat käytettävissä myös mobiilissa.

### 2.2.1 Pilvipalveluiden tarjoaminen

Pilvipalveluita voidaan tarjota omasta konesalista käsin tai kolmannen osapuolen toimesta. Lisäksi ne voidaan jakaa kahteen ryhmään, yksityiseen ja julkiseen pilveen. Yksityisellä pilvellä tarkoitetaan, että infrastruktuuri, jonka päällä pilvipalveluita ajetaan, on ensisijaisesti varattu kyseistä pilvipalvelua varten. Esimerkiksi kun pilvipalveluita tarjotaan omasta konesalista, sanotaan että pilvipalvelut ovat yksityisessä pilvessä. Kolmannen osapuolen toimesta pilvipalvelut voivat olla yksityisessä tai julkisessa pilvessä. Tässä yhteydessä yksityisen pilven määritelmä on sama: kolmas osapuoli varaa oman infrastruktuurin pilvipalveluille, joita halutaan tarjota yksityisestä pilvestä. Julkisella pilvellä tarkoitetaan, että pilvipalvelut ovat samoilla kolmannen osapuolen jaetuilla palvelimilla, joilla muidenkin asiakkaiden pilvipalvelut ovat. (Rouse 2015.)

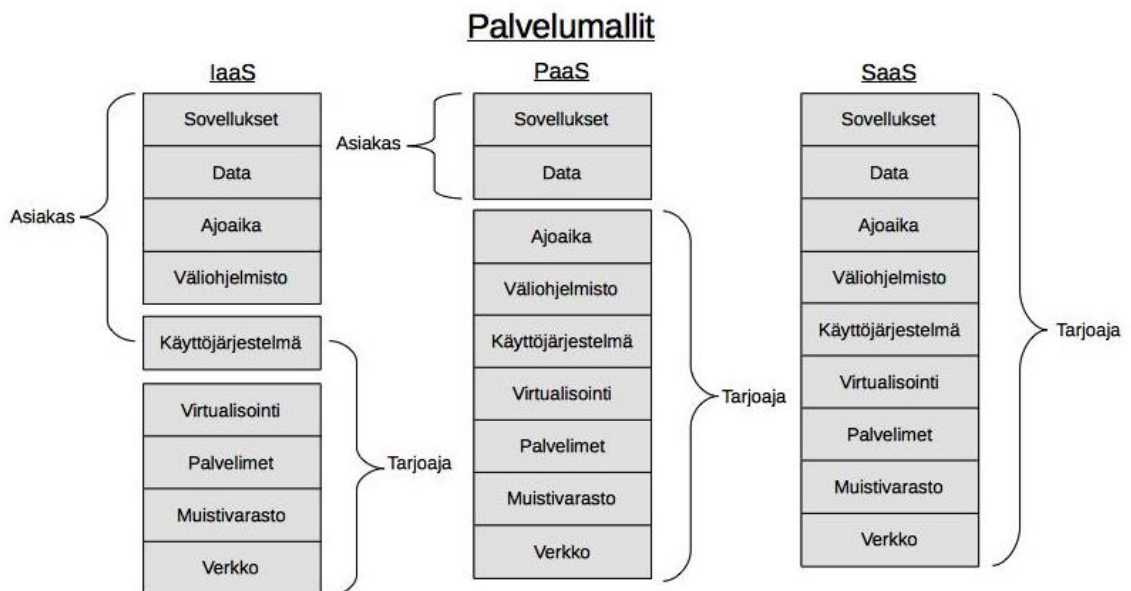
Sekä yksityisessä että julkisessa pilvessä sijaitsevat pilvipalvelut täyttävät edellä mainitut pilvipalveluiden määritelmät. Yksityisen ja julkisen pilven suurimmat erot ovat ne, kuinka paljon pilvipalveluille on varattu infrastruktuuriresursseja ja miten resursseja jaetaan. Pilvipalvelun tarjoaman palvelun luonne määrittää suurilta osin sen, kumpaa mallia tulee soveltaa. Yksityinen pilvi tarjoaa paremman tietoturvan palveluille, koska infrastruktuurin resursseja ei ole jaettu kuten julkisessa pilvessä. Julkisen pilven jaetuilla palvelimilla pilvipalveluiden tarjonta on kuitenkin halvempaa ja resursseja voidaan nopeasti saada lisää. (Rouse 2015.)

Tarjottavien palveluiden luonteesta riippuen kumpaakin mallia voidaan soveltaa myös yhdessä, silloin puhutaan hybridipilvestä. Tämä tarkoittaa sitä, että osa yrityksen pilvipalveluista voi sijaita yksityisessä pilvessä ja osa julkisessa pilvessä. (Hybrid cloud. 2015.) Hybridipilvi mahdollistaa menetelmän, jonka avulla yksityisestä pilvestä tarjottujen palveluiden suorituskykyä ja saatavuutta voidaan parantaa tilanteen niin vaatiessa. Menetelmän periaatteena on siirtää palvelu tai osa sen toiminnasta yksityisestä pilvestä julkiseen pilveen. Sen on tarkoitus olla automatisoitu prosessi, jota sovelletaan silloin kun palvelun käyttöaste kasvaa yllättäen yli yksityisen pilven käyttökapasiteetin. Menetelmästä käytetään nimeä Cloud bursting. Se soveltuu hyvin järjestelmiin, jotka eivät sisällä monimutkaisia integraatioita toisiin järjestelmiin tai muuhun IT-infrastruktuuriin. Koska siirto tapahtuu julkiseen pilveen, tietoturva tulee ottaa huomioon. Kriittisiä tai arkaluontoista dataa käsitteleviä palveluita ei välttämättä kannata siirtää julkiseen pilveen. (Rouse 2014.)



## 2.2.2 Pilvipalvelumallit

Pilvipalvelut voidaan jakaa pilvipalvelumalleihin riippuen, mitä ominaisuuksia palvelut tarjoavat asiakkaille. Pääasiassa pilvipalveluiden ominaisuudet perustuvat yhdeksään komponenttiin, jotka ovat muun muassa osa IT-infrastruktuuria ja joita tarvitaan palveluiden tarjoamiseen. Komponentteja ovat muun muassa sovellukset, data ja sovellusten ajoaika. Muita komponentteja ovat väliohjelmistot ja käyttöjärjestelmät. Väliohjelmistoilla tarkoitetaan ohjelmistoja, joita tarvitaan käyttöjärjestelmän lisäksi sovelluksen toiminnan takaamiseen. Taustalla olevia komponentteja ovat virtualisointialustat ja palvelimet, joiden päälle virtualisointialustat on asennettu. Lisäksi komponentteja ovat myös muistivarastot ja verkot. Kolme yleisintä pilvipalvelumallia ovat infrastruktuurin, sovellusalustan ja sovelluksen tarjoaminen pilvipalveluina. Katso kuviota 2. (Lau 2011.)



Kuvio 2. SaaS, PaaS ja IaaS -pilvipalvelumallit (Lau 2011.)

Infrastruktuuri palveluna eli IaaS (Infrastructure as a Service) -mallin periaatteena on tarjota asiakkaalle IT-infrastruktuuriin kuuluvia komponentteja. Komponentteja ovat virtualisointi, palvelimet, muistivarastot ja verkot. Käyttöjärjestelmän hallinta voi olla asiakkaan tai tarjoajan hallinnassa. Infrastruktuuri palveluna -mallille ominaisia piirteitä ovat muun muassa asiakkaalle tarjottavat automatisointiominaisuudet ja palveluiden hyvä skaalautuvuus tarvittaessa. Maksut perustuvat yleensä palvelun käytön määrään. Infrastruktuuri palveluna -malli soveltuu hyvin tilanteeseen, jolloin tarvitaan väliaikaisia lisäresursseja. Väliaikaisia lisäresurssitarpeita voivat aiheuttaa esimerkiksi nykyisen järjestelmän liiallinen kuormitus tai muu ylimääräinen resurssitarve. (Rouse 2015.)

Sovellusalusta palveluna eli PaaS (Platform as a Service) -mallin periaatteena on tarjota asiakkaalle laitteistot ja ohjelmistot sovelluskehitykseen tai sen tueksi. Asiakkaan hallinnoitavaksi jäävät sovellukset ja data. Sovelluksia ajetaan tarjoajan tarjoamassa ympäristössä. Sovellusalusta palveluna -mallille ominaisia piirteitä ovat mm. asiakkaalle tarjottava IT-tuki ja web-pohjainen käyttöliittymä. Myös sovellusalusta palveluna -mallin maksut perustuvat yleensä palvelun käytön määrään. Sovellusalusta palveluna -mallia kannattaa soveltaa, kun halutaan keskittää tiimin voimavarat itse sovelluksen kehitykseen. PaaS tarjoaja hallinnoi sovelluksen ajoympäristöä, tietokantoja ja muita resursseja. (Rouse 2015.)

Sovellus palveluna eli SaaS (Software as a Service) -mallin periaatteena on tarjota asiakkaalle sovellus/palvelu kokonaisuudessaan. Sovellus palveluna -mallin ominaisuuksia ovat muun muassa saatavuus, helppo hallinnointi ja automaattiset päivitykset. Esimerkiksi jokin yrityksen sisäiseen käyttöön tarvittava palvelu voidaan ostaa SaaS-palveluna. Siten yrityksen omia resursseja ei tarvita sen kehittämiseen vaan ne voidaan keskittää yrityksen tarjoamien palveluiden kehittämiseen. Sovellus palveluna – mallin maksut perustuvat myös palvelun käytön määrään. Lisäksi palvelun tilaus voidaan yleensä lopettaa tarvittaessa nopeasti ja siirtyä käyttämään esimerkiksi jotain toista palvelua, koska integraatioita SaaS-palveluihin ei ole. (Rouse 2016.)

### **2.3 Virtualisoidun palvelinympäristön hallinta**

Virtualisoidun palvelinympäristön hallinta muuttuu vaikeammaksi infrastruktuurin laajentuessa. Kun esimerkiksi virtuaalikoneiden määrä kasvaa ja yhteydet lisääntyvät, infrastruktuuri muuttuu monimutkaisemmaksi. Virtualisointiin tarkoitettujen tuotteiden avulla on tarkoitus yksinkertaistaa ja helpottaa virtuaalisen ympäristön hallintaa. Tuotteita on monenlaisia erilaisiin tarkoituksiin ja vaatimuksiin. Lähtökohta virtualisoidun ympäristön pystyttämiseen on virtualisointialusta. Se tarjoaa perusominaisuudet usean virtuaalikoneen pyörittämiseen. Tarpeiden ja vaatimusten kasvaessa on kannattavaa ottaa käyttöön VMware vSpheren kaltainen kokoelma virtualisointiin tarkoitettuja tuotteita. Tuotteiden avulla voidaan muun muassa parantaa resurssien hallintaa ja monitorointia sekä kirjoittaa lokitiedostoja. Kun virtualisoidun ympäristön ja sen päällä pyörivien pilvipalveluiden hallintaa halutaan tehostaa entisestään, on suositeltavaa ottaa käyttöön pilvipalveluiden hallintaa edistävä hallinta-alusta. Sen ominaisuuksiin kuuluu mm. orkestrointi- ja automatisointityökaluja.

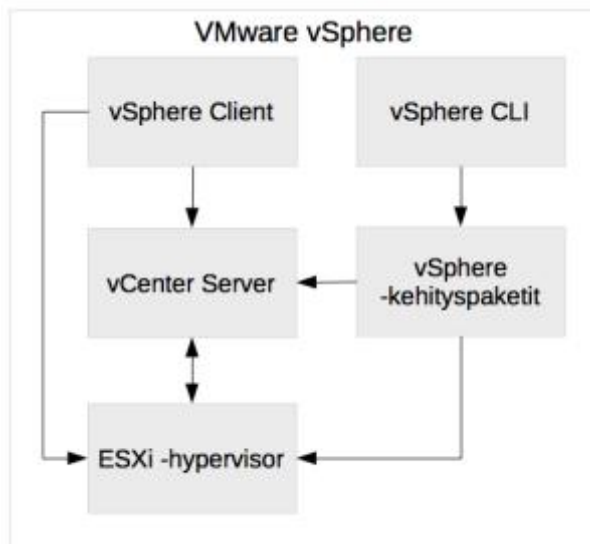
### 2.3.1 Virtualisointialusta

Virtualisointialusta (hypervisor) on ohjelmisto, jonka avulla voidaan luoda ja ajaa virtuaalikoneita. Virtualisointialustat tarjoavat yleensä myös muita ominaisuuksia pelkän virtualisoinnin lisäksi. Kuten edellisessä luvussa mainittiin, virtualisoinnin avulla voidaan muun muassa lyhentää häiriöaikoja sekä parantaa palveluiden saatavuutta ja responsiivisuutta.

Migraatio-ominaisuuden avulla virtuaalikoneita voidaan siirtää yhdeltä fyysiseltä palvelimelta toiselle, normaalin migraation aikana virtuaalikoneet eivät ole käytettävissä. Tosin jotkin virtualisointialustat sisältävät "live migration" -ominaisuuden, jonka avulla migraation aikana palveluita voidaan käyttää. "live migration" -ominaisuuden ansiosta loppukäyttäjälle ei koidu minkäänlaista näkyvää haittaa. Tunnettuja virtualisointiin tarkoitettuja tuotteita ovat muun muassa OpenStack, oVirt, CloudStack ja VMware vSphere. Tässä opinnäytetyössä perehdytään tarkemmin VMware vSphere -virtualisointialustaan. (Martin 2013.)

### 2.3.2 VMware vSphere

VMware vSphere on kokoelma virtualisointiin liittyviä sovelluksia. Sen avulla voidaan toteuttaa ja hallita suurta virtualisoitua IT-infrastruktuuria. Sovelluksien avulla voidaan hallita muun muassa virtuaalikoneita ja niiden resursseja, lokin kirjoitusta ja monitorointia.



Kuvio 3. VMware vSphere -arkkitehtuuri (VMware 2016.)

VMware vSphere -arkkitehtuuriin kuuluu graafinen vSphere Client -asiakasohjelma, jonka avulla päästään käsiksi vSphere Server -palvelimeen tai suoraan ESXi-hypervisorin eli

virtualisointialustaan. vSphere Client -asiakasohjelma on ladattava työpöytäsovellus. Siitä on saatavilla myös web-pohjainen vSphere Web Client -sovellus. Asiakasohjelmien lisäksi vSphere Server -palvelinta ja ESXi-hypervisorilla voidaan käyttää vSphere CLI - komentorivityökalujen ja -kehityspakettien avulla. Niitä voidaan käyttää myös automatisoitujen toiminnallisuuksien kehittämiseen ja kolmannen osapuolen sovellusten integroimiseen. (VMware 2016.)

VMware vSphere -virtualisointialustaa voidaan käyttää ilman vCenter Server -palvelinta. Asiakasohjelmien, komentorivityökalujen ja virtualisointialustan avulla voidaan toteuttaa yksinkertaisimpia virtualisointiin liittyviä toiminnallisuuksia. On kuitenkin suositeltavaa ottaa käyttöön myös vCenter Server -palvelin, kun halutaan lisää ominaisuuksia ja kun virtualisointialustojen määrä kasvaa, koska palvelimen avulla voidaan hallita useaa virtualisointialustaa samanaikaisesti. Ominaisuuksia ovat muun muassa resurssien jakaminen virtualisointialustojen välillä, lokitiedostojen luominen ja monitorointityökalut. (VMware 2016.)

ESXi on VMwaren käyttämä virtualisointialusta. ESXi asennetaan suoraan tietokonelaitteiston päälle, se ei tarvitse toimiakseen käyttöliittymää. Tämän kaltaisista ohjelmistoista käytetään usein nimeä "bare-metal". (VMware 2016.)

### **2.3.3 Pilvipalveluiden hallinta**

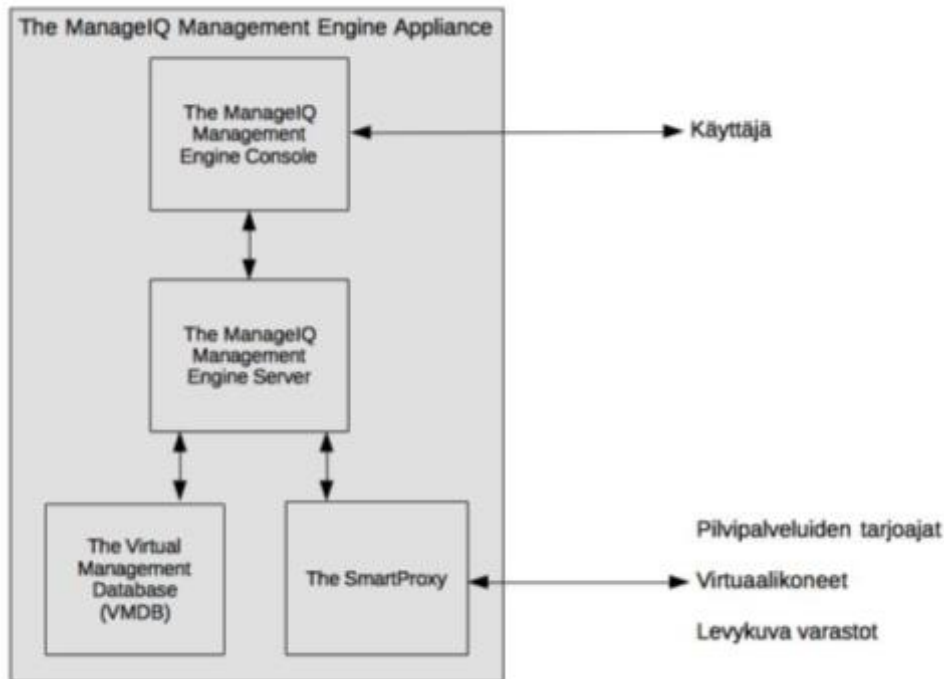
Virtualisointialustan päällä pyörivien pilvipalveluiden hallinta muuttuu vaikeammaksi, kun virtuaalikoneiden määrä kasvaa tai kun halutaan käyttää useampaa pilvipalveluidentarjoajaa samanaikaisesti. Pilvipalveluiden hallintaan tarkoitettujen sovellusalustat ovat ohjelmistoja, joiden tarkoitus on tehostaa virtualisointiin tarkoitettujen tuotteiden toimintaa entisestään. Sovellusalustat asennetaan virtualisointituotteiden oheen tukemaan niiden toimintaa. (Gartner. 2016.)

Pilvipalveluiden hallintaan tarkoitettujen sovellusalustan ominaisuuksia ovat muun muassa

- automatisointi- ja orkestrointityökalut
- integraatiomahdollisuudet muihin järjestelmiin
- analytiikan ja laskutuksen seuranta
- graafinen käyttöliittymä.

ManagelQ on pilvipalveluiden hallintaan tarkoitettu sovellusalusta, joka on avoimen lähdekoodin sovellus. ManagelQ tukee muun muassa seuraavia virtualisointiin tarkoitettuja tuotteita: VMware vSphere, Openstack, oVirt ja Cloudstack. Näiden lisäksi se

tukee myös ulkoisia pilvipalveluiden tarjoajia kuten esimerkiksi Amazon Web Services. (ManageIQ 2016.)



Kuvio 4. Esimerkki ManageIQ-arkkitehtuurista. (Red Hat 2015.)

The ManageIQ Management Engine Appliance on esiasennettu virtuaalikone. Se on ManageIQ-sovellusalueen keskeisin komponentti. Virtuaalikoneelle on asennettu The ManageIQ Engine Server -palvelin, jonka toiminta perustuu sille määriteltyihin rooleihin. Roolien avulla palvelimia voidaan jakaa erilaisiin tehtäviin. Oletuksena niille on asetettu joukko rooleja, jotka mahdollistavat yleisimmät toiminnot yhdellä palvelimella. (Red Hat 2015.)

Roolien avulla voidaan esimerkiksi estää pääsy jonkin ManageIQ-palvelimen käyttöliittymään, jos palvelin halutaan nimenomaan keskittää johonkin muuhun tarkoitukseen. Muita tarkoituksia voivat olla esimerkiksi monitorointitehtävät. Monitorointiroolein varustellun palvelimen tarkoitus on seurata virtualisointialustojen tapahtumia ja prosessoida niitä. Muiden roolien avulla voidaan esimerkiksi sallia tai estää palvelimelta virtuaalikoneiden käynnistäminen, sammuttaminen ja kloonaminen. (Red Hat 2015.)

## 2.4 Sovelluksien ja palveluiden hallinta

Virtualisointiin ja pilvipalveluiden hallintaan tarkoitettujen sovellusten toiminnallisuudet ja ominaisuudet ulottuvat virtuaalikoneiden luomiseen ja ajamiseen. Virtuaalikoneilla ajettavien sovelluksien ja sovelluksien tarjoamien palveluiden hallintaan tarvitaan erilliset

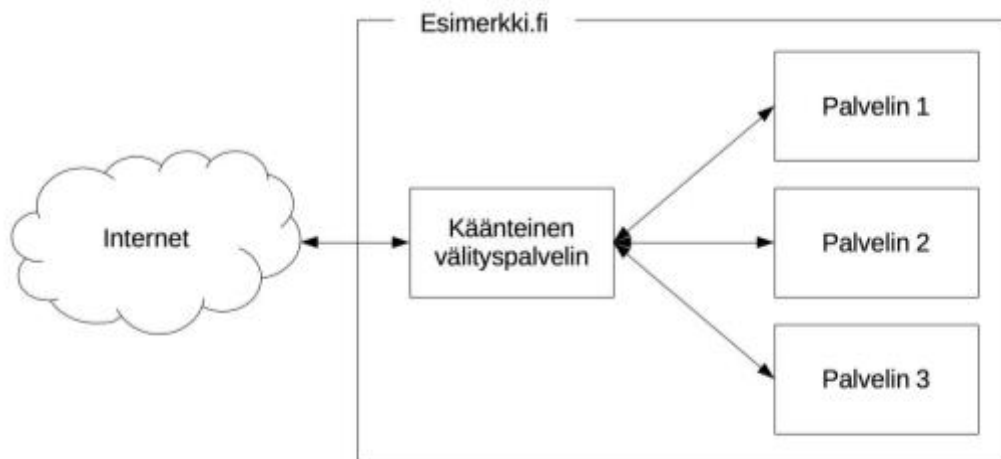
työvälineet. Seuraavissa luvuissa käydään läpi minkälaisilla työkaluilla virtuaalikoneilla pyöriviä sovelluksia ja palveluita voidaan hallinnoida.

### 2.4.1 Konfiguraatioiden hallinta

Konfiguraatioiden hallinnalla tarkoitetaan toimintoja, joiden avulla sovelluksien versioita ja versiopäivityksiä hallitaan. Ansible on yksi konfiguraatioiden hallintaan tarkoitettu sovellus. Sen avulla voidaan asentaa ja hallinnoida palvelimille asennettuja ohjelmia. Ansiblen tarkoituksena on tehdä järjestelmien automatisoinnista, orkestroinnista ja konfiguraatioiden hallinnasta yksinkertaisempaa ja kevyempää. Ansiblen toimintaperiaate perustuu Ansible Modules -nimisiin pienoishjelmiin. Pienoishjelmat suoritetaan kohdekoneilla SSH-protokollan avulla. Suorituksen jälkeen pienoishjelmat poistetaan. Ansible ei vaadi toimiakseen agenteja; sen takia toteutukset ovat kevyitä, huomaamattomia ja vaativat vähemmän ylläpitoa. Ansible käyttää yksinkertaista YAML-kieltä, jota on helppo lukea, kirjoittaa ja ylläpitää. (Ansible. 2016.)

### 2.4.2 Palveluiden tarjoaminen

Käyttäjän ja palvelun välillä oleva yhteys kulkee yleensä välityspalvelimen kautta, jonka tarkoitus on muun muassa jakaa kuormaa palvelimien välillä. Tässä projektissa palveluita tarjotaan käyttäjälle käänteisen välityspalvelimen avulla. Nginx on web-palvelin joka voi toimia käänteisenä välityspalvelimenä. Katso kuviota 5.

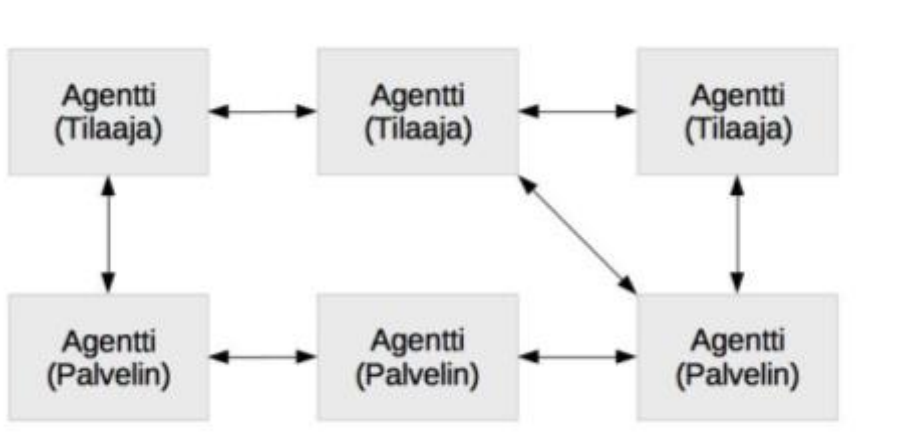


Kuvio 5. Käänteisen välityspalvelimen toimintaperiaate.

Käänteisen välityspalvelimen tarkoitus on toimia käyttäjän ja kohdepalvelimen välillä siten, että käyttäjä ei ole tietoinen välityspalvelimesta. Katso esimerkkitalanne kuviossa 5. Käyttäjä ottaa yhteyden internetin välityksellä esimerkki.fi-palveluun, käyttäjä ei

kuitenkaan tiedä, että liikenne ohjautuu välityspalvelimelle ja sitä kautta palvelimelle, joka tarjoaa palvelua. Käänteisen välityspalvelimen avulla voidaan käyttäjän huomaamatta parantaa palveluiden toimintaa. Yleisiä välityspalvelimen ominaisuuksia ovat muun muassa verkkoliikenteen seuranta ja kuorman jakaminen. Verkkoliikenteen seurannan avulla voidaan saada kiinni haitallisia pyyntöjä, joita on yritetty tehdä kohdepalvelimille. Kuorman jaon avulla liikennettä voidaan jakaa usealla eri kohde palvelimelle. (Nginx 2016.)

Käänteisen välityspalvelimen konfiguraation pitää määrittellä kohdepalvelimet, joille liikennettä ohjataan. Välityspalvelin ei kuitenkaan tiedä, milloin esimerkiksi uusi kohdepalvelin on pystyssä ja valmiina tarjoamaan palvelua, vaan konfiguraatio on tehtävä käsin. Tämä vaihe voidaan automatisoida palveluiden etsimiseen ja niiden tilan seuraamiseen tarkoitettujen työkalujen avulla. Yksi tämänlainen palvelu on nimeltään Consul, sen toiminta perustuu kohdepalvelimille asennettavista agenteista. Katso kuviota 6.



Kuvio 6. Consul-palvelun toimintaperiaate. (HashiCorp 2016.)

Consul-palvelun toiminta perustuu agenteihin, jotka ovat prosesseja. Agentin tila voi olla palvelin tai tilaaja, katso kuviota 6. Kun joukko koneita halutaan Consuliin piiriin, jokaiselle koneelle asennetaan agentti ja sen tilaksi valitaan tilaaja. Lisäksi tarvitaan yksi kone, jolla on agentti ja sen tilaksi on asetettu palvelin. Consulien toimintaperiaate perustuu agenttien keskusteluun keskenään. Palvelin-agentilla on enemmän ominaisuuksia kuin tilaaja-agentilla. Sen tarkoituksena on muun muassa keskustella toisten palvelin-agenttien kanssa, jotka sijaitsevat toisissa palvelinkeskuksessa. Consulien avulla saadaan selville verkkoon ilmestyneet uudet koneet ja niiden IP-osoitteet. Consulien tietojen ja välityspalvelimen avulla voidaan automatisoida uusien palvelimien lisääminen kuormanjakajan piiriin. (HashiCorp 2016.)

## 2.5 IT-infrastruktuurin automatisointi

Aiemmat luvut liittyvät opinnäytetyön toiminnalliseen osuuteen, tämä luku käsittelee jatkokehitysehdotukseen liittyvää ratkaisua. Virtualisointialustat, pilvipalveluiden hallintaan tarkoitetut sovellusalustat ja konfiguraatioiden hallintaratkaisut tarjoavat monenlaisia toimintoja virtuaalisen ympäristön, sovelluksien ja palveluiden hallintaan.

Toiminnallisuudet ovat suoritettavia työtehtäviä eli toimintoja, jotka noudattavat jotain tiettyä kaavaa ja joita suoritetaan usein. Prosessit sen sijaan koostuvat työtehtävistä, jotka toteutetaan tietyssä järjestyksessä ja ne voivat sisältää ehtoja. Kaikkien työtehtävien ja prosessien suorittamiseen ei välttämättä tarvita henkilöä suorittamaan niitä, vaan ne voidaan automatisoida.

Automatisoinnilla tarkoitetaan työtehtävien ohjelmointia automaattisiksi toiminnoiksi. Automatisoituja työtehtäviä voivat olla esimerkiksi web-palvelimen käynnistäminen tai web-palvelimen sammuttaminen. Orkestroinnilla sen sijaan tarkoitetaan prosessien ohjelmointia automatisoiduiksi työjonoiksi. Orkestroitu prosessi voi pitää sisällään esimerkiksi web-palvelimen asentamisen ja käynnistämisen sekä ilmoituksen lähettämisen siitä, että web-palvelin on toiminnassa. Orkestrointi on käytännössä automatisoitujen työtehtävien hallintaa.

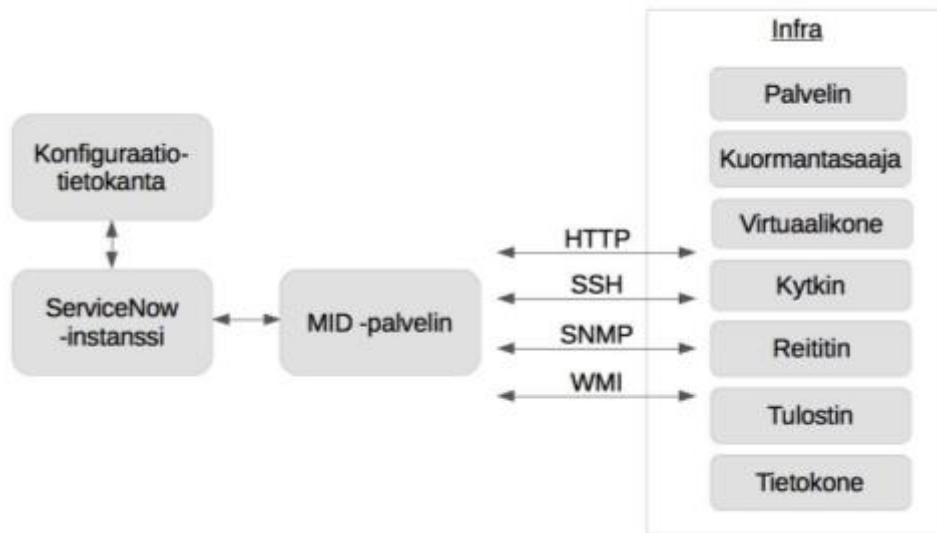
### 2.5.1 IT-operaatioiden hallinta

IT-operaatioiden hallinnalla tarkoitetaan päivittäisiä prosesseja, jotka liittyvät IT-infrastruktuurin hallintaan, ylläpitoon ja kehitykseen. IT-infrastruktuuri voi sisältää muun muassa sovelluksia, palveluita, tietovarastoja ja verkkoja. Lisäksi hallintaan liittyy yhteydet eri infrastruktuurin osien välillä. IT-operaatioiden hallinnasta käytetään termiä ITOM (IT Operation Management). IT-operaatioiden hallintaan tarkoitettujen ohjelmistojen tarkoituksena on muun muassa parantaa näkyvyyttä palveluihin ja niiden tilaan, nopeuttaa virhetilanteiden havainnointia ja käsittelyä sekä edistää orkestrointia. (Gartner 2016.)

### 2.5.2 ServiceNow

ServiceNow on saman nimisen yrityksen luoma sovellusalusta, jonka tarkoituksena on tukea IT-operaatioiden, IT-palveluiden ja liiketoiminnan hallintaa sekä sovelluskehitystä. Sovellusalustan eri ominaisuudet on tuotteistettu, ja jokainen tuote sisältää joukon sovelluksia. Tuotteita voidaan käyttää itsenäisesti tai useita tuotteita yhtä aikaa, jolloin ne voivat tukea toisten tuotteiden sovellusten toimintaa. ServiceNow voidaan määrittellä SaaS- tai PaaS-palveluksi koska sovellusalustalle voidaan kehittää omia sovelluksia (Vizard 2012).





Kuvio 7. ServiceNow-sovelluslujan toimintaperiaate.

Sovelluslujan toiminta perustuu konfiguraatitietokannassa olevan datan käsittelyyn ServiceNow-instanssin avulla, katso kuviota 7. Konfiguraatitietokannassa on kaikki oleellinen tieto organisaation IT-infrastruktuurista. Tietokantaan on tarkoitus tallentaa kaikki tieto niin fyysisistä komponenteista, kuten tietokoneista, palvelimista ja reitittimistä, kuin myös niihin liittyvistä yhteyksistä sekä myös palveluista, joita ne tarjoavat. ServiceNow-instanssi on sovelluslujan keskeisin komponentti. Se sisältää käyttöliittymän ja sovellukset. Konfiguraatitietokannassa olevaa dataa käsitellään sovellusten avulla. Jokainen sovellus tulkitsee ja kuvaa tietokannan dataa eri tavoin, sovelluksia voidaan käytännössä pitää erilaisina näkökulmina tietokantaan. Niiden avulla dataa luetaan, lisätään, päivitetään ja poistetaan. MID-palvelimen tarkoituksena on erilaisten yhteyksien avulla tutkia IT-infrastruktuuria ja lisätä dataa ServiceNow-instanssin sovellusten avulla konfiguraatitietokantaan. Katso kuviota 7.

ServiceNow-instanssi ja konfiguraatitietokanta pyörivät ServiceNow-palveluntarjoajan palvelimilla, niitä voidaan halutessaan pyörittää myös omalla palvelimella. MID-palvelin on sen sijaan tarkoitus sijoittaa omalle palvelimille palomuurin sisäpuolelle ja sillä on oltava oikeudet ja näkyvyys verkkoihin ja sen laitteisiin.



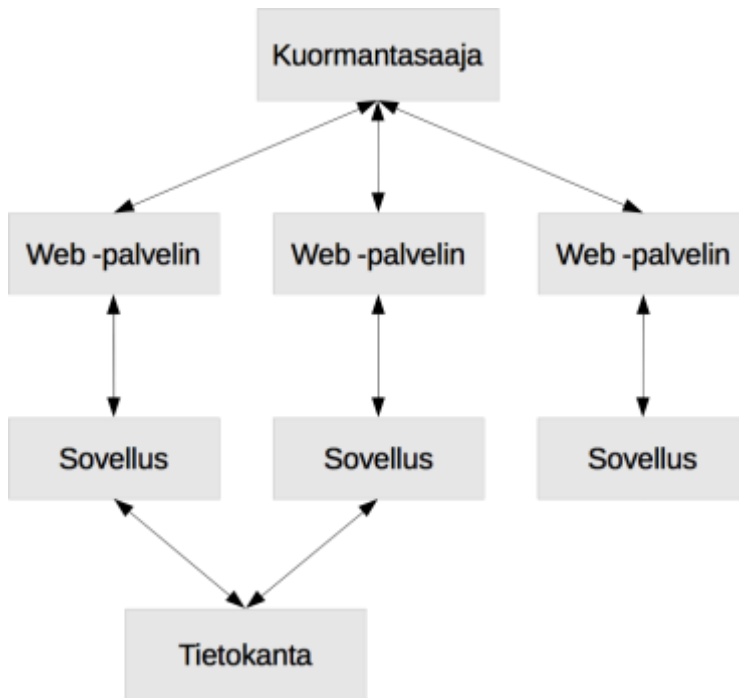
Kuvio 8. ServiceNow-sovellusalustan tuotteet

ServiceNow-sovellusalustan tuotteista mielenkiintoisin automatisoinnin kehitystyön kannalta on IT Operations Management. Se sisältää hyödyllisiä orkestrointiin, palveluiden näkyvyyteen ja tapahtumien hallintaan liittyviä sovelluksia.

ServiceNow (2016) mukaan IT Operations Management -tuotteeseen kuuluvat sovellukset ovat:

- Service Discovery
- Service Mapping
- Orchestration
- Configuration Automation
- Cloud Provisioning
- Event Management.

Service Discovery- tai lyhyemmin Discovery-sovelluksen avulla voidaan etsiä verkosta laitteita. Laitteet voivat olla tietokoneita, reitittimiä, tulostimia tai mitä vain muita laitteita, joita voidaan kytkeä verkkoon. Kun Discovery-sovellus löytää laitteen, se tutkii sen konfiguraation sekä tilan ja päivittää tiedot konfiguraatietietokantaan. Fyysisten laitteiden lisäksi sovellus osaa tunnistaa tiettyjä ohjelmistoja ja TCP-yhteyksiä. Tunnistettujen laitteiden, laitteiden ohjelmistojen ja TCP-yhteyksien avulla Discovery-sovellus osaa tunnistaa eri laitteiden välillä olevia yhteyksiä. Myös yhteydet tallennetaan konfiguraatietietokantaan. (ServiceNow 2016.)



Kuvio 9. Yksinkertaistettu esimerkki kuviosta, joka voidaan luoda Service Mapping -sovelluksen avulla.

Service Mapping -sovellus hyödyntää Discovery-sovelluksen keräämiä tietoja laitteista, ohjelmistoista ja niiden yhteyksistä. Sen avulla voidaan luoda kuvioita, joiden avulla on helpompi hahmottaa IT-infrastruktuuriin kuuluvien laitteiden ja ohjelmistojen yhteyksiä toisiinsa, katso kuviota 9. IT-infrastruktuurin parempi hahmotus parantaa käsitystä siitä, mitkä laitteet ja palvelut ovat riippuvaisia toisistaan. Lisäksi virhetilanteissa voidaan paikallistaa virheellinen laite, ohjelmisto tai yhteys sekä nähdä mihin muihin laitteisiin, ohjelmistoihin tai yhteyksiin virhetilanne vaikuttaa. (ServiceNow 2016.)

Event Management -sovellus tehostaa tapahtumien seuranta ja käsittelyä. Sovelluksen avulla voidaan luoda kuvioita ja kaavioita erilaisista tapahtumista. Tapahtumia voivat olla esimerkiksi virhetilanteet tai virtuaalikoneiden provisioinnit. Seurannan lisäksi tapahtumia voidaan käsitellä ja niihin voidaan lisätä tarvittaessa tietoja ja ohjeita, kuinka esimerkiksi virhetilanne voidaan korjata. Tapahtumiin voidaan lisätä myös toimintoja, esimerkiksi kriittiseen virhetilanneilmoitukseen voidaan automatisoida toiminto, joka lähettää esimerkiksi sähköpostin ja luo tiketin tapahtuneesta virhetilanteesta.

Orchestration-sovellus tarjoaa välineet monimutkaisten automatisoitujen työtehtävien orkestrointiin. Sovellus lisää uusia ominaisuuksia ServiceNow'n työjonoihin.

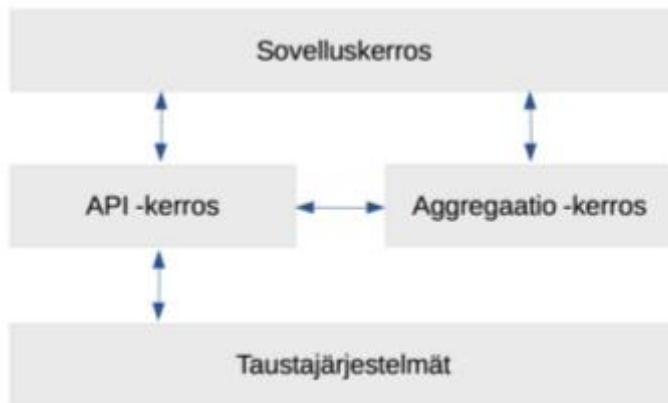
Orkestrointitoiminnot voidaan jakaa erilaisiin ryhmiin: perustoimintoihin, SSH-toimintoihin ja integraatiotoimintoihin, joiden avulla ServiceNow voidaan integroida kolmannen osapuolen sovelluksiin. (ServiceNow 2016.)

Perustoimintojen avulla voidaan tehdä yksinkertaisia työjonoihin ja prosesseihin liittyviä orkestrointitehtäviä, niitä ovat muun muassa DNS-selvitys, keruuyksikön ajaminen, SNMP-kysely ja SOAP-pyyntö. SSH-toimintojen avulla voidaan tehdä muun muassa seuraavia operaatioita Linux- ja Unix-pohjaisissa järjestelmissä 1) tiedostojen kopiointi, lukeminen ja muokkaus, 2) tiedostojen vertailu keskenään ja 3) shell-komentojen ajaminen.

(ServiceNow 2016.)

### 3 Asiakkaan virtualisoitu palvelinympäristö

Tässä kappaleessa käydään läpi yleisellä tasolla, minkälainen on asiakkaan tietojärjestelmäarkkitehtuuri ja mihin osaan arkkitehtuuria tämän opinnäytetyön toiminnallinen osuus liittyy.



Kuvio 10. Asiakkaan tietojärjestelmäarkkitehtuuri yleisellä tasolla.

Kuviossa 10 on kuvattu yleisellä tasolla asiakkaan ympäristön tietojärjestelmäarkkitehtuuri. Taustajärjestelmäkerros sisältää kaikki asiakkaan ja kolmannen osapuolen tietojärjestelmät, joita käytetään palveluiden tuottamiseen. Sovelluskerroksessa on sovellukset, jotka tuottavat näitä palveluita loppukäyttäjille. Sovelluskerroksen sovellukset eivät kommunikoi taustajärjestelmien kanssa suoraan vaan API- ja aggregaatiokerrosten välityksellä. Tietojärjestelmäarkkitehtuuri noudattaa palvelukeskeisen arkkitehtuurin periaatteita. Sovellus- ja taustajärjestelmäkerroksien välissä on sen takia rajapinta- ja aggregaatiokerros. Rajapinta kerroksen erilaisten rajapintojen tarkoituksena on tarjota jotain palvelua, joka on koostettu taustajärjestelmien palveluista. Rajapintakerroksen lisäksi on aggregaatiokerros, joka sisältää pienoisohjelmia/rajapintoja, jotka hyödyntävät vastaavasti rajapintakerroksen rajapintojen tarjoamia palveluita ja koostavat niistä palveluita, joita sovellukset voivat hyödyntää.

Tämä opinnäytetyö liittyy lähinnä sovellus-, rajapinta- ja aggregaatiokerroksien sovelluksiin ja rajapintoihin, jotka pyörivät virtuaalikoneilla. Sovelluksille ja rajapinnoille on omat tuotanto- ja testiympäristöt. Tuotanto- ja testiympäristöissä olevat sovellukset ja rajapinnat ovat omilla verkkoalueillaan. Yhteensä virtuaalikoneita on koko ympäristössä useita satoja. Opinnäytetyön tarkoituksena on mahdollistaa virtuaalikoneiden provisiointi/de-provisiointi itsepalveluna. Lisäksi on tarkoitus automatisoinnin avulla tehostaa prosesseja, jotka koostuvat koneiden provisioinnista, sovelluksen asennuksesta ja palvelun tarjoamisesta.

## 4 Automatisoinnin kehitystyö

Tässä luvussa esitellään, kuinka projektin toiminnallinen osuus toteutettiin eli kuinka palveluiden kehitysprosessin kolme eri vaihetta automatisoitiin. Oman konesalin automatisoinnin kehitystyön perustana on mahdollistaa virtuaalikoneiden automatisoitu provisiointi. Virtuaalikoneen provisiointi sisältää monta eri vaihetta. Virtuaalikoneen provisiointi käsin, ilman automatisointia, on aikaa vievä prosessi. Kun provisiointeja halutaan tehdä viikoittain kymmenille tai sadoille koneille, se ei ole enää kannattavaa. Virtuaalikoneiden automatisoidussa provisiointiprosessissa tulee ottaa huomioon DHCP, nimipalvelu ja virtuaalikoneiden oikeudet.

Automatisointiprosessin myötä provisioidut koneet saavat IP-osoitteensa DHCP:lla. Lisäksi uudelleen käynnistettyjen koneiden IP-osoitteet voivat vaihtua. Vaihtuvat IP-osoitteet tulee siis ottaa huomioon. Nimipalvelun avulla koneille saadaan aina toimivat osoitteet. IP-osoitteet on jaettu neljään verkkoalueeseen: API ja sovellus verkkoihin, sekä niiden testi ja tuotanto variaatioihin. Nimipalvelu määrittää verkkoaluetta kuvaavan nimen IP-osoitteelle. Nykyisessä ympäristössä osa IP-osoitteista voi olla kiinteästi määriteltä.

Virtuaalikoneiden oikeudet tulee ottaa huomioon provisiointivaiheessa. Tämän projektin osalta virtuaalikoneet on jaettu API- ja sovellus-ryhmään. Kumpaankin ryhmään kuuluvilla virtuaalikoneilla on omat käyttäjänsä niiden hallintaan. Ilman oikeita käyttäjäasetuksia virtuaalikoneille ei päästä käsiksi.

Asiakkaan IT-palveluiden automatisoinnin kehitystyö on pitkä prosessi. Yksi ensimmäisistä tavoitteista on aloittaa virtuaalikoneiden viikoittainen uudelleenprovisiointi. Ennen sitä tarvitaan kuitenkin menetelmät ja käytännöt, miten se voidaan käytännössä toteuttaa. Oman konesalin automatisoinnin kehitystyö on toistaiseksi jäänyt tämän osalta kesken. Tämän projektin tarkoituksena on selvittää, mitä on jo tehty, ja laatia toimintasuunnitelma siitä, miten projektia jatketaan eteenpäin. Toimintasuunnitelman tavoitteeksi rajataan se, että saadaan dokumentoidut menetelmät ja käytännöt sille, kuinka automatisoitu provisiointi voidaan toteuttaa omassa konesalissa. Toimintasuunnitelman eri vaiheista raportoidaan ja lopuksi tehdään yhteenveto, kuinka projekti onnistui.

### 4.1 Nykyisen tilanteen kartoitus

Ensimmäinen vaihe oli selvittää, miksi automatisoinnin kehitystyö oli jäänyt kesken, mihin asti siinä oli edetty ja voidaanko sitä jatkaa. Selvitys aloitettiin keräämällä aiheesta kaikki mahdollinen dokumentaatio. Seuraavaksi haastateltiin automatisoinnin

kehitystyöprojektissä aiemmin mukana olleita henkilöitä. Haastatteluiden ja dokumentaatioiden avulla saatiin parempi kuva projektista ja sen tavoitteista. Ilmeni että automatisoinnin kehitystyö oli jäänyt kesken, koska yksi projektin jäsenistä oli lähtenyt projektista pois. Projektin jäsenen vastuulla oli ollut vaihe, joka oli tärkeä osa projektia. Ilman tätä vaihetta automatisoinnin kehitystyöprojektia ei voitu viedä eteenpäin. Projektin puuttuvan jäsenen paikalle ei tuolloin löydetty korvaajaa.

Kesken jääneen projektin tilanne tehtyjen vaiheiden osalta oli seuraava. VMware vSphere -virtualisointialustan päälle oli asennettu pilvipalveluiden hallintaan tarkoitettu sovellusalusta, ManageIQ. Sen graafisen käyttöliittymän avulla voitiin provisoida virtuaalikoneita. Virtuaalikoneille voitiin määritellä muun muassa nimi, muistin ja prosessorien määrät sekä se, mihin verkkoon ne halutaan provisoida. Koneiden provisiointi tapahtui ManageIQ-käyttöliittymän tai REST-pohjaisen web-rajapinnan kautta. Projektin kesken jäänyt vaihe liittyi ohjelmakoodiin, joka käytti ManageIQ-sovellusalustan REST-rajapintaa. Ohjelmakoodin avulla voitiin luoda koneita komentoriviltä. Lisäksi se oli yksi osa sovellusten ja palveluiden automatisointiprosessia.

Kun automatisoinnin kehitystyö -projektin kesken jäänyt tilanne oli selvä, aloitettiin toimintasuunnitelman tekeminen. Ensimmäinen vaihe oli ohjelmakoodin testaus ja kehitys. Toimivan ohjelmakoodin avulla kehittäjille saatiin palvelu, jonka avulla voidaan provisoida virtuaalikoneita itsepalveluperiaatteella. Tällä hetkellä virtuaalikoneiden provisiointi pitää erikseen tilata sähköpostitse ja kehittäjän täytyy odottaa, kunnes tilaus on valmis. Lisäksi provisiointeja tekee siihen erikoistunut henkilö.

#### **4.2 Virtuaalikoneiden provisiointi itsepalveluna**

Koska automatisoinnin kehitystyöprojekti oli jäänyt kesken, ohjelmakoodin toimivuudesta nykyisessä arkkitehtuurissa ei ollut takeita. Arkkitehtuuri on voinut muuttua olennaisesti. Ohjelmakoodia testattiin ja siinä havaittiin puutteita. Ohjelmakoodista puuttui olennaisia muuttujia ja virtuaalikoneet provisioituivat väärin verkkoihin. Korjauksen jälkeen ohjelmakoodi otettiin käyttöön ”virtuaalikoneiden provisiointi itsepalveluna” -toiminnallisuutena. Ohjelmakoodin toiminta dokumentoitiin. Sen tarkoituksena on toimia ohjeistuksena, kuinka virtuaalikoneita voidaan provisoida itsepalveluna sen sijaan että virtuaalikoneita tarvitsee pyytää erikseen s-postitse.

Ohjelmakoodin ominaisuuksia ovat

- koneiden provisiointi/de-provisiointi
- provisioitujen koneiden listaus
- käynnissä olevien provisioitujen koneiden listaus
- provisioitujen koneiden tietojen ja tilan näyttäminen.

Yllämainittujen ominaisuuksien lisäksi ohjelmakoodia varten on jatkokehitysehdotuksia. Kun ohjelmakoodi aloittaa uuden virtuaalikoneen provisioinnin, se näyttää karkeasti provisiointiin kuluvan ajan. Aika ei aina pidä paikkaansa vaan se on suuntaa antava. Tarkoituksena on saada selville tarkempi aika, joka näytetään käyttäjälle. Lisäksi, kun ohjelmakoodi ilmoittaa koneen olevan valmis, kone ei ole vielä saanut IP-osoitetta DHCP:lta. Ohjelmakoodia voitaisiin muuttaa siten, että koneen ilmoitetaan olevan valmis vasta, kun IP-osoite on saatavilla. Ohjelmakoodiin halutaan lisätä myös ominaisuus luoda useita virtuaalikoneita samanaikaisesti samoilla konfiguraatioilla. Konfiguraatioiden validointia voitaisiin myös kehittää pidemmälle. Näin vältetään virheellisten konfiguraatioasetusten aiheuttamilta ongelmilta. Esimerkiksi erikoismerkkien käyttöä voitaisiin rajoittaa virtuaalikoneen nimessä tai käyttäjälle voitaisiin näyttää viesti, jos muistin määräksi on määritelty kohtuuttoman suuri arvo.

### **4.3 Automatisoitu kuormantasaus**

Asiakkaan tietojärjestelmässä palveluita tarjotaan käyttäjille kuormantasaajan välityksellä. Kuormantasaajana on käytössä Nginx, ja se on konfiguroitu toimimaan käänteisenä välityspalvelimenä. Kun uusi virtuaalikone on provisioitu ja sille on asennettu sovellus, liikenteen ohjaus täytyy käsin lisätä Nginx-välityspalvelimen konfiguraatioon. Automatisoitu liikenteen ohjaus kuormantasaajalle vähentää käsin tehtävää konfiguraatiota ja on osa automatisoinnin kehitystyötä.

Automatisoitu kuormantasaus edellyttää, että virtuaalikoneilla pyörivistä palveluista saadaan seuraavat tiedot:

- tieto kun uusi sovellus on asennettu ja valmiina tarjoamaan palvelua
- virtuaalikoneen IP-osoite
- virtuaalikoneen verkko
- sovelluksen tarjoaman palvelun tila
- ympäristökohtaisia muuttujia kuten esimerkiksi onko palvelu testi vai tuotantoympäristössä.



Edellä mainitut tiedot voidaan selvittää palveluiden etsimiseen ja palveluiden tilan seurantaan tarkoitetun työkalun avulla. Tämän kaltaista työkalua ei ennestään ympäristössä käytetty. Tähän tarkoitukseen valittiin työkaluksi Consul. Consulien avulla pystytään seuraamaan, kun uusia palveluita ilmenee, ja ilmoittamaan tieto eteenpäin esimerkiksi Nginx-välityspalvelimelle. Consulien käyttöönotto edellytti monia toimenpiteitä: Sovelluksien asennusprosessia muutettiin siten että virtuaalikoneelle asennettiin sovelluksen lisäksi myös Consul-agentti. Consul-agentin tilaksi asetettiin ”tilaaja”. Lisäksi provisioitiin virtuaalikone, josta tehtiin Consul-palvelin. Palvelimen Consul-agentin tilaksi asetettiin ”palvelin”. Consulien toiminta perustuu agenttien väliseen kommunikointiin keskenään. Tässä yhteydessä, kun uusi virtuaalikone oli provisioitu ja sovellus sekä Consul asennettu, agentti alkoi kommunikoida muiden samassa verkossa olevien agenttien kanssa. Consul-palvelimella pyörivä palvelin-agentti kerää edellä mainitut tiedot tilaaja-agenttien avulla ja ilmoittaa tiedon Nginx-välityspalvelimelle.

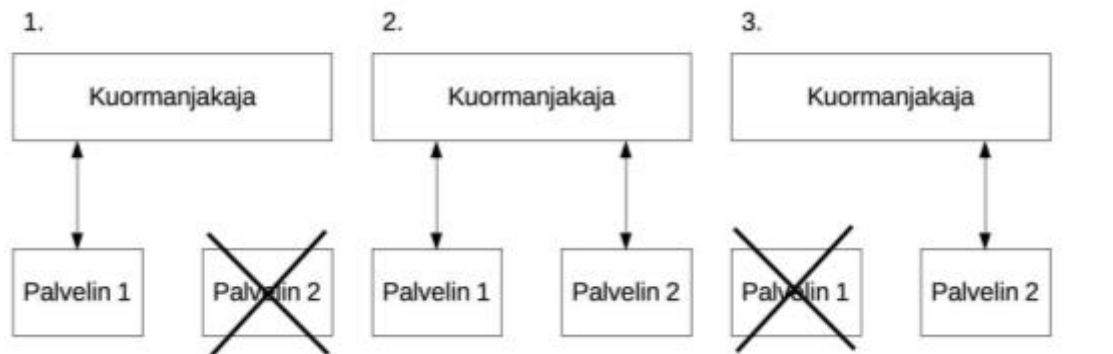
Seuraava vaihe oli luoda sellainen Consul-arkkitehtuuriratkaisu, joka soveltuisi asiakkaalle parhaiten. Suunnittelussa tuli ottaa huomioon se, kuinka Consul-palvelimet jaetaan verkkojen välille. Consul-palvelimien suositeltu määrä on vähintään 3 per verkko, jotta Consul-palvelun toiminta olisi vikasietoista. Virtuaalikoneita provisioidaan neljään eri verkkoon. Nähtiin, että verkot on hyvä pitää eristyksissä toisistaan, koska osaa käytetään testi- ja osaa tuotantokäytössä. Consul-palvelimia olisi tarvittu 12 kappaletta, mutta päädyttiin kuitenkin siihen, että luodaan uusi verkko, josta avataan portit neljään edellä mainittuun verkkoon. Uuteen verkkoon on tarkoitus tuoda Consulien kaltaisia palveluita, jotka tarvitsevat pääsyn useisiin eri verkkoihin. Uuteen verkkoon provisioidaan 6 Consul-palvelinta, joista puolet ovat testi- ja puolet tuotantoympäristöä varten.

#### **4.4 Virtuaalikoneiden orkestroitu uudelleenprovisiointi viikoittain**

Tässä luvussa käydään läpi suunnitelma, miten automatisoinninkehitystyötä voidaan jatkaa hyödyntäen ohjelmakoodia ja automatisoitua kuormantasausta. Virtuaalikoneiden provisiointiin tarkoitetun ohjelmakoodin ja orkestroidun kuormantasauksen avulla automatisointia voidaan viedä entistä pidemmälle. Seuraava vaihe on toteuttaa orkestroitu virtuaalikoneiden uudelleenprovisiointi kerran viikossa.

Virtuaalikoneiden uudelleenprovisiointi viikoittain edesauttaa muun muassa sitä, että virtuaalikoneiden tietoturvapäivitykset pysyvät ajan tasalla eikä niitä tarvitse asentaa erikseen jokaiselle virtuaalikoneelle. Lisäksi virtuaalikoneille kerääntyvät turhat tiedostot tuhoutuvat eivätkä ne vie turhaan tilaa.

Virtuaalikoneiden viikoittaiseen uudelleenprovisiointiin halutaan soveltaa rolling release -menetelmää. Tämä tarkoittaa, että virtuaalikoneita voidaan ajaa alas siten, että palvelut ovat koko ajan saatavilla. Menetelmä perustuu siihen, että kohteet de-provisioidaan ja provisioidaan vuorotellen ja seuraava kohde vasta kun edellinen on valmis. Orkestroidun kuormantasauksen ansiosta välityspalvelin saa tiedon, kun palvelu on poistunut käytöstä ja kun se tulee taas käytettäväksi. Menetelmää voidaan havainnollistaa kuvan avulla, katso kuviota 11.



Kuvio 11. rolling release -menetelmä.

Virtuaalikoneiden viikoittainen uudelleenprovisiointi edellyttää muutoksia muun muassa ohjelmakoodiin. Ohjelmakoodiin täytyy lisätä konfiguraatio-ominaisuus, jonka avulla voidaan määrittellä, mikä sovellus virtuaalikoneelle asennetaan, kun se on provisioitu. Lisäksi tarvitaan orkestrointiväline, jonka avulla jokainen automatisoitu vaihe saadaan koottua yhdeksi kokonaiseksi prosessiksi. Eli konfiguraatioiden määrittely ja sen pohjalta virtuaalikoneen provisiointi ja sovelluksen asennus virtuaalikoneelle sekä prosessin ajastus ohjelmoidaan toistumaan kerran viikossa. Ehdotukseni orkestrointivälineeksi on ServiceNow-sovellusalusta. Seuraavassa luvussa käydään läpi projektin tuloksia ja sitten esitellään jatkokehitysehdotuksen muodossa, miten ServiceNow-sovellusalusta soveltuisi orkestrointivälineeksi.

## 5 Tulokset

”Virtuaalikoneiden provisiointi itsepalveluna” -toiminnallisuuden ja automatisoidun kuormantasauksen avulla automatisoinnin kehitystyötä on onnistuttu viemään eteenpäin. Automatisoitujen vaiheiden avulla on säästetty ja tullaan säästämään aikaa sekä resursseja. Lisäksi automatisoidut vaiheet mahdollistavat, että automatisoinnin kehitystyötä voidaan viedä entistä pidemmälle. Esimerkiksi edellisessä luvussa on kuvattu suunnitelma, kuinka virtuaalikoneiden uudelleenprovisiointi kerran viikossa voidaan toteuttaa. Seuraavaksi käydään läpi, minkälaisia hyötyjä automatisoinnin kehitystyöllä on ollut kehittäjän ja loppukäyttäjän näkökulmasta.

### 5.1 Hyödyt kehittäjille

Automatisoitujen vaiheiden merkittävin hyöty kehittäjien kannalta on ajansäästö. Aiemmin koneiden provisiointiprosessi oli pitkä ja kesti kauan. Kehittäjä lähetti provisiointipyynnön sähköpostitse virtuaalikoneiden provisioinnista vastaavalle henkilölle. Provisioinneista vastaava henkilö teki provisiointeja muiden töiden ohella, joten provisiointipyynnöissä kului vaihteleva määrä aikaa. Kun virtuaalikoneen provisiointi oli valmis, kehittäjä sai tiedot uudesta virtuaalikoneesta sähköpostilla ja pystyi jatkamaan siitä mihin jäi. Kehittäjältä kului aikaa provisiointipyynnön tekemiseen. Työnkulku myös katkesi usein, koska provisioinnin valmistumista piti odottaa. ”Virtuaalikoneiden provisiointi itsepalveluna” -toiminnallisuuden avulla kehittäjät voivat itse provisioida virtuaalikoneita oman aikataulun mukaan ja he saavat ne heti käyttöön. Lisäksi provisioinneista vastaava henkilö voi myös keskittyä muihin työtehtäviin.

Myös automatisoitu kuormantasaus säästää kehittäjien aikaa. Nginx-välityspalvelimen konfiguraatioita päivitetään automaattisesti Consul-palvelun avulla. Kehittäjien ei tarvitse käsin muokata konfiguraatioita. Automatisoinnin kehitystyöstä hyödytään lisää, jos esimerkiksi orkestroidaan virtuaalikoneiden viikoittainen uudelleenprovisiointi. Silloin kehittäjien ei tulisi muun muassa enää huolehtia tietoturvapäivityksistä ja virtuaalikoneille kasautuvista turhista lokitiedostoista.

### 5.2 Hyödyt loppukäyttäjille

Kehittäjien lisäksi myös palveluiden loppukäyttäjät hyötyvät automatisoinnin kehitystyöstä. Automatisoinnin myötä kehittäjillä on enemmän aikaa käytettävissä muihin työtehtäviin. Palveluiden kehitysvauhti muun muassa nopeutuu, eli loppukäyttäjät saavat nopeammin uusia palveluita. Seuraavassa kappaleessa pohditaan ja esitetään jatkokehitysehdotus,

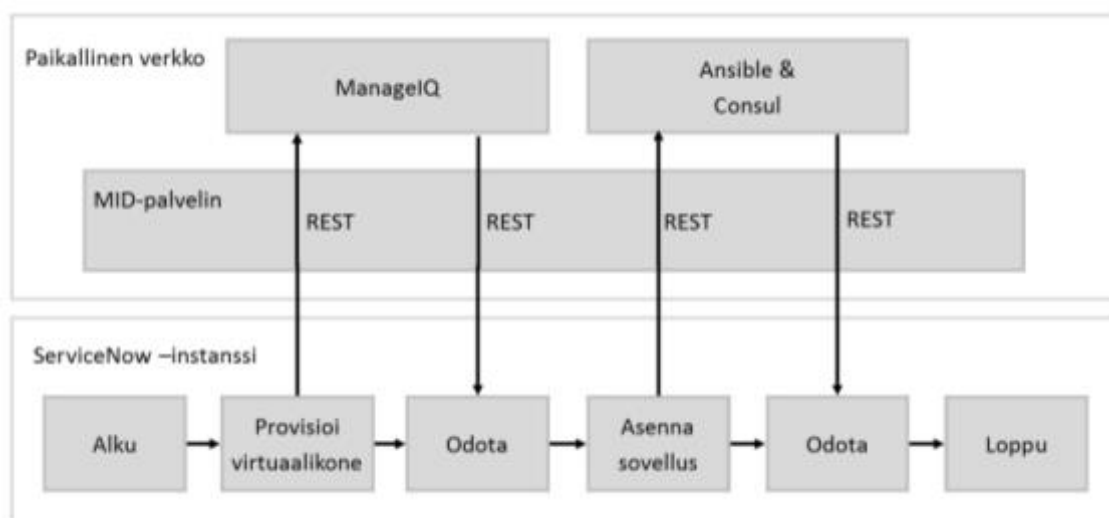
miten automatisoinnin kehitystyötä voidaan viedä vielä pidemmälle ja mitä hyötyjä voidaan saavuttaa.

## 6 Jatkokehitysehdotus

Tässä luvussa esitellään, kuinka automatisoinnin kehitystyötä voitaisiin jatkaa jatkokehitysehdotuksen muodossa. Uusien palveluiden kehitysprosessissa on kolme vaihetta, jotka ovat 1) automatisoitu, virtuaalikoneen provisiointi, 2) sovelluksen asennus virtuaalikoneelle ja 3) sovelluksen lisääminen välityspalvelimen konfiguraatioon. Jotta aiemmin luvussa 4 kuvattu virtuaalikoneiden uudelleenprovisiointi viikoittain voitaisiin toteuttaa, koko prosessi pitää ensin automatisoida eli orkestroida. IT-operaatioiden hallintaan tarkoitettujen työkalujen avulla voidaan muun muassa orkestroida monimutkaisia prosesseja. Lisäksi niiden avulla voidaan parantaa näkyvyyttä palveluiden tilaan, nopeuttaa virhetilanteiden käsittelyä ja diagnosointia.

Seuraavaksi esitellään, miten uusien palveluiden kehitysprosessi voitaisiin orkestroida ServiceNow-sovellusalustan avulla ja mitä hyötyjä voidaan saavuttaa. ServiceNow-sovellusalustan käyttöönotto edellyttää ServiceNow-instanssin hankkimista sekä MID-palvelimen asennusta osaksi IT-infrastruktuuria. ServiceNow-instanssi kommunikoi tietojärjestelmien kanssa MID-palvelimen välityksellä.

ServiceNow-sovellusalustan ominaisuuksiin kuuluu muun muassa työjonojen luominen, niiden avulla voidaan orkestroida prosesseja. Työjonot ovat ketjuja, jotka koostuvat aktiviteeteista, ne ovat ohjelmoituja tehtäviä, jotka voivat sisältää ehtoja. Työjonot voidaan ajastaa toistumaan esimerkiksi kerran viikossa. Tässä projektissa työjono voisi olla seuraavanlainen, katso kuviota 12.



Kuvio 12. Esimerkki siitä minkälainen työjono voisi olla

Työjono, sen sisältämät aktiviteetit sekä työjonon ajastus määritellään ServiceNow-instanssissa ja tallennetaan konfiguraatietietokantaan. Työjonon ajaminen tapahtuu ServiceNow-instanssin toimesta. Työjonon sisältämät aktiviteetit, jotka kommunikoivat tietojärjestelmän kanssa, ajetaan MID-palvelimen avulla kohdekoneille. ServiceNow-sovellusalusta tukee muun muassa REST-kutsuja.

ServiceNow-sovellusalustan orkestrointiominaisuuksien lisäksi sovellusalusta tarjoaa ominaisuuksia, joiden avulla voitaisiin parantaa näkyvyyttä palveluiden tilaan ja nopeuttaa sekä parantaa virhetilanteiden käsittelyä. Esimerkiksi Discovery-sovelluksen avulla voitaisiin etsiä verkosta virtuaalikoneita ja palveluita sekä tallentaa tiedot niistä konfiguraatietietokantaan. Service Mapping -sovelluksen avulla verkosta löytyneistä virtuaalikoneista ja palveluista voidaan luoda kuvioita, joiden avulla voidaan hahmottaa infrastruktuurin rakennetta. Lisäksi kuvioiden avulla voidaan paikantaa virhetilanteita ja sitä, mihin palveluihin virhetilanteet vaikuttavat. ServiceNow-sovellusalustan kaltaisella tuotteella tietojärjestelmän automatisointia ja orkestrointia voitaisiin viedä vielä nykyistäkin pidemmälle.

## 7 Pohdinta

Tämän opinnäytetyöprojektin tuloksena asiakkaan automatisoinnin kehitystyötä on viety eteenpäin. Automatisoinnin kehitystyö on luonteeltaan jatkuvaa kehitystä ja työ jatkuu opinnäytetyöprojektin jälkeekin.

Opinnäytetyöprojektin edistäminen oli ajoittain hidasta. Automatisoinnin kehitystyön kohteena oli laaja prosessikokonaisuus. Prosessin eri vaiheiden automatisoinnin työstäminen edellytti usean eri henkilön yhteistyötä. Monen henkilön aikataulujen sovittaminen yhteen oli ajoittain hankalaa. Tämän kaltaista projektia varten olisi hyvä nimittää varsinainen projektitiimi.

IT-alalla uusia teknologioita ja tekniikoita kehittyy jatkuvasti. Testauksesta ja soveltamisesta pyritään tekemään helppoa ja yksinkertaista. Ympäristössä, johon tämä opinnäytetyöprojekti kohdistuu, tapahtui muutoksia jo opinnäytetyöprojektin aikana. Esimerkiksi Docker-konttitekniologiaa on alettu hyödyntämään ja palveluita on siirretty omasta konesalista julkiseen pilvipalveluun. Automatisoinnin kehitystyötä tehtäessä kannattaa ottaa huomioon uudet teknologiat ja tekniikat, jotka voivat oleellisesti vaikuttaa kehitystyöhön.

Tämän opinnäytetyöprojektin aikana olen tutustunut prosessien kehitykseen ja automatisointiin. Erityisesti tässä projektissa tuli tutuksi uusien palveluiden kehitysprosessi. Se alkaa uuden palvelun kehityksestä ja päättyy, kun palvelu on tarjolla loppukäyttäjille. Olen tutustunut useisiin tuotteisiin ja palveluihin, jotka ovat osa tätä kokonaisuutta. Lisäksi olen oppinut huomaamaan kehityskohteita prosesseissa ja prosessien eri vaiheissa, sekä suunnittelemaan automatisointiratkaisuja, joiden avulla saavutetaan ajallisia ja rahallisia säästöjä.

## Lähteet

Ansible. 2016. How Ansible works. Luettavissa: <http://www.ansible.com/how-ansible-works>. Luettu 23.5.2016.

Gartner. 2016. Cloud Management Platforms. Luettavissa: <http://www.gartner.com/it-glossary/cloud-management-platforms>. Luettu 23.5.2016.

Gartner. 2016. ITOM (IT Operations Management) Software. Luettavissa: <http://www.gartner.com/it-glossary/itom-it-operations-management-software/>. Luettu: 18.10.2016.

HashiCorp. 2016. Consul architecture. Luettavissa: <https://www.consul.io/docs/internals/architecture.html>. Luettu 18.10.2016.

Lau, W. 2011. A comprehensive Introduction to Cloud Computing. Luettavissa: <https://www.simple-talk.com/cloud/cloud-development/a-comprehensive-introduction-to-cloud-computing/>. Luettu 18.10.2016.

ManagelQ. 2016. Luettavissa: <https://pemcg.gitbooks.io/mastering-automation-in-cloudforms-and-manageiq/content/chapter1.html>. Luettu: 18.10.2016.

Martin, N. 2013. Let's get this straight: VM live migration. Luettavissa: <http://searchservvirtualization.techtarget.com/feature/Lets-get-this-straight-VM-live-migration>. Luettu 18.10.2016.

Martin, N. 2013. Virtualization vs. cloud: Let's get this straight. Luettavissa: <http://searchservvirtualization.techtarget.com/feature/Virtualization-vs-cloud-Lets-get-this-straight>. Luettu 23.5.2016.

Nginx. 2016. What is a reverse proxy vs. Load balancer. Luettavissa: <https://www.nginx.com/resources/glossary/reverse-proxy-vs-load-balancer/>. Luettu 18.10.2016.

Red Hat. 2015. Red Hat CloudForms 3.1 Management Engine 5.3 Lifecycle and Automation Guide. Luettavissa: [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_CloudForms/3.1/pdf/Management\\_Engine\\_5.3\\_Lifecycle\\_and\\_Automation\\_Guide/Red\\_Hat\\_CloudForms-3.1-](https://access.redhat.com/documentation/en-US/Red_Hat_CloudForms/3.1/pdf/Management_Engine_5.3_Lifecycle_and_Automation_Guide/Red_Hat_CloudForms-3.1-)



Management\_Engine\_5.3\_Lifecycle\_and\_Automation\_Guide-en-US.pdf. Luettu: 18.10.2016.

Rouse, M. 2014. Cloud bursting. Luettavissa: <http://searchcloudcomputing.techtarget.com/definition/cloud-bursting>. Luettu 23.5.2016.

Rouse, M. 2015. Infrastructure as a Service (IaaS) definition. Luettavissa: <http://searchcloudcomputing.techtarget.com/definition/Infrastructure-as-a-Service-IaaS>. Luettu 23.5.2016.

Rouse, M. 2015. Platform as a Service (PaaS) definition. Luettavissa: <http://searchcloudcomputing.techtarget.com/definition/Platform-as-a-Service-PaaS>. Luettu 23.5.2016.

Rouse, M. 2015. Private cloud (internal cloud or corporate cloud). Luettavissa: <http://searchcloudcomputing.techtarget.com/definition/private-cloud>. Luettu 23.5.2016.

Rouse, M. 2016. Software as a Service (SaaS). Luettavissa: <http://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service>. Luettu 18.10.2016.

Sajan, S. 2016. Physical Vs Virtual server: Which one should you choose?. Luettavissa: <https://bobcares.com/blog/virtualization-or-physical-server/>. Luettu 23.5.2016.

ServiceNow. 2016. Discovery. Luettavissa: <http://wiki.servicenow.com/index.php?title=Discovery#gsc.tab=0>. Luettu 23.5.2016.

ServiceNow. 2016. Get started with Service Mapping. Luettavissa: [https://docs.servicenow.com/product/service\\_mapping/concept/c\\_ServiceMappingOverview.html](https://docs.servicenow.com/product/service_mapping/concept/c_ServiceMappingOverview.html). Luettu 23.5.2016.

ServiceNow. 2016. IT Operations Management. Luettavissa: [https://docs.servicenow.com/product/service\\_mapping/concept/c\\_ServiceMappingOverview.html](https://docs.servicenow.com/product/service_mapping/concept/c_ServiceMappingOverview.html). Luettu 18.10.2016.

ServiceNow. 2016. Orchestration Activities. Luettavissa: [http://wiki.servicenow.com/index.php?title=Orchestration\\_Activities#gsc.tab=0](http://wiki.servicenow.com/index.php?title=Orchestration_Activities#gsc.tab=0). Luettu 18.10.2016.

Vizard, M. 2012. Moving from SaaS to PaaS. Luettavissa:

<http://www.itbusinessedge.com/cm/blogs/vizard/moving-from-saas-to-paas/?cs=50265>.

Luettu 23.5.2016.

VMware. 2016. VMware vSphere, a Platform for Virtualization and Cloud Infrastructure.

Luettavissa: [https://pubs.vmware.com/vsphere-](https://pubs.vmware.com/vsphere-50/index.jsp#com.vmware.vsphere.introduction.doc_50/GUID-AFFD451D-F9B8-4EEE-BB9C-BA69CEC43BC2.html)

[50/index.jsp#com.vmware.vsphere.introduction.doc\\_50/GUID-AFFD451D-F9B8-4EEE-BB9C-BA69CEC43BC2.html](https://pubs.vmware.com/vsphere-50/index.jsp#com.vmware.vsphere.introduction.doc_50/GUID-AFFD451D-F9B8-4EEE-BB9C-BA69CEC43BC2.html). Luettu 23.5.2016.

VMware. 2016. VMware vSphere Components and Features.

[https://pubs.vmware.com/vsphere-](https://pubs.vmware.com/vsphere-50/index.jsp#com.vmware.vsphere.introduction.doc_50/GUID-A8B8E6DC-A881-4F14-AEC1-D17365731E68.html)

[50/index.jsp#com.vmware.vsphere.introduction.doc\\_50/GUID-A8B8E6DC-A881-4F14-AEC1-D17365731E68.html](https://pubs.vmware.com/vsphere-50/index.jsp#com.vmware.vsphere.introduction.doc_50/GUID-A8B8E6DC-A881-4F14-AEC1-D17365731E68.html). Luettu 23.5.2016.