TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

TIMO MÄHÖNEN

OHJELMISTOYRITYKSEN TIETOJÄRJESTELMIEN, TIETOTURVAN JA VALVONNAN KEHITYS

Diplomityö

Tarkastaja: professori Jarmo Harju
Tarkastaja ja aihe hyväksytty
Tieto- ja sähkötekniikan
tiedekuntaneuvoston kokouksessa
4. kesäkuuta 2014

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Tietotekniikan koulutusohjelma
MÄHÖNEN, TIMO: Ohjelmistoyrityksen tietojärjestelmien, tietoturvan ja valvonnan kehitys
Diplomityö, 54 sivua
Marraskuu 2014
Pääaine: Tietoliikennetekniikka
Tarkastaja: professori Jarmo Harju
Avainsanat: tietoliikenne, tietoturva, identiteetin- ja pääsynhallinta, virtualisointi, pilvipalvelut, ohjelmistoyritys

Nykyisten ohjelmistoja tuottavien konsulttiyritysten toiminta-alue on todella laaja. Ohjelmistoja tuottavan yrityksen odotetaan asiakkaan näkökulmasta tarjoavan myös monipuoliset tietojärjestelmät ja työkalut, jotka täyttävät nykyaikaisen projektityön vaatimukset. Tällaisten tietojärjestelmien ja niihin liittyvän teknologian hallinta ei kuitenkaan ole varsinkaan pienille ohjelmistoyrityksille ydinosaamista. Tässä työssä tutkitaan millaisia ulkoisia edellytyksiä tietojärjestelmien ajamisella on. Lisäksi tavoitteena on kerätä tietoa järjestelmistä vastaavalle ylläpidolle, jotta tietojärjestelmiin liittyvä kehitystyö olisi mahdollista.

Työ alkaa tietojärjestelmien ja sen osien määrittelemisellä. Teoria- ja määritelmä-osiossa tutustutaan verkon arkkitehtuuriratkaisuihin, virtualisointiin, pilvipalveluihin sekä erinäisiin tietoturvan osa-alueisiin. Näitä osa-alueita käydään läpi myöhemmin käytännön esimerkkien sekä case-tutkimusten kautta.

Case-tutkimuksia tehtiin kaksi kappaletta. Molemmissa caseissa kuvataan ensin nykytila, käydään läpi syyt muutokselle, esitellään eri vaihtoehdot ja lopulta käydään läpi valittu vaihtoehto.

Ensimmäinen case käsittelee Atostek Oy:n työntekijöiden etäyhteydet mahdollistavan VPN-palvelun päivitystä. Casessa käydään läpi päivitys vanhasta IPsec-pohjaisesta ratkaisusta uuteen SSL/TLS-pohjaiseen ratkaisuun.

Toinen case käsittelee organisaation käyttöön tulevan sisäisen pilvipalvelun toteuttamista avoimen pilvipalvelualustan päälle. Alustaksi valikoitui OpenStack. Pilvipalvelun tarkoituksena on mahdollistaa itsepalveluperiaatteen mukainen virtuaalikoneiden toimitus ja näin ollen ketteröittää organisaation projektien toimintaa sekä helpottaa ylläpidon työtaakkaa.

Työssä saatiin organisaation käyttöön uutta tietoa kaikilta tutkituilta osa-alueilta. Opittuja menetelmiä ja ratkaisuja on päästy hyödyntämään käytännössä niin hallinnollisella puolella kuin jokapäiväisessä organisaation työssäkin. Työn tulokset ovat osoittautuneet hyödyllisiksi, koska niiden seurauksena ylläpidon työtaakka on keventynyt ja hallinnolla on parempi kokonaiskuva organisaation pääsynhallinnasta. Lisäksi ylläpidolla on nyt valmiudet kehittää ja toimittaa projektien ja niiden asiakkaiden tarvitsemat työkalut ja tietojärjestelmät entistä tehokkaammin.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Information Technology
MÄHÖNEN, TIMO: Development of security, monitoring and information systems in a software company
Master of Science Thesis, 54 pages
NOVEMBER 2014
Major: Communications engineering
Examiner: Professor Jarmo Harju
Keywords: networking, information security, identity and access management, virtualization, cloud services, software company

The operating range of current software consulting companies is really broad. Customers are expecting that the company who is providing software consulting also provide the tools and information systems to meet the requirements of the modern project work. Usually running such systems does not belong the core competence of these companies. The main objective of this thesis is to examine external conditions and collect information to the staff whose duty in the organization is to run those tools and systems.

The thesis begins by defining the information systems and related components. During theory and definitions part of the thesis we will take a closer look on different network architectures, virtualizations, cloud services and related information security. And after that we show how those concepts and theory can be used in practice. Practical part of the thesis consists of real world examples and two case studies.

There were two case studies completed during this thesis. Both cases are structured so that first we go through the current state of the system. After that we list the main reasons for the change and present different options to implement the change. And finally we go through the steps to implement the chosen option.

First case is about the upgrade of the Atostek Oy VPN-system from old IPsec based solution to the new SSL/TLS based solution.

Second case is about implementing in premise private cloud service to Atostek Oy. The chosen solution is built on OpenStack and it is aiming to offer self-service-like cloud services to speed up the delivery of virtual machines and also decrease the workload of company's IT-staff.

During this thesis we managed to gather new knowledge from all studied subareas. Learned methods and solutions have been successfully utilized in the management and operational parts of the organization. Results of this thesis have been proved to be useful and as a result IT-staff has now the tools and methods to successfully deliver and develop information systems and tools related to the needs of customers and projects.

ALKUSANAT

Tätä diplomityötä varten tehty työ aloitettiin alkusyksystä vuonna 2013 ja itse kirjoitustyö on tehty vuoden 2014 aikana. Tahdon kiittää oikoluvusta, kommenteista ja ideoista ohjaaja Jaakko Perkiötä sekä tarkastaja Jarmo Harjua. Erityisesti haluan kiittää äitiäni Toini Mähöstä ja tätiäni Eeva Väisästä opiskeluideni tukemisesta sekä Paula Väätäistä vähän kaikesta. Lisäksi haluan kiittää Atostek Oy:tä tämän työn rahoittamisesta sekä työkavereitani diplomityön tekemiseen motivoimisesta.

Tampereella 9.10.2014

Timo Mähönen
timo@puijo.biz

SISÄLLYS

TERMIT JA LYHENTEET

Active Directory	Microsoftin Windows-toimialueen käyttäjätietokanta sekä hakemistopalvelu. Sisältää tiedot käyttäjistä, tietokoneista sekä verkon resursseista.
Broadcast-domain	Looginen verkon osa, jonka alueella kaikki laitteet voivat tavoittaa toisensa linkkikerroksella. Käytännössä reitittimet ja muut korkeamman kerroksen laitteet erottavat broadcast-domainit toisistaan.
DMZ	Demilitarized Zone
Hypervisor	Hypervisor on fyysiselle koneelle asennettu käyttöjärjestelmä, joka huolehtii virtualisoinnista.
IaaS	Infrastructure as a Service
IAM	Identity and Access Management
IDS	Intrusion Detection System
IPsec	Internet Protocol security
IPS	Intrusion Prevention System
JSON	JavaScript Object Notation, yksinkertainen avoimen standardin tiedostomuoto tiedonvälitykseen.
LDAP	Lightweight Directory Access Protocol on hakemistopalveluiden, kuten Active Directory, käyttöön tarkoitettu verkkoprotokolla.
NaaS	Network as a Service
PaaS	Platform as a Service
RADIUS	Remote Authentication Dial In User Service
RBAC	Role Based Access Control
RMF	Risk Management Framework
SaaS	Software as a Service
SDDC	Software Defined Data Center
SNMP	Simple Network Management Protocol
SPB	Shortest Path Bridging
SSL/TLS	Security Sockets Layer/Transport Layer Security
SSO	Single Sign On, kertakirjautuminen
STP	Spanning Tree Protocol
TRILL	Transparent Interconnection of Lots of Links
VLAN	Virtuaalilähiverkko on tekniikka, jolla fyysinen tietoverkko voidaan jakaa loogisiin osiin. VLANien käyttö vaatii tuen kytkimiltä ja reitittimiltä. VLAN ID erottaa eri VLANit.
VPN	Virtual Private Networks
VRF	Virtual Routing and Forwarding
vSwitch	Virtuaalinen verkkokytkin

1 JOHDANTO

Ohjelmistoja tuottavien konsulttiyritysten työkenttä on nykyään todella laaja. Asiakkaat odottavat että ohjelmistoja tarjoava asiantuntijayritys pystyy toimittamaan tilatun ohjelmiston lisäksi myös kaikki tarvittavat työkalut, jotka täyttävät nykyaikaisen projektityön vaatimukset. Tällaisten tietojärjestelmien ja niihin liittyvän teknologian hallinta ei kuitenkaan ole varsinkaan pienille ohjelmistoyrityksille ydinosaamista. Tietojärjestelmien kannalta pilvipalveluiden ja virtualisoinnin hyödyntäminen laajassa mittakaavassa on kustannustehokkuuden takia välttämätöntä. Lisäksi viimeistään nykyisen maailmantilanteen vuoksi alalla kuin alalla on herätty tietoturvan kasvaneisiin vaatimuksiin, joiden syvällisempi ymmärtäminen on hyvin monesti jätetty puhtaasti konsulttiyritykselle.

Tässä diplomityössä ei pohdita millaisia itse käytettävien tietojärjestelmien tai niiden sisällön tulisi olla vaan tarkoituksena on keskittyä ulkoisien edellytysten pohdiskeluun sekä kerätä tietojärjestelmien ylläpitäjille tärkeää tietoa eri alueilta jatkokehitystä varten. Tarkemmin sanottuna diplomityö rajautuu siihen millaisia, ohjelmistoyritysten tietojärjestelmien kannalta, tietoverkkoja, pilvipalveluita ja erilaisia virtualisointeja on olemassa sekä millaisia kehityksen ja käytön kannalta välttämättömiä tietoturva- ja valvontaratkaisuja on tarjolla.

Työn perustana toimii työn tilaajaorganisaation eli Atostek Oy:n ylläpitäjänä toimivien henkilöiden huomioimat puutteet ja kehityskohteet samaisen ohjelmistoyrityksen tietojärjestelmiin liittyvän työn puitteissa. Työn tavoitteina ovat ylläpitotyöhön liittyvän kokonaiskuvan kehittäminen ja lyhyen aikavälin mahdollisuuksien kartoittaminen, joilla ylläpitotyötä voitaisiin järkevöittää ja helpottaa niin tietoverkkojen, virtualisoinnin kuin tietoturvan kannalta.

Tietoverkot käydään läpi teoreettisella tasolla tutustumalla tietoverkon suunnittelumalleihin. Pilvipalveluista ja virtualisoinnista käydään läpi niiden luonne, rakenne ja erilaiset toteutusmallit. Tietoturva rajataan käsittelemään verkon tietoturvaa, identiteetin- ja pääsynhallintaa sekä tietojärjestelmien valvontaa. Aihealueiden käsittelyssä ei mennä syvälle teknisiin yksityiskohtiin vaan pyritään pitämään läpikäynti käsitteellisellä tasolla.

Rakenteellisesti työ jakautuu seitsemään lukuun, joista ensimmäisenä toimii tämä luku eli johdanto. Luvussa 2 käydään läpi työn rajauksen mukaisesti tietoverkkoja, virtualisointia sekä erilaisia pilvipalvelukonsepteja ja -malleja. Luvussa 3 perehdytään tietoturvaan verkon tietoturvan, identiteetin- ja pääsynhallinnan, organisaation riskienhallinnan sekä valvonnan kautta. Luku 4 sisältää aikaisempien lukujen aihealueiden sovellusta käytännössä esimerkkiorganisaation avulla. Luku 5 sisältää kaksi erilaista Atostek Oy:lle tehtyjä case-tutkimusta, joista ensimmäisessä käydään läpi kyseisen or-

ganisaation VPN-palvelun (Virtual Private Networks) päivitys ja jälkimmäisessä organisaation yksityisen pilvipalvelu konseptia. Luvussa 6 analysoidaan ja arvioidaan diplomityön aikana saatuja tuloksia sekä käydään läpi jatkokehitysideoita. Luku 7 sisältää yhteenvedon työn tuloksista.

2 OHJELMISTOYRITYKSEN TIETOJÄRJESTELMÄT

Tässä luvussa käydään läpi tämän diplomityön kannalta tietojärjestelmän määritelmä sekä sen toteuttavat osat kuten tietoverkko, pilvipalvelut ja virtualisointi. Tässä luvussa pitäydytään vielä pitkälle yleisessä näkökulmassa ja teoriassa eikä vielä syvennytä pelkästään ohjelmistoyrityksen erityistarpeisiin.

Tietoverkoista käydään läpi teoreettisella tasolla tietoverkon rakenne verkkotyyppeineen. Lisäksi käydään hieman läpi perinteistä verkon suunnittelumallia sekä otetaan katsaus tulevaan. Pilvipalveluista käydään läpi niiden luonne ja rakenne sekä erilaiset pilvimallit ja pilvipalveluiden palvelumallit. Virtualisoinnista käydään läpi erilaiset virtualisointitekniikat sekä se mistä koko virtualisoinnissa on kyse.

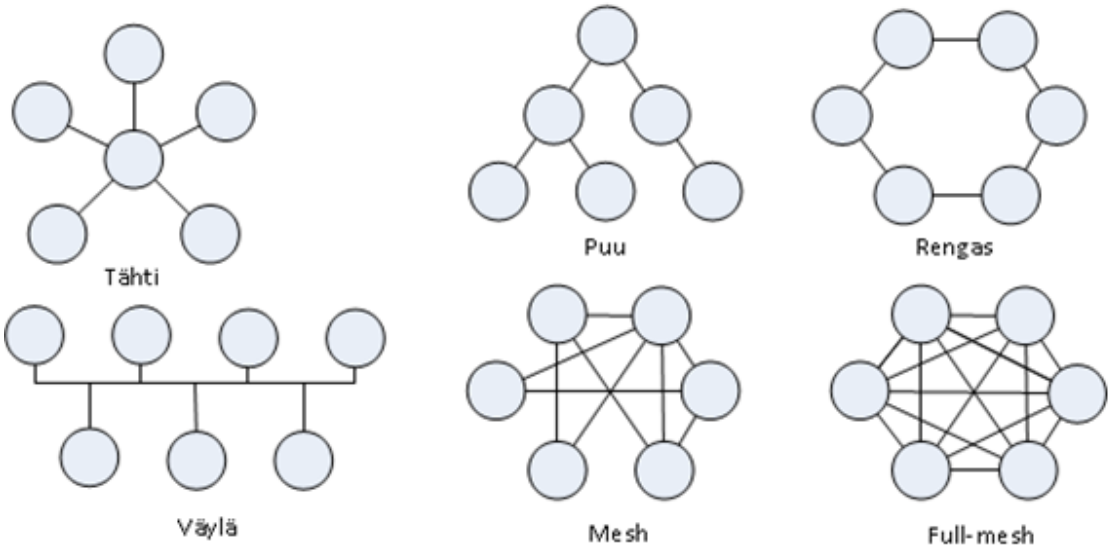
2.1 Tietojärjestelmän määritelmä

Tietojärjestelmä on monista erilaisista ohjelmistokomponenteista ja laitteistoista koostuva järjestelmä, jonka tarkoituksena on kerätä, tallentaa ja käsitellä sitä informaatiota mitä varten kyseinen tietojärjestelmä on luotu. Tietojärjestelmän on siis tarkoitus mahdollistaa tehokas ja helppo toiminta, joka liittyy kyseiseen tiedon käyttämiseen. Tietojärjestelmät ovat yleensä laajempia ja monitahoisempia kuin pelkät tietokoneohjelmistot, joiden yhteydessä niistä puhutaan. Ohjelmistoyrityksissä tietojärjestelmiä ovat esimerkiksi tehtävienhallintaohjelmisto Atlassian JIRA sekä versionhallinta Subversion. Näiden eri järjestelmien välille voidaan luoda yhteys jolloin puhutaan tietojärjestelmien integroinnista. Tässä diplomityössä tietojärjestelmien kannalta keskitytään tietojärjestelmät yhdistäviin tietoliikenneverkkoihin, tietojärjestelmien tarvitsemiin pilvipalvelualustoihin ja virtualisointiratkaisuihin sekä tietojärjestelmiin liittyvään tietoturvallisuuteen identiteetin- ja pääsynhallinnan sekä verkon tietoturvan kannalta. Jotta ymmärtäisimme tietojärjestelmien kokonaiskuvan määrittelemme ja paneudumme seuraavissa alaluvuissa tietojärjestelmien tietoverkkoihin, pilvipalveluihin, virtualisointitekniikoihin sekä tietoturvaan.

2.2 Tietoverkon määritelmiä

Tietoverkolla liitetään yhteen siinä olevia laitteita, jolloin mahdollistetaan tietoverkon sisäinen liikennöinti laitteiden välillä. Tietoverkon perusrakenne koostuu solmuista ja solmuja toisiinsa liittävistä yhteysväleistä. Solmut voivat olla verkon aktiivilaitteita kuten reitittimiä, kytkimiä ja langattomia tukiasemia. Solmut voivat olla myös päätelaittei-

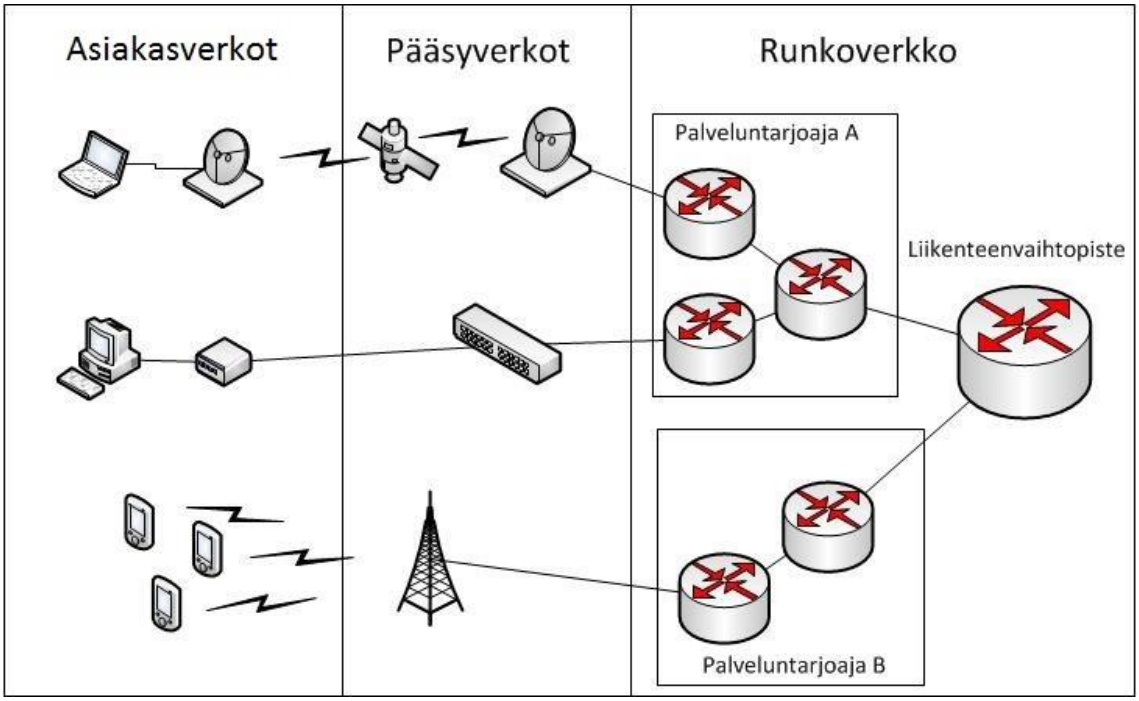
ta kuten työasemia tai servereitä. Tietoverkkoja on monen tyyppisiä, mutta tämän diplomityön aihepiirissä ovat ainoataan IP-pohjaiset tietoverkot. Ylemmällä tasolla kuvatessa solmut voivat olla myös yksittäisiä pienempiä verkkoja. Tietoverkon kannalta solmut ja yhteysvälit muodostavat verkon topologian joiden yleisimpiä malleja on kuvattu kuvassa 2.1. Näitä ovat esimerkiksi tähti, puu, rengas, mesh, full-mesh ja väylä.



Kuva 1: Tietoverkko topologioiden yleisimmät mallit

Reititystä miettiessä helpoin näistä toteuttaa ja laajentaa on rengastopologia. Rengastopologiassa liikennevuot ovat helposti arvattavia, mutta huonona puolena verkon konvergenssi on hidasta. Full-mesh topologia sen sijaan on topologioista kaikkein redundantein virheitä vastaan. Samalla se on myös kallein ja työläin toteuttaa, koska kaikki solmut ovat liittyneinä toisiinsa. Käytännössä siis uuden solmun lisääminen topologioihin tarkoittaa rengastopologiassa konfigurointityötä vain naapurisolmuissa kun full-mesh topologiassa vaaditaan työtä kaikissa verkon solmuissa.

Perinteisesti eri verkot on jaettu kolmeen osaan: asiakasverkkoihin (engl. customer network), pääsyverkkoihin (engl. access network) ja runkoverkkoihin (engl. backbone network). Ylimmällä tasolla ajatellen asiakasverkot käsittävät asiakkaiden kuten yritysten tai kotikäyttäjien omat verkot. Pääsyverkoilla tarkoitetaan verkkoja, joilla asiakasverkot yhdistetään runkoverkkoon. Runkoverkoilla puolestaan tarkoitetaan palveluntarjoajien (engl. service provider) ja runkoverkko-operaattoreiden verkkoja. Operaattoreiden runkoverkot liittyvät toisiinsa joko ostamalla siirtokapasiteettia toisista verkoista (engl. transit), suoraan peeraamalla (engl. peering) tai yhteisten liikenteen vaihtopisteiden kautta (engl. internet exchange point). Sekä asiakas-, pääsy- että runkoverkot voidaan jakaa vielä pienempiin osiin. Esimerkiksi suuremman yrityksen tai yliopiston verkko saattaa sisältää sekä asiakas-, pääsy- että runkoverkkoja. Lisäksi jokainen näistä verkoista voidaan jakaa vielä tarkemmin toiminallisiin osiin kuten ylläpito- ja tuotantoverkko. Kuvassa 2 on esitetty esimerkki IP-verkkojen jaosta asiakas-, pääsy- ja runkoverkkoihin [1][2].

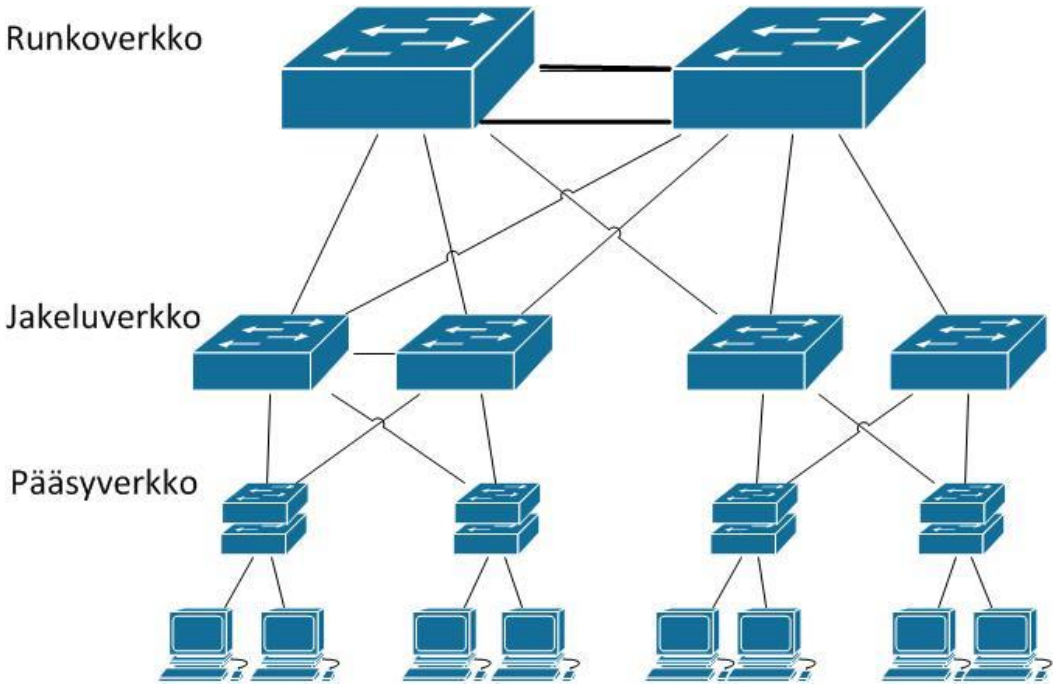


Kuva 2: Suosituksen ITU Y.1001 mukainen IP-verkkojen jako [2].

Nykyaikaisissa langallisissa verkoissa päätelaitteet kuten työasemat ja palvelimet ovat liitettyinä tähtitopologian mukaisesti. Lähiverkkojen aktiivilaitteet muodostavat yleensä tähtiä tai puita. Pääsyverkkojen topologiaksi muodostuu tähti, puu tai väylä. Runkoverkoissa tarvittavan luotettavuuden aikaan saamiseksi topologiaksi valitaan yleensä rengas tai mesh, jolloin yhden linkin katkeaminen ei aiheuta koko verkon hajoamista. Reaalimaailman tilanteessa tietoverkot kuitenkin harvoin ovat topologioidensa kannalta näin yksioikoisia vaan ne sisältävät usein piirteitä useammasta eri topologiasta. Seuraavissa alaluvuissa esitellään jo vuosia verkkosuunnittelussa alan de facto standardina toiminut hierarkkinen suunnittelumalli sekä uudempi konesalien kasvaviin tarpeisiin kehitetty lehti-ranka suunnittelumalli (engl. spine-leaf model) sekä sen muunnelmat.

2.2.1 Tietoverkon hierarkkinen suunnittelumalli

Tietoverkot ovat nykyään kriittinen osa yritysten toimintaa, joten tietoverkon hallinta ja ylläpito on syytä toteuttaa toimivasti. Organisaation hallinnoimat tietoverkko voidaan jakaa kolmeen toiminnalliseen kerrokseen: pääsy- (engl. access), jakelu- (engl. distribution) ja runkoverkko (engl. core). Tätä samaa mallia voidaan soveltaa niin pienissä kuin suurissakin verkoissa. Lisäksi tämä malli mahdollistaa verkon rakenteen modulaarisuuden, jolloin verkon skaalautuvuus ja tehokkuus paranee. Esimerkki hierarkkisesta verkkosuunnittelusta on esitetty kuvassa 3 [4].



Kuva 3: Esimerkki hierarkkisesta lähiverkon suunnittelusta [4].

Tässä mallissa verkon eri toiminnollisuudet on jaettu seuraavasti:

Pääsyverkko

Pääsyverkkokerros toimii suorassa yhteydessä loppukäyttäjien päätelaitteisiin kuten työasemiin, tulostimiin ja IP-puhelimiin. Tärkeimpänä tehtävänä pääsyverkolla on tarjota pääsy verkkoon sekä kontrolloida sitä millä päätelaitteilla on pääsy verkkoon. Pääsyverkon aktiivilaitteita voivat olla esimerkiksi reitittimet, kytkimet, sillat, hubit ja langattoman verkon tukiasemat [4].

Jakeluverkko

Jakeluverkkokerroksen tehtävänä on kerätä pääsyverkon laitteilta tuleva data ennen kuin se siirretään kohti runkoverkkoa ja lopullista päämääräänsä. Jakeluverkko kontrolloi eri verkkoliikenteiden voita säännöstöjen avulla sekä rajaa verkon broadcast-domaineja huolehtimalla pääsyverkko-kerroksella määriteltyjen virtuaalisten lähiverkkojen eli VLANien (Virtual Local Area Networks) välisestä reitityksestä. Broadcast-domainilla tarkoitetaan loogista verkon osaa, jonka alueella kaikki laitteet voivat tavoittaa toisensa linkkikerroksella. Käytännössä reitittimet ja muut korkeamman kerroksen laitteet erottavat broadcast-domainit toisistaan. Lisäksi VLANeilla voidaan segmentoida liikennettä verkkokerroksella eri käyttäjien kesken vaikka käytettäisiin fyysisesti samoja verkon aktiivilaitteita. Esimerkiksi yliopiston verkossa voidaan laitoksen työntekijöiden, opiskelijoiden ja vierailijoiden liikenne jakaa omiin VLANeihinsa.

Jakeluverkossa laitteina toimivat järeämmät kytkimet, joita kutsutaankin monesti keräilykytkimiksi. Yleensä jakeluverkon laitteet on kahdennettu redundanssin ja korkean saatavuuden turvaamiseksi [4].

Runkoverkko

Hierarkkisen verkkosuunnittelumallin runkoverkko-kerros on mallin suorituskykyinen ranka. Runkoverkkokerros on kriittinen osa verkkoa, sillä se yhdistää eri jakeluverkot keskenään, joten on tärkeää että runkoverkko on suunniteltu korkea saatavuus sekä redundanttisuus mielessä. Käytännössä tällä tarkoitetaan että runkoverkon linkit ja laitteisto on vähintään kahdennettu. Runkoverkko myös yhdistää koko verkon ulkoisiin resursseihin kuten Internettiin. Runkoverkkokerros kerää liikenteen jakeluverkkokerrokselta, joten sen on pystyttävä välittämään todella suuriakin määriä dataa.

Pienemmissä verkoissa on tavallista että hierarkkista mallia käytetään niin sanotusti luhistuneen ytimen mallina (engl. collapsed core), jolloin jakelu- ja runkoverkkokerros yhdistetään yhdeksi kerrokseksi [4].

2.2.2 Hierarkkisen suunnittelumallin edut

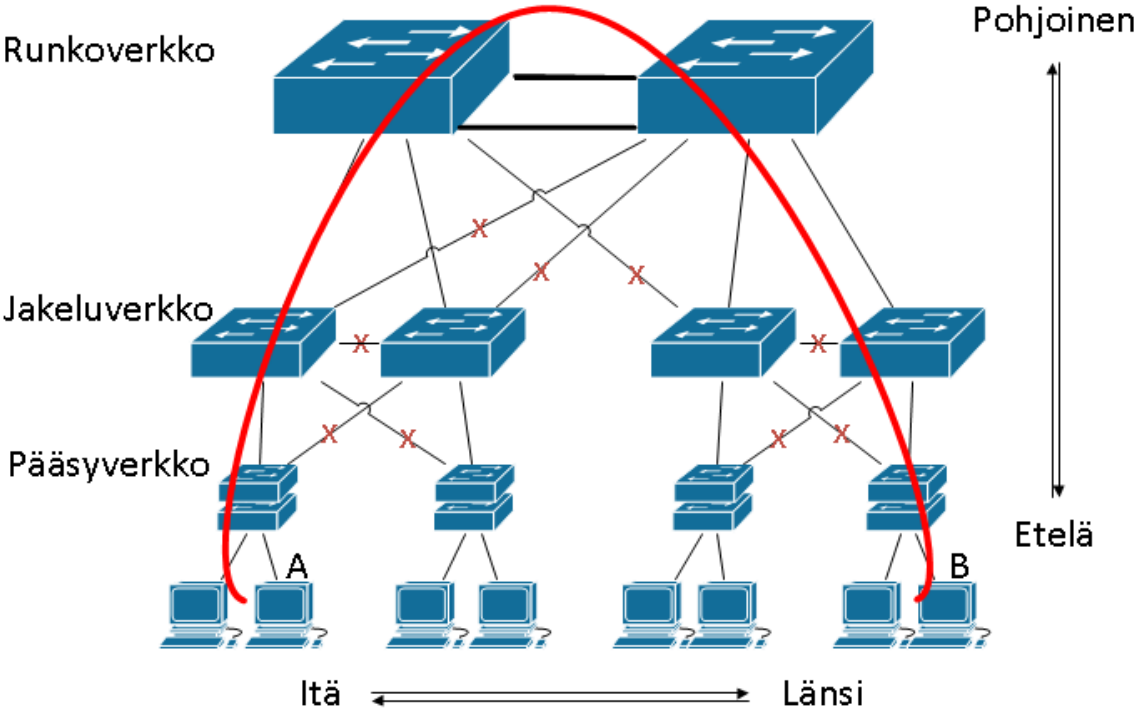
Hierarkkisen suunnittelumallin etuina voidaan pitää kustannustehokkuutta, skaalautuvuutta, redundanttisuutta, suorituskykyä, turvallisuutta, hallittavuutta ja ylläpidettävyyttä. Pienempien organisaatioiden kannalta suurimmat edut kuitenkin painottuvat kustannustehokkuuteen, skaalautuvuuteen ja ylläpidettävyyteen, joista kahta viimeksi mainittua avataan hieman seuraavaksi.

Hierarkkisesti suunniteltu verkko skaalautuu todella hyvin. Mallin modulaarisuus mahdollistaa helpon tavan monistaa verkon osia verkon kasvaessa. Jokaisen verkon osan ollessa johdonmukainen on laajennuksien suunnittelu ja toteutus helppoa. Esimerkiksi jos suunnittelumallissa jokaista kymmentä pääsykerroksen kytkintä varten tarvitaan kaksi ristiin kytkettyä jakelukerroksen kytkintä, voidaan lisätä pääsyverkon kytkimiä kunnes kymmenen pääsyverkon kytkintä tulee täyteen jonka jälkeen lisätään jakeluverkkoon kytkimiä.

Ylläpidettävyys on hierarkkisesti suunnitelluilla verkoilla hyvä, sillä niihin liittyvät huoltotoimenpiteet ja ongelmanratkaisut on helppo ja nopea toteuttaa. Esimerkiksi jos tietty määrä käyttäjistä putoaa verkosta voidaan ongelma rajata nopeasti tiettyyn pääsyverkon kytkimeen, jonka jälkeen päästään käsiksi ongelmaan nopeasti ja korjaus sujuu vaivattomasti joko vaihtamalla kyseinen kytkin tai korjaamalla kytkimeen liittyvä ongelma. Samalla helpottuu uuden kytkimen hankintaan liittyvät ongelmat, sillä eri kerroksien kytkimillä on eri tehtävät, josta seuraa että tietyn kerroksen kytkimeltä vaaditaan vain tietyt ominaisuudet. Samalla säästyy rahaa kun pääsyverkossa voidaan käyttää halvempia kytkimiä kuin jakelu- tai runkoverkossa.

2.2.3 Tietoverkon lehti-ranka arkkitehtuuri

Hierarkkinen suunnittelumalli on ollut jo vuosia perustana verkkojen suunnittelulle ja se onkin osoittautunut erittäin toimivaksi. Sen suurin etu on maksimoida etelä-pohjoissuunnan liikenne (engl. north-south network traffic) eli käytännössä tämä arkkitehtuuri maksimoi verkkoon tulevan sekä lähtevän liikenteen. Nykyisissä konesaleissa itä-länsisuuntaisen (engl. east-west network traffic) liikenteen vaatimukset ovat kasvaneet radikaalisti eli käytännössä konesalien sisällä liikennöidään valtavasti, joten jotain uutta on täytynyt kehittää vastaamaan paremmin nykyisiä tarpeita. Kuvassa 4 on esitetty sekä etelä-pohjoissuuntainen kuin itä-länsisuuntainen liikenne verkossa [18].

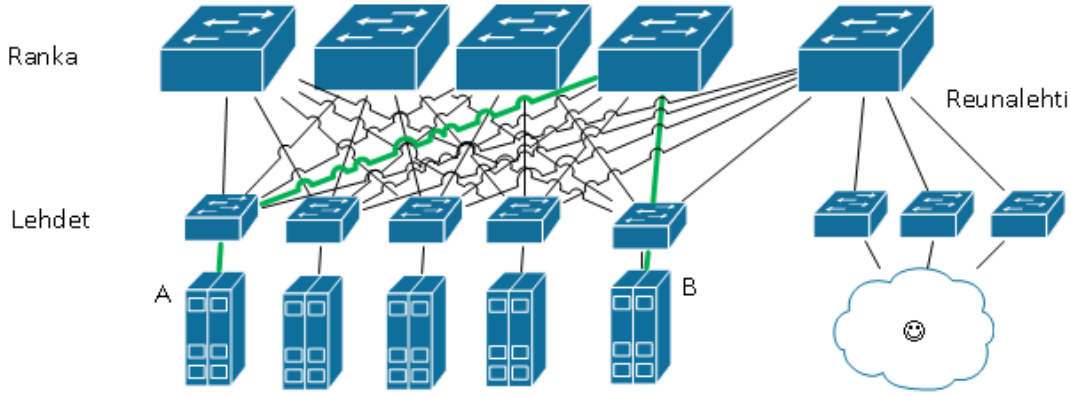


Kuva 4: Verkon erisuuntaiset liikenteet [18].

Otetaan esimerkkinä tilanne, jossa palvelin A (sovelluspalvelin) ja palvelin B (tietokantapalvelin) sijaitsevat samassa suuressa konesalissa, mutta kuitenkin verkkopuun eri haaroissa. Perinteisesti hierarkkisesti suunnitellussa verkossa liikenne joutuu aina kiertämään runkoverkon kautta, jolloin verkkohyppyjen (engl. hop count) määrä nousee suureksi. Kuvassa 4 tämä liikenne on kuvattu punaisella. Verkkolaitteet on yhdistetty toisiinsa useilla erillisillä linkeillä vikasietoisuuden varmistamiseksi, mutta vain yksi linkki on aktiivisesti käytössä muiden ollessa estettyinä johtuen STP:n (engl. Spanning Tree Protocol) käytöstä. STP:n estämät reitit on esitetty kuvassa 4 punaisin rastein. Käytännössä STP varmistaa ettei siirtokerroksella pääse syntymään silmukoita. Tämä kuitenkin johtaa siihen että osa linkeistä jää täysin käyttämättä. STP:n muodostama verkkopuu on aina pohjois-eteläsuuntainen, joten verkon koko kapasiteettia ei päästä hyödyntämään varsinkaan itä-länsisuunnassa [10][11].

Näitä havaittuja ongelmia vastaan kehitetty ranka-lehti arkkitehtuuri, joka tunnetaan myös nimellä jaettu runko (engl. Distributed Core). Tämä arkkitehtuuri perustuu

kahteen eri osaan eli ranka- sekä lehtikytkimiin. Perinteisestä mallista poiketen tässä arkkitehtuurissa kaikki verkkopolut ovat välittäviä. Tämä mahdollistuu kun perinteinen STP korvataan uudemmilla protokollilla kuten TRILL:llä (engl. Transparent Interconnection of Lots of Links) tai SPB:llä (engl. Shortest Path Bridging) eli IEEE 802.1aq:lla, jolloin verkon koko kapasiteetti saadaan tehokkaammin hyödynnettyä. Kuvassa 5 on esitetty yksinkertainen lehti-ranka arkkitehtuuri [11][12][13][28].



Kuva 5: Yksinkertainen ranka-lehti arkkitehtuuri [12].

Lehti

Lehdet voidaan mieltää hiearkkisen mallin pääsyverkoksi eli lehden tarjoavat verkkoyhteyden verkon päätelaitteille eli käytännössä konesalien servereille. Tärkein osa tätä arkkitehtuuria on kuitenkin se että jokainen lehti on kytketty jokaiseen rankakytkimeen. Tästä seuraa se että jokainen lehteen yhdistetty palvelin on kolmen verkkohypyn päästä mistä tahansa muusta serveristä konesalissa kun oletetaan palvelimien olevan eri lehdissä. Esimerkkinä tästä on esitetty liikennöinti A:lta B:lle kuvassa 5. Erikoistapauksena lehdille on reunalehti (engl. border leaf), jonka kautta liikennöidään konesalin ulkopuolelle.

Ranka

Tämän arkkitehtuurin ranka voidaan mieltää hierarkkisen mallin runkoverkoksi. Hierarkkisessa arkkitehtuurissa runkoverkko koostui muutamista järeistä ristiin kytketyistä kytkimistä tai reitittimistä kun taas ranka koostuu monista korkean porttimäärän ja suuren kytkentäkapasiteetin omaavista kytkimistä. Rangan laitteita ei myöskään ole kytketty toisiinsa vaan liikennöinti tapahtuu aina yhdellä rankahypyllä lehdestä toiseen lehteen.

2.2.4 Lehti-ranka suunnittelumallin edut

Perinteinen hierarkkinen suunnittelumalli on edelleen varsin toimiva kunnes eri toiminnallisten kerrosten väliset linkit alkavat ylikuormittua. Perinteisissä lähiverkoissa tämä on melko harvinaista ja ainakin lähivuosina lehti-ranka-mallia nähtäneen lähinnä suuremmissa konesaleissa, jossa siitä todella on hyötyä. Esimerkiksi Cisco kertoo vuotuisessa raportissaan että keskimääräinen konesalien sisäinen liikenne kattoi jopa 76 %

koko konesalin liikenteestä vuonna 2012 ja sen uskotaan vain kasvavan vuoteen 2017 mennessä. Käyttäjäliikenteen jäädessä vuonna 2012 keskimäärin 17 %:iin konesaliliikenteestä voidaan varsinkin isompien konesalien hyödyt lehti-ranka-mallille nähdä selvästi [18].

2.3 Pilvipalveluiden määritelmiä

Pilvipalvelut ovat tulleet lähivuosina käyttöön laajalti niin pieniin kuin suuriinkin yrityksiin. Yritykset ja erilaiset organisaatiot ostavat pilvipalveluita käytännössä kustannustehokkuuden ja ylläpidon helppouden takia. Pilvipalveluiden määritelmä on termi pilveä lukuun ottamatta edelleen hyvin häilyvä. Yksinkertaisimmillaan pilvipalvelulla tarkoitetaan Internet-verkossa tapahtuvaa tiedostojen jakoa ja varastointia jonkin ulkopuolisen tahon omistamalla palvelimella. Työskentely ei tapahdu enää ainakaan puhtaasti omalla koneella vaan palvelu tai sen osa siirretty suoritettavaksi pilvipalveluun. Pilvipalvelulle on tyypillistä, että pilvi-infrastruktuurin rakenteet kuten fyysiset palvelimet ja verkot, joissa laskenta tapahtuu, häivytetään käyttäjän näkyvistä. Palveluntarjoaja voi siirtää virtuaalikoneita tai kokonaisia virtuaalisia verkkoja tarpeidensa mukaan palvelinkeskusten välillä tämän näkymättä pilvipalvelun käyttäjälle mitenkään. Käyttäjän kannalta pilvipalvelu on siis palvelumalli, jossa jotakin palvelunosaa tarjotaan käyttäjälle maksua vastaan. Tyypillisiä piirteitä pilvipalvelulle ovat itsepalvelu, massahyödykehinnoittelu sekä läpinäkyvä skaalautuvuus [3].

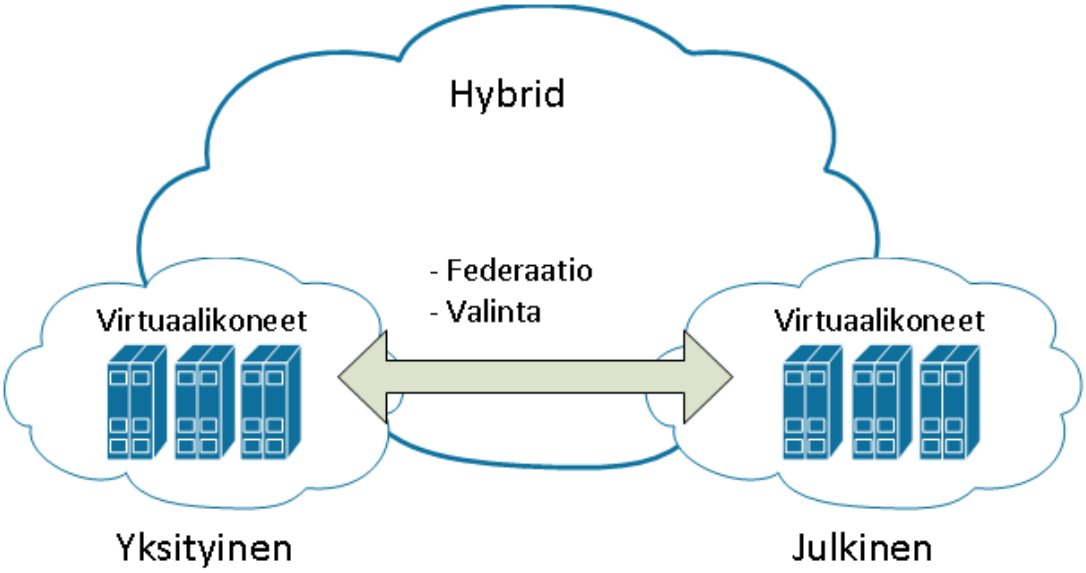
Itsepalvelulla tarkoitetaan pilvipalveluiden tapauksessa sitä että käyttäjän on kyettävä itse luomaan tilaamaansa palveluun resursseja, käyttäjätilejä pilvipalveluntarjoajan käyttöliittymän kautta. Yleensä tämä käyttöliittymä on toteutettu web-pohjaisena sovelluksena.

Massahyödykehinnoittelulla tarkoitetaan sitä että palvelu toimitetaan palvelusopimuksen mukaisin ehdoin ja käyttö tapahtuu itsepalveluna. Jos palveluntarjoamat ehdot tai palvelun vaihtoehdot tyydytä voi palveluntarjoajan aina vaihtaa toiseen.

Läpinäkyvällä skaalautuvuudella tarkoitetaan pilvipalvelujen tapauksessa sitä että palvelusta maksetaan käytön mukaan. Esimerkiksi hinnoittelun perustuessa maksimi käyttäjämäärä voidaan palveluntarjoajan käyttöliittymän kautta ostaa tarvittaessa lisää käyttäjiä, jolloin myös hinta nousee. Näin pilvipalvelut mahdollistavat hyvän skaalautuvuuden yrityksen tarpeeseen eikä yritys maksa niin sanotusti turhasta.

2.3.1 Erilaiset pilvipalveluiden jakelumallit

Julkisella pilvellä (engl. public cloud) tarkoitetaan pilvimallia, jossa yleensä laskentakapasiteettia tai palveluja ostetaan ja hyödynnetään julkisen internetin välityksellä. Tässä mallissa ostettava palvelun, infrastruktuurin tai kapasiteetin ylläpito on kokonaan luovutettavissa kokonaan pilvitarjoajan vastuulle, jolloin palvelun ostaja voi keskittyä paremmin omaan ydinliiketoimintaansa. Julkinen pilvipalvelu on kuvattu muiden jakelumallien kanssa kuvassa 6 [14][29].



Kuva 6: Erilaiset pilvipalveluiden jakelumallit [29].

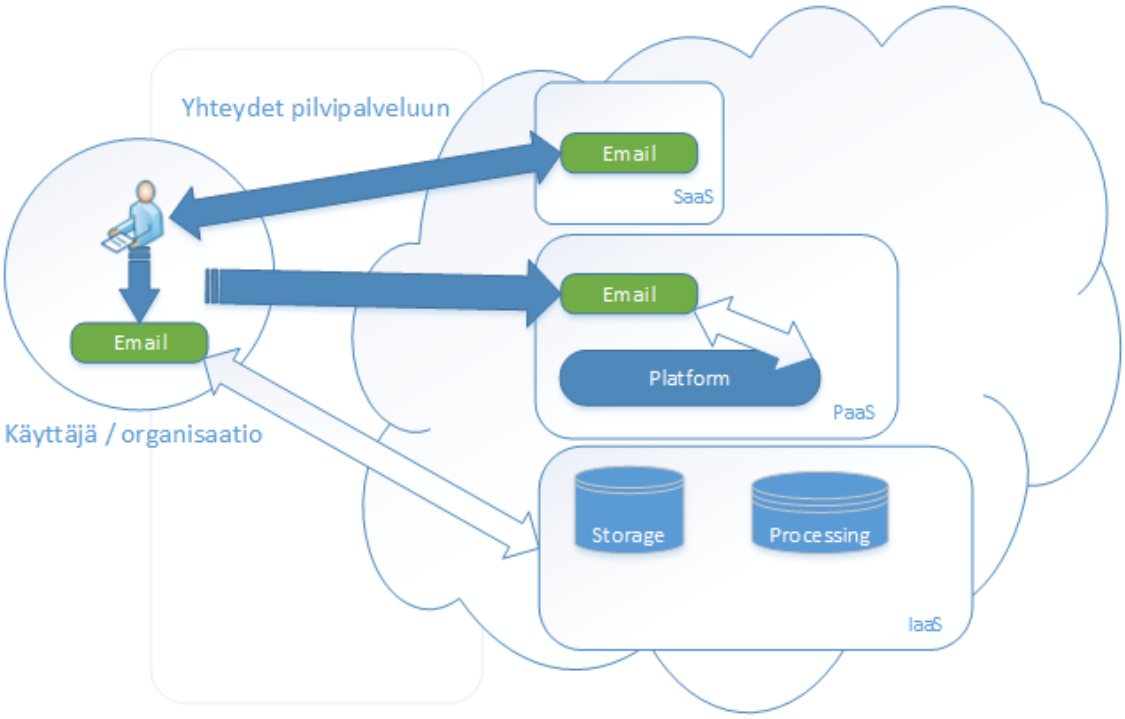
Yksityisellä pilvellä (engl. private cloud) tarkoitetaan pilvimallia, joka tarjoaa laskentakapasiteettia tai palveluita rajalliselle määrälle käyttäjiä. Kuvan 6 mukaisesti yksityinen pilvipalvelu sijaitsee yrityksen omassa konesalissa. Muita nimityksiä tälle mallille ovat esimerkiksi sisäinen pilvi tai yrityspilvi. Hallinnosta ja ylläpidosta tässä mallissa vastaa palvelua käyttävän organisaation oma ylläpito. Yksityisessä pilvimallissa pilvi-infrastruktuuri ja täten myös data sijaitsee organisaation omassa konesalissa (engl. datacenter), joten tietoturvan tason voidaan ajatella olevan parempi tai ainakin paremmin tiedossa eikä ulkoiselle toimijalle luovuteta näin liikaa vastuuta. Perinteiseen organisaatioon omaan vahvasti virtualisoituun konesaliin verrattaessa erona on se että konesalin hallinta siirtyy enemmän laitteistokerrokselta virtualisointikerrokselle [14][29].

Hybridi-pilvellä (engl. hybrid cloud) tarkoitetaan pilvimallia, joka koostuu vähintään yhdestä yksityisestä ja julkisesta pilvestä. Esimerkiksi yksityinen pilvipalvelun toimittaja muodostaa kumppanuussuhteen jonkin julkisen pilvipalvelun tarjoajan kanssa. Esimerkkinä voidaan miettiä kuvan 6 mukaista tapausta, jossa organisaation omissa tiloissa eli käytännössä konesalissa sijaitsee yksityinen pilvipalvelu. Organisaation oma yksityisessä pilvessä ajetaan luottamuksellista reaaliaikadataa. Vuosien saatossa arkistoitavaa dataa kertyy paljon, jonka ei lasketa olevan enää niin bisneskriittistä. Organisaatio ostaa julkiselta pilvitarjoajalta yksityisen pilvensä jatkoksi julkista pilveä ja näiden kahden erillisen palvelun välille muodostetaan luottamussuhde eli federaatio, jonka jälkeen voidaan puhua hybridimallisesta pilvestä. Tätä hybridiä voidaan käyttää esimerkiksi siten että tieto luokitellaan tarkasti ja tämän perusteella bisneskriittinen ja luottamuksellinen reaaliaikadata säilytetään edelleen organisaatioin omassa konesalissa, mutta ei bisneskriittinen arkistoitava data tallennetaan julkiseen pilvipalveluun [14][29].

2.3.2 Pilvipalveluiden palvelumallit

Pilvipalveluihin on muodostunut erilaisia palvelua malleja, jotka voidaan lukea pilvipalveluiden alle. Tässä alaluvussa käydään läpi erilaiset palvelumallit, joista ensimmäisenä käsitellään SaaS eli Software as a Service-malli.

SaaS eli ohjelmistoja palveluna on vuosituhannen vaihteessa syntynyt malli sovellusten myyntiin ja levitykseen. Tyypillisesti SaaS-sovelluksia käytetään selaimella internetin yli hinnoittelun perustuessa kuukausimaksuun ja käyttäjämäärään. Määritelmän mukaan SaaS:ssa tuotteen omistajuus on palveluntarjoajalla, joka huolehtii asennus-, ylläpito- ja huoltotoimista. Asiakas pystyy näin keskittymään vain palvelun käyttämiseen. Palvelumalleista SaaS sopii yleensä parhaiten vähemmän teknisille käyttäjille. Esimerkiksi Google Docs on SaaS-palvelu. Kuvassa 7 on kuvattu kuinka tämän kappaleen SaaS-palvelumalli sekä myöhemmin mainittivat PaaS- ja IaaS-palvelumallit järjestyvät käyttäjän näkökulmasta. Käyttäjällä tarkoitetaan tässä tapauksessa pilvipalvelun asiakasta [5][15].



Kuva 7: SaaS-, PaaS-, IaaS-palvelumallit käyttäjän näkökulmasta

PaaS (engl. Platform as a Service) poikkeaa SaaS-mallista siten että tuotteiden tai tarkemmin ohjelmien omistaja on asiakas, joka käyttää palveluntarjoajan alustaa oman ohjelmistonsa ajamiseen. Käyttäjän kannalta PaaS-palvelumalli on kuvattu kuvassa Kuva 7. PaaS on käyttäjälle SaaS:ia monimutkaisempi hallittava, mutta se tarjoaa loistavat mahdollisuudet esimerkiksi ohjelmistokehittäjille ottaa käyttöön testata, päivittää ja ylläpitää omia ohjelmia ilman ylläpidollisia päänsärkyjä. PaaS-alusta sisältää yleensä tarvittavat työkalut ja ympäristöt esimerkiksi web- tai mobiiliapplikaatioiden käyttöönottoon. Yleensä PaaS:n käyttö vaatii kuitenkin vähintään jollain tasolla ohjelmisto-osaamista, jotta PaaS:n päälle saadaan rakennettua ja säädettyä kaikki toimimaan.

Parhaimmillaan PaaS tarjoaa ohjelmistokehittäjille mahdollisuuden keskittyä siihen missä he ovat parhaita eli kirjoittaa hyvää koodia. Skaalautuvuus ja resurssiongelmat ulkoistuvat palveluntarjoajalle. Esimerkiksi Microsoftin Azure sekä Heroku ovat PaaS-palveluja [5][6].

NaaS (engl. Network as a Service) eli verkko palveluna on palvelumalli, jossa asiakkaalle toimitetaan verkko palveluna. Todellisuudessa NaaS-palvelumallissa on kysymys enemmänkin yhtenäisestä palvelun hallinnasta, jolloin prosessointi- ja verkkolaitteistoa ei nähdä erillisinä vaan niitä voidaan hallita tehokkaasti yhtenäisen alustan avulla. Tämän seurauksena verkkoinfrastruktuuri muuttuu täysin läpinäkyväksi käyttäjän näkökulmasta. Suurimpina etuina voidaan pitää palvelun joustavuutta sekä ylläpitokulujen kutistumista. Asiakkaan ei tarvitse huolehtia esimerkiksi verkon VLAN-konfiguroinneista tai palomuureista erikseen vaan tekniset määritykset näille tehdään automaattisesti, kun esimerkiksi lisäresurssiksi tarkoitettu palvelin lisätään verkkoon. Varsinkin uusille yrityksille NaaS tarjoaa erittäin kustannustehokkaan palvelumallin, sillä verkkolaitteistoa ei erikseen tarvitse ostaa, vaan käyttäjän näkökulmasta riittää, että työasema on yhteydessä internettiin [15].

IaaS (engl. Infrastructure as a Service) eli infrastruktuuri palveluna on palvelumalli, jossa koko infrastruktuuri verkkoineen ja palveluineen siirretään virtualisoituun ympäristöön. IaaS-palvelun käyttäjän kannalta IaaS näkyy kuin mikä tahansa verkko palvelimineen. Parhaimmillaan uusia palvelimia sekä verkkoja kuin myös verkkoihin palvelimien liittämisiä voidaan suorittaa suoraan IaaS-palveluntarjoajan käyttäjälle tarjoamasta käyttöliittymästä käsin. Kuvassa 7 kuvataan kuinka IaaS-palvelumallin käyttäjä saa maksamalleen summalle vastinetta prosessointikapasiteettina eli muistina ja prosessoritehona sekä säilytystilaa levytilana [5].

Edellä mainitut SaaS, PaaS ja IaaS ovat yleisimmät käytössä olevat pilvipalveluiden palvelumallit. Näiden lisäksi on olemassa paljon muitakin palvelumalleja tai akronyymejä ja radikaaleimpana puhutaan XaaS:sta (anything as a Service), joka tarkoittaa jonkunlaista hybriditoimitusta joka sisältää sekä SaaS, PaaS tai IaaS elementtejä sekä mahdollisesti myös muita kuten CaaS (Communication as a Service), jolla tarkoitetaan yrityksen sisäisten ja ulkoisten viestintäpalveluiden kuten pikaviestimien ja VoIP-sovellusten ostamista palveluna.

2.4 Virtualisointi

Pilvipalveluille yleisiä ominaispiirteitä ovat resurssien tehokas dynaaminen käyttö ja elastisuus sekä mahdollisuus maksaa enemmän tai vähemmän palvelusta riippuen prosessointitehon sekä säilytyskapasiteetin tarpeen mukaan. Teknisesti tarkasteltaessa tämän on mahdollistanut virtualisointiteknologioiden yleistyminen sekä verkkoyhteyksien nopeuksien kasvu.

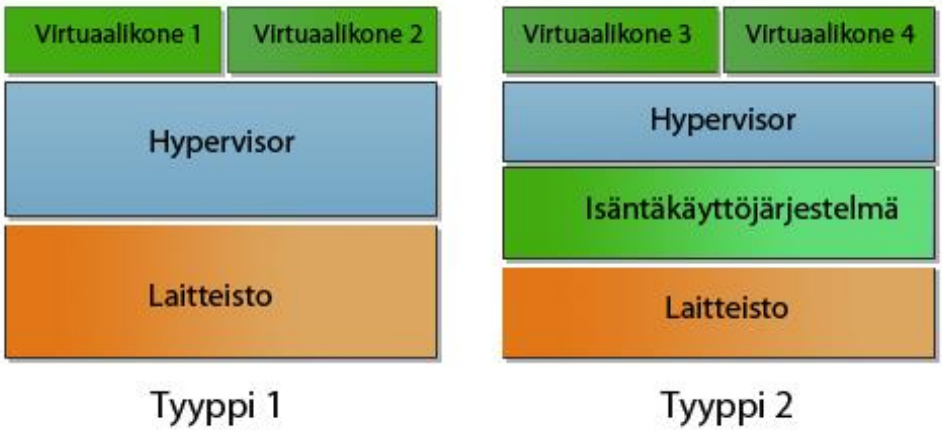
Virtualisoinnille on monia syitä kuten resurssien jakaminen usean käyttäjän tai asiakkaan kesken tehokkaasti, verkkojen tai palveluiden eristäminen toisistaan, resurs-

sien yhdistäminen, palveluiden ja resurssitarpeen dynaamiset muutokset ja virtualisoidun ympäristön hallinnan helppous.

Varsinaisesti virtualisoinnissa ei ole mitään uutta ja ihmeellistä, mutta sen käyttötavat ovat lähivuosina monipuolistuneet ja periaatteessa mitä tahansa voidaan nykyään virtualisoida tehokkaasti. Tässä luvussa esitellään kevyesti millaisia erilaisia virtualisointimahdollisuuksia nykyään on ja kuinka näitä hyödynnetään käytännössä. Loppukäyttäjille virtualisointi ei varsinaisesti näy mitenkään paitsi yleensä pienempinä kuukausimaksuina palvelusta.

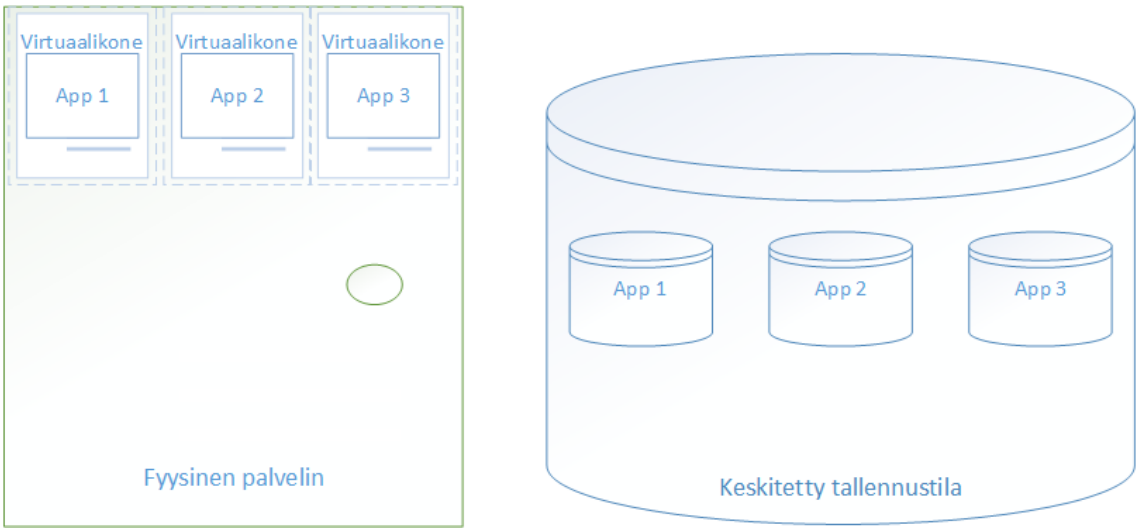
2.4.1 Palvelinvirtualisointi

Palvelinvirtualisoinnissa fyysiset palvelinresurssit kuten yksittäiset prosessorit, muistit ja kokonaiset palvelinraudat piiloutuvat virtuaalisten palvelinten alle. Virtualikoneita ajavaa palvelinalustaa kutsutaan hypervisoriksi, joka onkin niin palvelinvirtualisoinnin kuin monen muunkin virtualisoinnin ydin. Hypervisorit jaetaan tyypin 1 ja tyypin 2 hypervisoreihin. Tyypin 1 hypervisor on ajossa suoraan fyysisellä palvelimella tai laitteistolla ja tyypin 2 hypervisoria ajetaan erillisen käyttöjärjestelmän päällä. Hypervisor tyyppien ero on kuvattu kuvassa 8 [30].



Kuva 8: Erilaiset hypervisor tyypit [30].

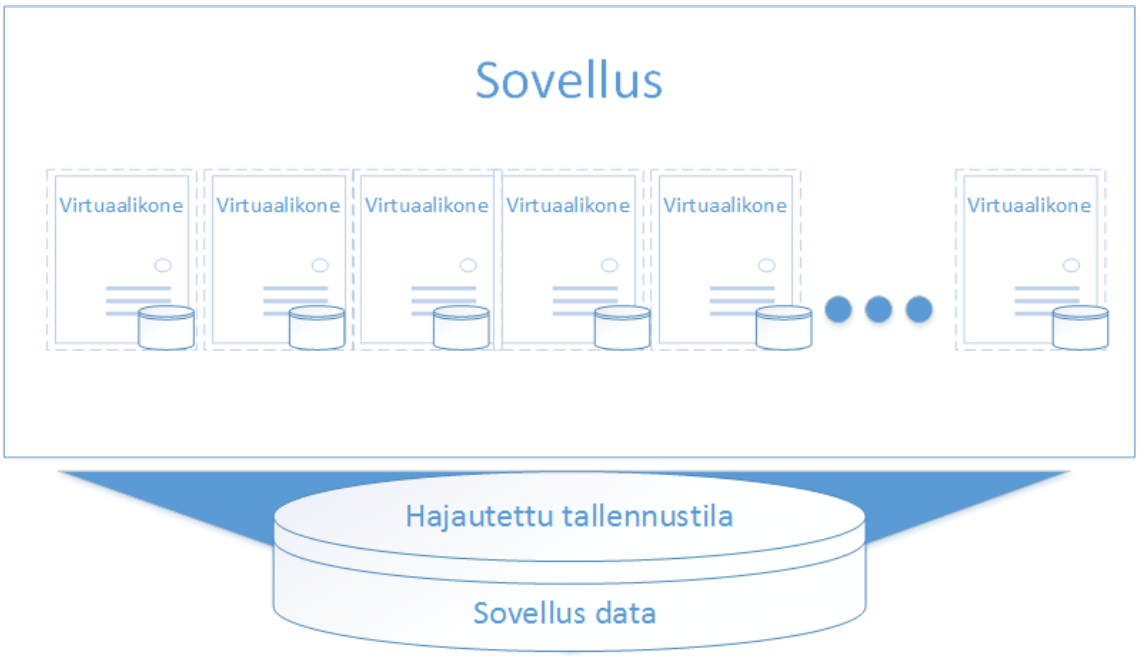
Fyysisillä palvelimilla siis voidaan ajaa useita kevyempiä virtuaalisia palvelimia, jolloin fyysiset resurssit saadaan usein huomattavasti paremmin käyttöön. Mikään ei estä ajamasta myös hypervisoreita virtualisoituina, mutta hahmottamisen helpottamiksi oletamme että hypervisoria ajetaan suoraan fyysisellä palvelimella eli on hypervisor tyyppiä 1. Tällaista mallia pidetään klassisena palvelinvirtualisointina (engl. classic virtualization), joka on esitetty kuvassa 9 [16].



Kuva 9: Klassinen palvelinvirtualisointi [16].

Kuvan 9 mukaisesti konesalissa on useita fyysisiä palvelimia, joiden sisällä ajetaan useita virtualisoituja palvelimia, jotka ajavat sisässään sovelluksia. Lisäksi kuvassa 9 on mukana keskitetty levytila, jota niin virtuaalikoneet kuin myös sovellukset käyttävät [16].

Erityisesti massiivisiksi sovelluksiksi paisuneet Google sekä sosiaalisen median palvelut kuten Facebook ja Twitter ovat luoneet toisenlaisen mallin, jota nimitetään käänteiseksi virtualisoinniksi (engl. inverse virtualization). Toisin kuin perinteisessä virtualisointimallissa, jossa virtuaalipalvelin luodaan jokaista sovellusta kohden, tässä mallissa yhtä sovellusta kohden luodaan monta palvelinta. Palvelinvirtualisointi siis tapahtuu sovelluskerroksella, jolloin yksittäinen palvelin ajaa vain pientä osaa sovelluksesta. Käänteinen virtualisointi malli on esitetty kuvassa 10 [16].



Kuva 10: Käänteinen palvelinvirtualisointi [16].

Tässä mallissa on myös ominaista että dataa ei tallenneta niin kuin klassisessa mallissa eli suureen ja tehokkaaseen keskitettyyn tallennustilaan vaan jokainen sovelluksen virtualisoinnissa oleva palvelin sisältää oman pienen tallennustilansa. Jokainen palvelin siis sisältää pienen osan jättimäistä hajautettua tallennustilapoolia. Hyviä esimerkkejä tällaisesta tallennuksesta ovat esimerkiksi OpenStackin Swift, HDFS (Hadoop Distributed File System) sekä GFS (Google File System) [16].

Virtualisoitua ympäristöä kuvataan monesti fyysisten ja virtuaalisten palvelinten suhteella, josta käytetään monesti p2v-lukua (engl. physical to virtual ratio). Kyseisellä luvulla kuvataan kuinka monta virtuaalista palvelinta ympäristössä keskimäärin on yhtä fyysistä palvelinta kohden. Etuina palvelinvirtualisoinnissa ovat esimerkiksi seuraavat asiat: samalla fyysisellä koneella voidaan ajaa useita eri käyttöjärjestelmiä. Nämä eri käyttöjärjestelmät voidaan myös sammuttaa tai uudelleen käynnistää toisistaan riippumatta. Virtualikoneille voidaan myös eristää järjestelmiä, jolloin niiden siirto niin kuin yleensäkkin virtualikoneiden siirto toiselle fyysiselle palvelimelle on helppoa.

Palvelinvirtualisoinnissa täytyy kuitenkin muistaa se tosiasia että virtualisointi ei synnytä tyhjästä uusia resursseja vaan kovassa rasituksessa fyysisen koneen resurssien loppuessa myös kaikkien virtualisoitujen palvelinten resurssit loppuvat. Lisäksi jos hypervisorilla ajettaville virtuaalikoneille on annettu käytännössä rajattomat resurssit käyttöönsä voi yksittäinen suuressa rasituksesa oleva virtuaalikone hyydyttää koko isäntäkoneen virtuaalikoneet. Myös varsinainen virtualisointialusta, jonka päällä virtuaalikoneita ajetaan, kuten esimerkiksi VMWaren vSphere tai avoimen lähdekoodin Virtualbox käyttävät osansa fyysisen koneen resursseista. P2V-luvun seuraaminen ja virtualisoidun ympäristön valvonta ja optimointi on tärkeässä roolissa riittävän hyvän palvelulaadun takaamiseksi.

2.4.2 Työpöytävirtualisointi

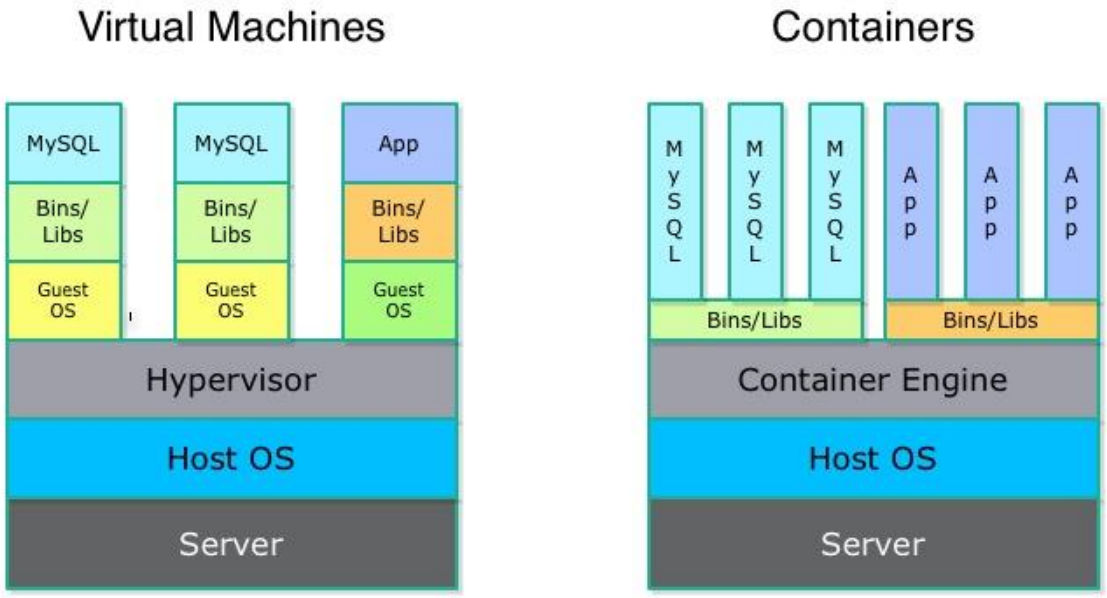
Työpöytävirtualisointi on tapa tarjota työaseman ohjelmat tai koko käyttöjärjestelmä samaan tapaan virtualisoituna kuin palvelinvirtualisoinnissa. Virtuaaliset työpöytien käyttöjärjestelmät tai ohjelmat ajetaan hypervisorin päällä. Käytännössä siis työaseman ohjelmat ajetaan keskitetyllä palvelimella oman työaseman sijaan. Tämän virtualisoinnin etuina voidaan pitää ainakin helpompaa hallintaa ylläpidolle, nopea toipuminen esimerkiksi päätelaitteen häviämisen tai rikkoontumisen takia, parempi tietoturva sekä mahdolliset kustannussäästöt.

Työpöytä ei enää ole sidottu enää yhteen päätelaitteeseen vaan mahdollisesti kaikkia samoja toimintoja kuin perinteisellä työasemalla voidaan nyt käyttää mobiilisti kevyt clientilla (engl. thin client), kannettavalla tietokoneella, pöytäkoneella sekä tabletilla tai älypuhelimella [17].

2.4.3 Sovelluskonttivirtualisointi

Sovelluskontti virtualisoinnilla (engl. application container virtualization) tarkoitetaan tapaa, jolla yhdellä Linux-kernelillä ajetaan useita eristettyjä sovellusinstansseja. Sovel-

lusinstanssit on paketoitu omiin sovelluskontteihinsa kaikkine riippuvuuksineen kuten sovelluskomponentteineen ja kirjastoineen. Sovelluskontit ovat siis toisistaan eristettyjä ja niitä voidaan ajaa useita yhdellä koneella samaan tapaan kuin hypervisor ajaa virtuaalikoneita perinteisessä palvelinvirtualisoinnissa. Sovelluskontit kuitenkin jakavat saman kernelin tai mahdollisesti joitakin muitakin Linuxin ytimen palveluista keskenään, jolloin usean sovelluskontin ajaminen samalla laitteistolla on kevyempää kuin vastaavan määrän erillisiä virtuaalikoneita. Heikkoutena tässä tavassa on kuitenkin se että jokaisessa sovelluskontissa on ajettava samaa käyttöjärjestelmää ja kerneliä kuin mitä alla oleva isäntäpalvelin tarjoaa. Sovelluskontin käynnistys ei siis käynnistä erillistä käyttöjärjestelmää vaan se lataa kyseisen sovelluksen kaikkine tarvittavine osineen eristettyyn ympäristöönsä. Kuvassa 11 on esitetty periaatteellinen ero sovelluskonttivirtualisoinnin ja perinteisen palvelinvirtualisoinnin välillä [38].



Kuva 11: Virtuaalikone ja sovelluskontti [38].

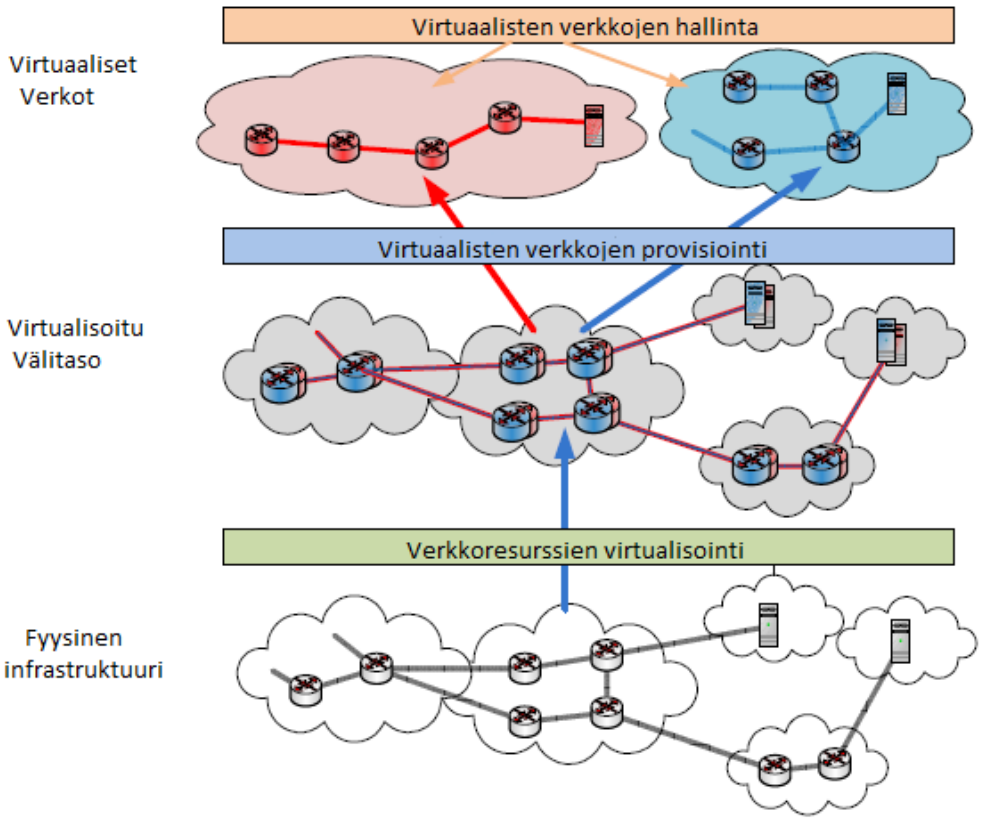
Kuvasta 11 näkyy selvästi erillisen hypervisorin ja sovelluskonttien ajamiseen tarvittavan konttimoottorin (engl. Container Engine) ero. Sovelluskontti ajetaan käyttöjärjestelmän sisällä ja se sisältää vain paketoidun sovelluksen riippuvuudet kun taas perinteisempi hypervisor ajaa ohjelmistoa, joka emuloi laitteistoa hypervisorin päällä ajettavalle käyttöjärjestelmille [38].

Tällä hetkellä eniten kiinnostusta herättävä sovelluskonttivirtualisoinnin sovellus on nimeltään Docker. Käytännössä Docker sovelluskonttijärjestelmä koodille ja sillä voidaan hallita ohjelmistoja konttien sisällä. Itse kontit käyttävät LXC-tekniikkaa, joka hyödyntää Linux kernelin nimiavaruutta ja Cgroupseja eristäessään kontit toisistaan. Hyötyinä tälle tekniikalle erityisesti ohjelmistoyritysten kannalta on se että ohjelmistokehittäjä voi paketoida konttiinsa koko ohjelmistonsa riippuvuuksineen, konfiguroida asetukset mieleisekseen, testata kontin kotiorganisaatiossaan tai omalla koneellaan ja tämän jälkeen siirtää ohjelmistonsa sovelluskonttivirtualisointia tarjoavalle pilvipalveluntarjoajalle ajoon. Sovelluskontti tarvitsee pilvipalvelulta vähemmän muistia ja levyti-

laa kuin kokonainen virtuaalikone, joten sen käyttö on useassa tapauksessa halvempaa. Lisäksi uusien sovelluskonttien käynnistäminen on todella nopeaa, joten ohjelmiston suosion kasvaessa tai laskiessa järjestelmän skaalaus ylös- tai alaspäin on nopeaa [38].

2.4.4 Verkkovirtualisointi

Verkkovirtualisoinnin tavoitteena on ottaa nykyiset verkon palvelut, ominaisuudet ja asetukset, joita tarvitaan virtuaalisten verkkojen provisioinnissa kuten VLANit, VRF:t (engl. Virtual Routing and Forwarding), palomuurit säännöt ja reitityssäännöt ja erottaa ne fyysisestä verkosta, jonka päälle verkko virtualisoidaan. Tämä virtualisoitu ohjelmallinen verkkokerros on tämän jälkeen täysin automatisoitavissa erilaisiin tarkoituksiin. Käytännössä verkkovirtualisoinnin tarkoitus on tehdä verkoille sama minkä palvelinvirtualisointi teki palvelimille. Fyysisen verkon tehtäväksi jää tarjota korkean saatavuuden IP-pohjainen siirtoverkko ja virtualisointikerroksella hoidetaan muut tehtävät tavallisista siirtokerroksen kytkemistehtävistä vaativiin reititys- ja tietoturvaominaisuuksiin asti. Virtualisoidun verkkoympäristön perus arkkitehtuuri on esitetty kuvassa 12 [19][21].



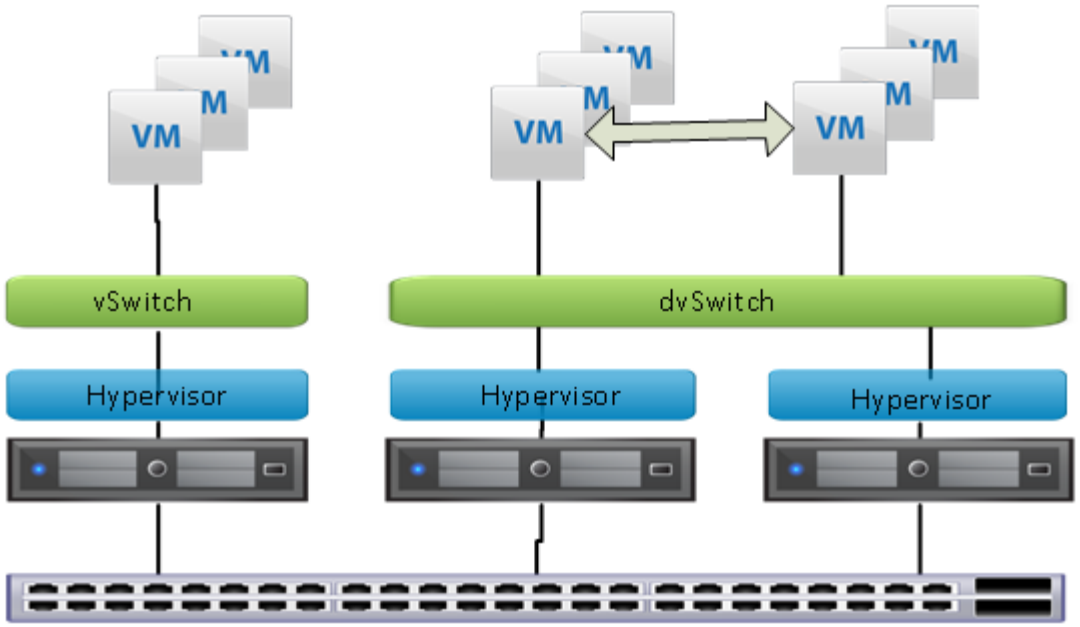
Kuva 12: Virtualisoidun verkon perusarkkitehtuuri [21].

Kuvan 12 mukaisesti fyysinen verkkokerros on segmentoitu kahdelle erilliselle virtuaaliselle verkolle. Nämä kaksi loogisesti täysin erillistä verkkoja käyttävät samoja verkon fyysisiä laitteita.

Verkkovirtualisoinnin hyötyjä kuvataksemme hyvänä esimerkkinä toimii ongelma, jossa virtuaalikoneita siirretään fyysiseltä koneelta toiselle. Fyysisen koneen muuttuessa myös virtuaalikoneen fyysinen yhteys ulkomaailmaan muuttuu. Käytännös-

sä fyysinen verkkokortti eli eli pNic (engl. physical nic) vaihtuu toiseksi ja verkkokaapeleita joudutaan mahdollisesti kytkemään uusiksi. Tämä ongelma on verkkovirtualisoinnilla käytännössä pystytty ratkaisemaan. Kun virtuaalikone siirretään toiseen aliverkkoon, sen IP-osoite täytyy muuttaa, sillä IP-osoite toimii virtuaalikoneelle sekä paikantimena (engl. locator) kuin myös tunnistimena (engl. identifier). Virtuaalikone on siis paljon helpompaa siirtää saman aliverkon sisällä kuin aliverkosta toiseen.

Tavallisesti jokaisella virtuaalikoneella on yksi tai useampi virtuaalinen verkkokortti, jotka liittyvät hypervisorin ajamaan virtuaalisen kytkimeen eli vSwitchiin (engl. virtual switch). VSwitch voidaan liittää fyysiseen verkkokorttiin ja näin saada verkkoyhteys ulospäin muihin verkkoihin. Yksittäinen hypervisorin sisässä ajettava vSwitch on esitetty kuvan 13 vasemmassa reunassa.



Kuva 13: Vswitch ja dVswitch yhdistettynä samaan fyysiseen reitittimeen [31].

Käytännössä tämä vswitch voidaan levittää useampien eri fyysisten koneiden kesken, jolloin puhutaan dvswitchistä (engl. distributed virtual switch). Kahden hypervisorin kesken jaettu dvswitch on esitetty kuvan 13 oikeassa reunassa. Tässä tapauksessa eri fyysiset koneet voivat toimia samassa virtualisoidussa aliverkossa alla olevata fyysisestä verkosta riippumatta. Tällaisia tuotteita on tarjolla esimerkiksi yrityksillä kuten VMWare, mutta myös avoimempia ratkaisuja on paljon kehitteillä kuten OpenvSwitch ja laajempana tekniikkana OpenFlow.

OpenFlow

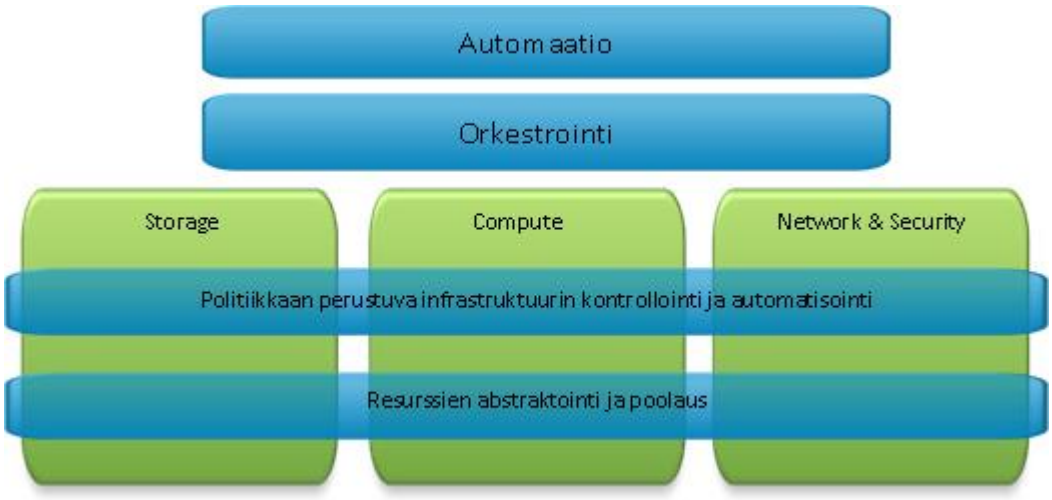
OpenFlow on avoin verkkotekniikka, joka sisältää sekä ohjelmarajapinnan (engl. application programming interface) että protokollan. Ohjelmarajapinnan kannalta miettien OpenFlow:lla voidaan ohjelmoida verkkolaitteiden, kuten kytkinten ja reitittimien, sisäisiä kytkentätauluja. Protokolla mielessä OpenFlow muodostaa kontrollereineen ja ohjelmistoineen täysin ohjelmoitavan ja virtualisoitavan verkkoalustan. OpenFlow on

tällä hetkellä tarjolla lisäosana esimerkiksi HP:n, NEC:n, Ciscon ja Juniperin kytkimiin, ohjelmallisiin kytkimiin kuten Niciran alun perin kehittämä OpenVSwitchiin sekä Linux pohjaisiin kotireitittimiin asentuvaan OpenWRT:hen [8].

OpenFlow:n mallissa verkkolaitteiden ohjelmoitavuus keskitetään ohjauskerrokselle, jota hallitsee yksittäinen verkkokontrolleri. Kontrolleri tekee kaikki kytkentä ja prosessointipäätökset kun verkon kytkimet itsessään toimivat vain näiden päätösten toimeenpanijoina ja kytkevät paketteja kontrollerin päätösten mukaan. Uuden pakettivirran saapuessa kytkin ohjaa pakettivirran kontrollerille, joka puolestaan ohjeistaa kytkimen tai kytkimet verkkopolulla kytkemään flow:n eli uudet paketit ja sitä seuraavat paketit haluamallaan tavalla. OpenFlow itsessään määrittää protokollan kuinka kontrolleri tai kytkimet kommunikoivat. Yleisimmät nykyiset OpenFlow:n käyttätapaukset ovat pääsyn hallinnassa ja konesaleissa [8].

2.4.5 Software Defined Data Center

Software Defined Data Center eli SDDC on konesali konsepti, jossa kaikki infrastruktuuri on palvelimia, verkkoja sekä tallennustilaa myöten virtualisoitu ja se toimitetaan palveluna. Ohjauskerros (engl. control plane) on täysin automatisoitu ohjelmallisesti. Käytännössä SDDC on virtualisoinnin ja pilvipalveluiden kannalta seuraava askel, sillä se mahdollistaa niin sanotun ITaaS (IT as a Service) tuottamisen asiakkaille. Kuvassa 14 on esitetty yksinkertaistettu SDDC:n ylätason toteutus [9].



Kuva 14: Yksinkertaistettu SDDC toteutus [9].

Kuten kuvasta 14 selviää SDDC:n aikaansaamiseksi levytila, laskenta eli palvelimet ja verkko on täytynyt kokonaan virtuaalisoida, jotta niitä voidaan tehokkaasti ohjata ohjelmallisesti. Erilliset resurssit abstraktoidaan resurssipooleihin, jolloin niitä voidaan kontrolloida luotujen politiikkojen perusteella. Politiikat saatetaan käyttöön automatisoinnin avulla. Käyttäjän näkökulmasta käyttäjä valitsee itsepalvelukatalogista halutun palvelun ja automaatio hoitaa loput. Esimerkiksi palvelimen tapauksessa tämä tarkoittaisi sitä että kuvan 14 mukaiselle compute-klusterille luodaan palvelin, jolle varataan

levytila storage-klusterista. Samalla verkko- ja palomuuriasetukset asetetaan sopiviksi automaattisesti [9].

SDDC vaatii siis käytännössä kaikkien osien virtualisointi, joka ei ole vielä mahdollista kaikille organisaatioille. Varsinkin verkkojen virtualisointi on tällä hetkellä vielä kehitysasteella, mutta esimerkiksi avoimen lähdekoodin pilvipalvelualusta OpenStackin verkkovirtualisointiosan eli Neutronin kehitys on ollut lupaavaa. SDDC:n kehitys on kuitenkin kiinnostavaa seurattavaa, koska se pakottaa perinteisten organisaatioiden IT:n miettimään täysin uusiksi omia prosessejaan ja tekemistään.

3 TIETOTURVA JA VALVONTA OHJELMISTO-YRITYKSISSÄ

Perinteisesti datan, järjestelmien tai verkkojen turvaaminen ei ole ollut suurella prioriteetilla pienissä tai edes keskisuurissa yrityksissä. Lähivuosien kehitys on kuitenkin johtanut siihen että näiden organisaatioiden asiakkaina olevat suuremmat yritykset tai organisaatiot ovat jo pidempään panostaneet tietoturvaan. Tämän seurauksena myös alihankkijoille on tullut entistä kovempia vaatimuksia tietoturvan osalta. Tietoturvaan on siis ollut pakko alkaa keskittämään enemmän resursseja myös Pk-yrityksissä [7].

Ohjelmistoyritys eroaa siinä mielessä monista pk-yrityksistä, että sen oikea omaisuus piilee tekijöissä ja siinä tiedossa mitä tekijänsä tuottavat eli käytännössä ohjelmistosuunnittelijoiden tuottamassa ohjelmistokoodissa ja organisaation luomissa asiakassuhteissa. Kaupallisessa mielessä hyvin hoidettu tietoturva suojelee organisaation mainetta niin asiakkaiden kuin potentiaalisten uusien asiakkaidenkin hankinnassa ja säilyttämisessä. Näiden faktojen valossa voimme todeta että tietoturvasta tinkiminen aiheuttaa suuren riskin ohjelmistoyritykselle ja riskin toteutuessa voi johtaa jopa hyvinkin nopeaan konkurssiin. Tietoturvan toteuttamista pohtiessa täytyy kuitenkin muistaa että tietoturvan taso on aina kompromissi käytettävyyden ja tietoturvapolitiikan tiukkuuden välillä. Suurten organisaatioiden mielestä pienempiä ohjelmistoyrityksiä käytetään, koska ne ovat ketteriä ja joustavia. Tätä pienempien ohjelmistoyritysten suurinta vahvuutta ei haluta tietoturvapolitiikalla rampauttaa. Täydellistä tietoturvaa ei ole olemassakaan, joten tällaista ohjelmistoyrityksissä on turha lähteä hakemaan. Hyvä tietoturva voidaan kuitenkin toteuttaa toimivasti ja esimerkiksi keskittämällä pääsynhallinta saadaan helpotettua IT-organisaation toimintaa

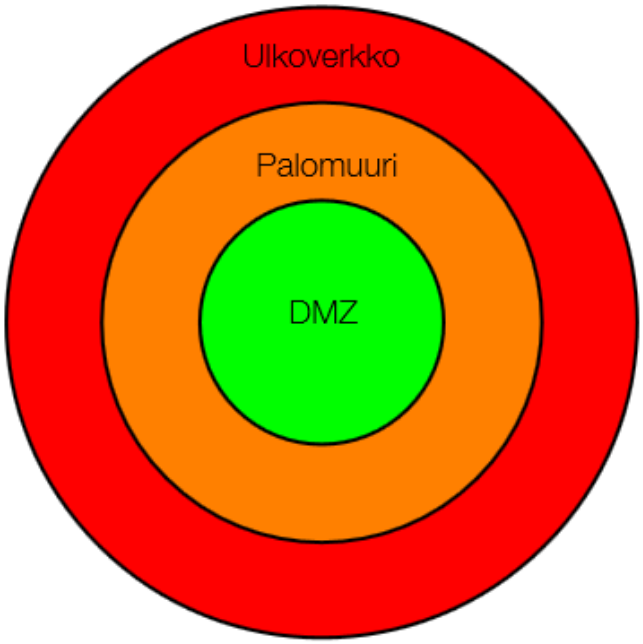
Tämän luvun alaluvuissa käydään tarkemmin läpi tämän diplomityön rajauksen mukaisesti verkon tietoturvaa, identiteetin- ja pääsynhallinnan menetelmiä sekä verkon valvontaa. Verkon tietoturvan kohdalla mietitään pääsynhallintaa, etäyhteyksiä ja tietoturvajärjestelmiä kuten palomuurit ja hyökkäyksien havainnointi sekä estojärjestelmät. Verkon valvonnassa käydään läpi tapoja, järjestelmiä ja protokollia, joilla verkon valvonta voidaan toteuttaa. Identiteetin- ja pääsynhallinnasta käydään läpi perusperiaatteet ja konseptit.

3.1 Verkon tietoturva

Verkon tietoturvalla pyritään suojaamaan itse verkko sekä verkkoon liitettyjä laitteita. Verkon tietoturva on osa tietoturvan kerrospuolustusta, jolla suojataan koko organisaatiota. Aihealueena verkon tietoturva on hyvinkin laaja, joten tässä aliluvussa keskitytään

IP-verkkojen suojaamiseen, etäyhteyksiin sekä tunkeutumisen esto- ja havaitsemisjärjestelmiin.

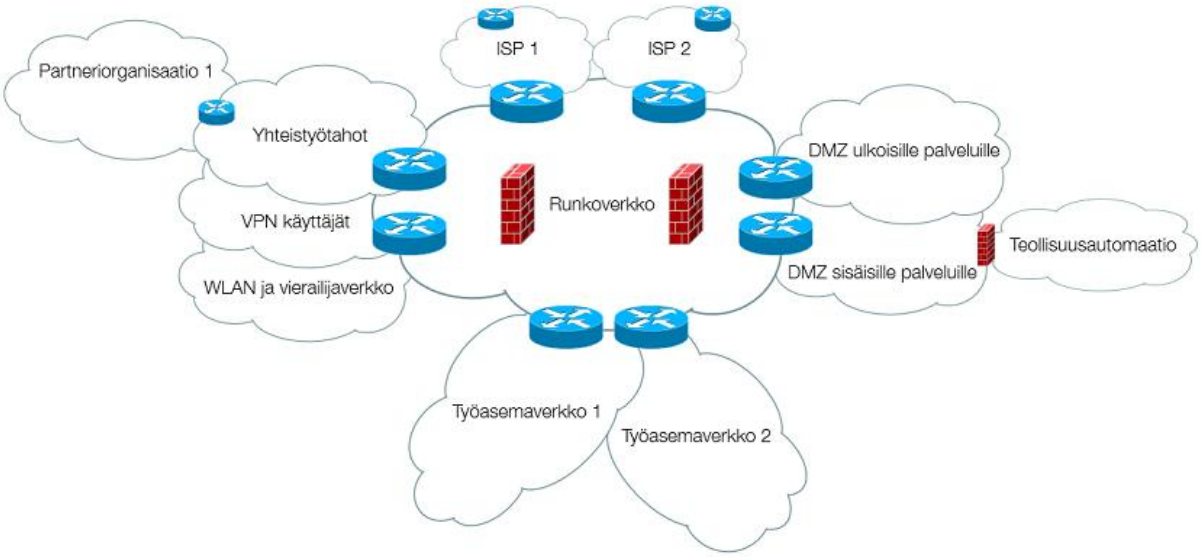
IP-verkkojen suojaaminen lähtee hyvästä verkkosuunnittelusta, hyvin suunniteltu verkko on tietoturvallinen ja toisinpäin. Verkonsuojaaminen tulee toteuttaa niin sanottuna kerrospuolustuksena, jossa verkko segmentoidaan erilaisiksi alueiksi. Alueita erottavat palomuurit. Kerrospuolustusmallin toteutukseen kuuluu se, että verkon suojaus toteutetaan jokaisella TCP/IP-protokollapinon tasolla fyysisestä kaapeloinnista ja laitteista sovelluskerrokselle asti. Kerrospuolustusmalliin kuuluu myös tietojärjestelmän sisäinen kerrospuolustus johon sisältyy esimerkiksi käyttöjärjestelmän ohjelmistopalomuuri, virustorjunta, sekä ohjelmistojen sisäiset suojaukset kuten syötteiden tarkastaminen, mutta tässä luvussa keskitytään vain alempien kerrosten suojaamiseen. Yksinkertaistettu malli verkkojen osastoinnista ja kerrospuolustuksesta on esitetty kuvassa 15 [39].



Kuva 15: Yksinkertainen IP-verkkojen osastointi kerrospuolustusmallin mukaisesti [39].

Kuvan 15 mukaisesti ulkoverkko ja DMZ-alue (engl. Demilitarized Zone) erotetaan toisistaan palomuurilla ja palomuuriin avataan pääsy DMZ:lla sijaitseville palveluluille vain tarvittaessa. DMZ:n ja ulkoverkon välillä tapahtuva yhdistäminen kannattaa sallia vain kontrollipisteen kautta, joka on tässä tapauksessa palomuuri. Palomuurissa kommunikointia voidaan valvoa ja rajoittaa toimivasti. Kontrollipisteelle ei tule sallia minkäänlaista kiertomahdollisuutta, koska tällöin rajoitukset ja valvonta eivät enää toimi. Samaa toimintatapaa on syytä hyödyntää kaikilla TCP/IP-protokollapinon kerroksilla. Esimerkiksi kerroskytkinten ja reititinten on syytä sijoittaa lukittuihin tiloihin ja sallia pääsy vain verkkoa ylläpitäville tahoille. Iso osa kerrospuolustuksen toteutusta onkin osastoida ja luokitella palvelut, jolloin palvelut saadaan sijoitettua niiden vaatimalla tietoturvallisuus alueelle ja toimiva verkon segmentointi on mahdollista.

Laajemmassa kuvassa verkon segmentointi on syytä toteuttaa käyttötarkoitusten mukaisesti. Kuvassa 16 on esitetty esimerkki, jossa keskikokoisen organisaation verkko on segmentoitu verkon käyttötarkoitusten mukaisesti.



Kuva 16: Esimerkki-organisaation verkon segmentoinnista [39].

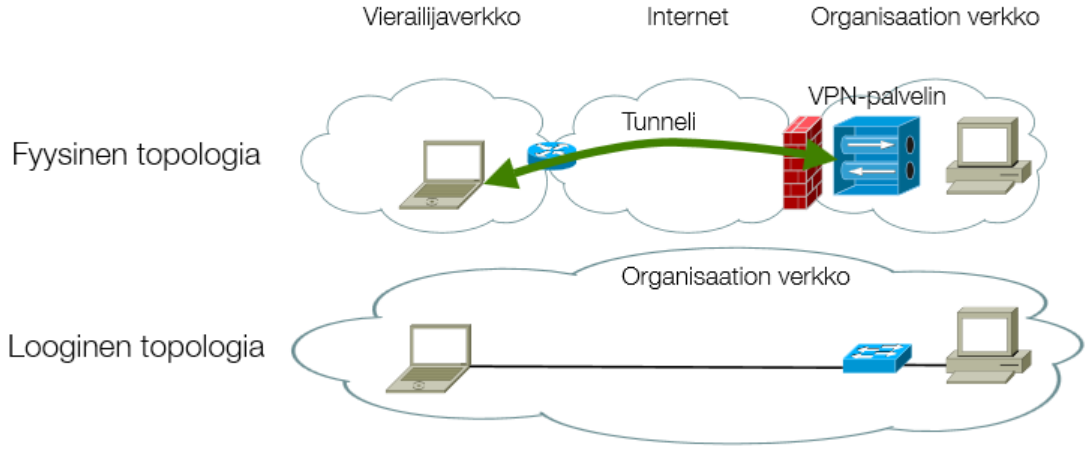
Kuvan 16 mukaisessa organisaatiossa on verkosta segmentoitu omiin verkkoihinsa esimerkiksi työasemat, VPN-käyttäjät, DMZ ja yhteistyötahot. Tämän lisäksi on verkkolaitteiden etähallintaa varten hyvä luoda erillinen ylläpitoverkko, jos se vain on toteutettavaan ympäristöön mahdollista.

Verkkosegmenttien väliin sijoitettu palomuuri sallii tai estää liikenteen siihen konfiguroitujen pääsylistojen mukaan. Palomuuri ei kuitenkaan auta mitään, jos hyökkääjä pääsee sen tavalla tai toisella ohittamaan esimerkiksi murtautumalla palomuurista avattuun palveluun. Tällaisten tilanteiden havaitsemiseen on kehitetty erilaisia järjestelmiä, joista käytetään yleisemmin nimityksiä IDS- (engl. Intrusion Detection System) tai IPS-järjestelmä (engl. Intrusion Prevention System). Yleisesti ottaen IDS-järjestelmillä pyritään havaitsemaan verkossa tapahtuvaa epätavallista toimintaa ja IPS-järjestelmillä pyritään havaitsemaan sekä proaktiivisesti myös estämään havaitut uhkat. Käytännössä havaitseminen tapahtuu esimerkiksi siten että IDS- tai IPS-järjestelmä tarkkailee verkon normaalia liikennettä aikansa ja luo tämän perusteella profiilin joka vastaa normaalia toimintaa verkossa. Tätä profiilia vasten jatkossa verrataan verkossa tapahtuvaa liikennöintiä ja tarvittaessa raportoidaan tai käynnistetään vastatoimet liikenteen poiketessa tarpeeksi tavanomaisesta profiilista. IDS- ja IPS-järjestelmät ovat osa tehokasta verkon valvontaa, johon tutustutaan yleisellä tasolla enemmän alaluvussa 3.4 [40].

Nykyään niin ohjelmistoyrityksissä kuin muissakin yrityksissä etätyönteko on arkipäivää niin kotoa kuin asiakkaidenkin tiloista. Organisaation palveluiden käyttäminen etäyhteyksien yli on kuitenkin toteuttava tietoturvallisesti. Yleisesti turvalliseksi koettu tapa on toteuttaa etäyhteydet VPN-tekniikkaa käyttäen. VPN-yhteys voi olla joko salattu tai salaamaton, mutta tässä yhteydessä keskitymme vain salattuihin yhteyksiin.

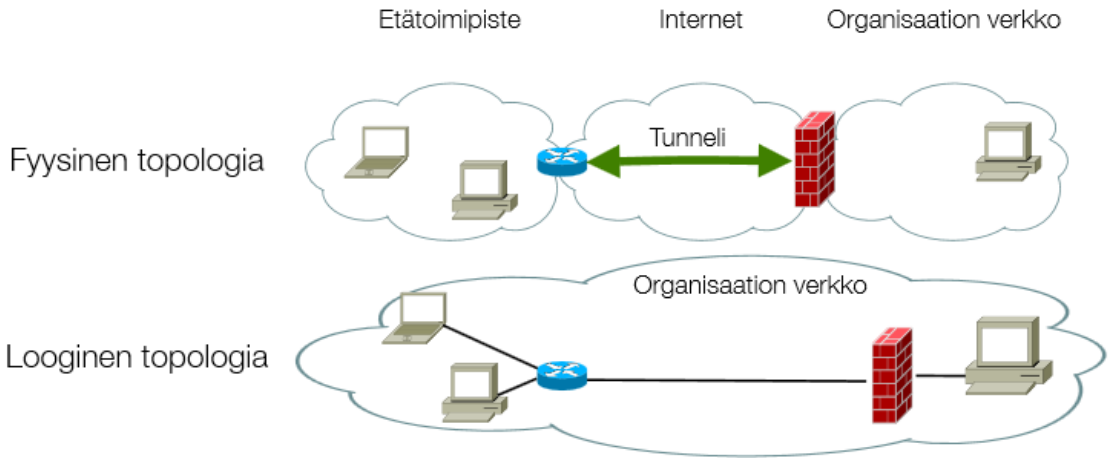
Isommassa kuvassa VPN:t voidaan jakaa kahteen eri malliin: organisaatioiden välisiin yhteyksiin eli site-to-site-VPN:iin ja asiakas-VPN:iin eli client-VPN:iin.

VPN tarkoittaa organisaation ulkopuolella olevien yksittäisten koneiden tai verkkojen liittämistä loogisella tasolla organisaation verkkoon tunnelin avulla. Käytännössä joko verkko- tai linkkikerroksen protokolla kehystetään verkko- tai siirtokerroksen protokollan sisään. Kuvassa 17 on esitetty esimerkki yksittäisen koneen liittämisestä organisaation verkkoon [41].



Kuva 17: Esimerkki Client-VPN:n käytöstä [41].

Kuvassa 17 organisaation työntekijä liittyy etäältä organisaationsa kotiverkkoon Internetin yli käyttäen client-VPN:ää. VPN-yhteyksien avulla voidaan liittää toisiinsa myös organisaation etätoimipisteiden verkkoja sekä muiden organisaatioiden verkkoja. Kuvassa 18 on esitetty organisaatioiden välisten verkkojen liittäminen toisiinsa [41].



Kuva 18: Esimerkki eri verkkojen liittämisestä toisiinsa käyttäen site-to-site-VPN:ää [41].

Organisaation liittäessä toisiinsa omia verkkojaan toimitaan yleensä kuvan 18 mukaisesti. Organisaatioiden välisien verkkojen yhteenliittäminen tehdään yleensä eteisverkkojen DMZ:jen yhteenliittämisenä. DMZ:lle sijoitetaan tai sieltä tehdään ohjaus niihin palveluihin, joita organisaatioiden välillä halutaan jakaa.

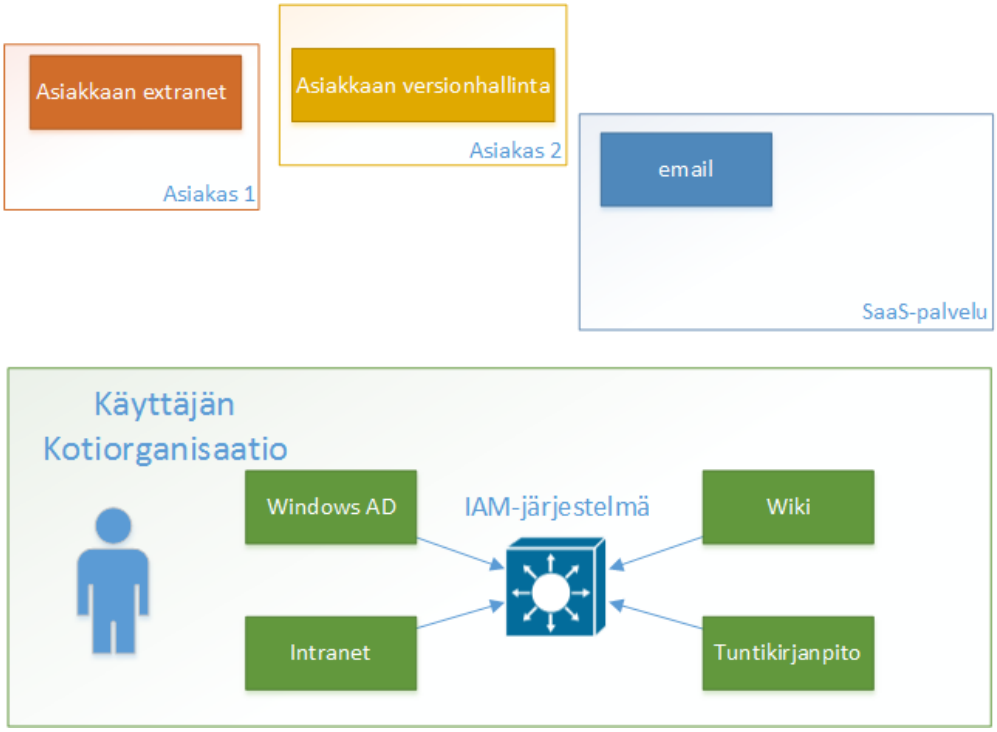
3.2 Identiteetin- ja pääsynhallinta

Seuraavissa alaluvuissa määrittelemme joitakin identiteetin- ja pääsynhallinnan kannalta tärkeitä asioita. Samaisiin asioihin palaamme myöhemmissä luvuissa 4 ja 5 käytännön kannalta. Ensiksi määrittelemme mitä on keskitetty identiteetin hallinta ja millaisia järjestelmiä sen toteuttamiseksi on olemassa. Toisena asiana määrittelemme autentikoinnin eli identiteetin todentamisen tavat. Kolmantena asiana määrittelemme käyttövaltuuksien hallinnan eli auktorisoinnin.

Keskitetty identiteetin hallinta

Käyttäjätunnukset ja niihin liittyvät käyttövaltuudet ovat tuttuja ohjelmistoyrityksissä työskenteleville niin kuin kaikille muillekin tänä päivänä erilaisia tietojärjestelmiä käyttäville. Henkilölle kertyy yleensä useita käyttäjätunnuksia ja salasanoja, jotka eivät monesti riipu millään tavalla toisistaan. Identiteetin- ja pääsynhallinnan käsitteet alkoivat muotua vuosituhannen vaihteessa kun käyttäjätunnuksia ja salasanoja vaativat palvelut alkoivat todella yleistyä. Organisaatioissa havahduttiin siihen lisätyömäärään minkä työntekijöiden käyttäjätunnusten avaaminen, sulkeminen ja unohtuneiden salasanojen hallinta aiheuttaa. Lisäksi tajuttiin tietoturvaongelmat, joita vanhat sulkematta jääneet käyttäjätunnukset heikkoine salasanoineen aiheuttivat [22].

Tietojärjestelmiä varten alettiin kehittää tuotteita, joiden avulla käyttäjän käyttäjätunnuksia, salasanoja ja käyttövaltuuksia eli niin sanottua identiteettiä voitaisiin hallita keskitetymmin. Kuvassa 19 on esitetty tyypillinen käyttäjän tilanne nykyisissä ohjelmistoyrityksessä, jossa on otettu käyttöön keskitetty identiteetin- ja pääsynhallintajärjestelmä eli IAM-järjestelmä (engl. Identity and Access Management) [22].



Kuva 19: Tyypillinen käyttäjän tilanne ohjelmistoyrityksessä, jossa on hyödynnetään jotakin IAM-järjestelmää. Eri väriset laatikot kuvaavat eri identiteettien kattavuutta [22].

Tällaisten tuotteiden avulla on monissa organisaatioissa toteutettu käyttäjille yksi organisaation sisäinen identiteetti johon on kytketty yksi käyttäjätunnus-salasana-pari. Tällä tunnukselle käyttäjä voi kirjautua kaikkiin organisaatioin sisäisiin palveluihin mihin hänellä on oikeus. Käyttäjän poistuessa organisaatiosta ylläpidon tehtävä helpottuu huomattavasti johtuen yhdestä ainoasta suljettavasta tunnuksesta [22].

Identiteetin- ja pääsynhallinta on kuitenkin mitä suurimmissa määrin johtamisongelma, sillä näppärinkään tekninen väline ei kykene auttamaan organisaation käyttäjätunnusten, salasanojen ja käyttövaltuuksien hallinnassa jos organisaatio ei ole pystynyt määrittelemään määrämuotoisia toimintatapoja kuten miten käyttäjävaltuudet määräytyvät ja kuka niitä saa pyytää sekä antaa [22].

Keskitetyn identiteetin- ja pääsynhallinnan hyötyinä voidaan pitää tietoturvallisuutta, tehokkuutta ja sitä että se mahdollistaa kokonaan uusia toimintatapoja tai -malleja, jotka eivät olisi muuten mahdollisia. Keskitetyn hallinnan kautta tietoturvallisuus paranee, sillä keskitetysti hallinnoituihin käyttäjätunnuksiin voidaan liittää salasanaa vahvempia tunnisteita ylläpitokulujen pysyessä kurissa keskitetyn rakenteen johdosta. Myös identiteetinhallinnan perisynnin torjunta eli käyttövaltuuksien poisto käyttövaltuuden perusteen loppumisen johdosta helpottuu. Samalla tehokkuus ja palvelutaso paranevat kun käyttäjätunnuksia ei tarvitse enää ylläpitäjän eikä käyttäjän kannalta pitää erillisille järjestelmille. Uudet toimintatavat mahdollistavat esimerkiksi luottamussuhteiden eli federaatioiden luonnin eri organisaatioiden välille, jolloin samoja käyttäjätunnuksia voidaan käyttää eri organisaatioiden palveluissa [22].

Ohjelmistoyritysten tarpeet keskitetyssä identiteetinhallinnassa ovat yleensä suhteellisen pieniä, joten varsinaisiin isoihin ja kalliisiin IAM-järjestelmiin ei monessakaan yrityksessä ole sijoitettu. Olennaisena osana IAM-järjestelmän teknologia-arkkitehtuuriin kuuluu LDAP-hakemisto (engl. Lightweight Directory Access Protocol), joka on laajasti käytetty standardi hakemistopalvelu identiteettitiedon tarjoamiseen sitä tarvitseville palveluille. LDAP-hakemistosta voidaan hakea hakukriteerin täyttäviä tietueita, esimerkiksi tietyn käyttäjätunnuksen haltijan attribuutteja. Käyttäjätietueeseen liittyy myös käyttäjän salasana, joten käyttäjä voidaan autentikoida LDAP-hakemistoa vasten. LDAP-määritysperhe määrittelee myös kuljetuskerroksen (TCP) päällä ajettavan request/response-protokollan, jolla asiakasohjelmat voivat asioida LDAP-hakemiston kanssa. Protokollan tärkeimmät viestit ovat search, jolla haetaan tietoja hakemistosta ja bind, jonka avulla hakemisto voi autentikoida käyttäjän. Koska LDAP-protokolla itsessään ei huolehdi tiedonsiirron turvallisuudesta, tunneloidaan se yleensä TLS/SSL-protokollan (Security Sockets Layer/Transport Layer Security) yli. Koska LDAP-hakemistot sisältävät henkilötietoja, niihin ei yleensä sallita liikennettä sisäverkon ulkopuolelta. LDAP-protokollaa toteuttavia järjestelmiä ovat esimerkiksi avoimen lähdekoodin slapd sekä Microsoftin Active Directory, jonka sisältämä LDAP-palvelu onkin yksi yleisimmin käytössä olevista LDAP-palveluista.

Ideaalitilanteessa kaikki organisaation järjestelmät olisivat organisaation keskitetyn identiteetinhallinnan piirissä. Käytännössä ideaaliin kuitenkin päästään harvoin eikä siihen pyrkiminen ole edes tarkoituksenmukaista, koska jokaisen integraation rakenta-

minen ja ylläpito maksaa, ja järjestelmän rajapinnoista riippuen integraatio voi pahimmillaan olla työlästäkin. Yleensä lähtökohtana on käydä organisaation tietojärjestelmät läpi ja käyttää harkintaa siitä, kannattaako se integroida organisaation keskitettyyn identiteetinhallintaan. Käytännössä varsinkin sellaiset palvelut, joissa sulkematta unohtuneet tunnukset aiheuttavat suurimmat tietoturvariskit kuten henkilökunnan VPN-palvelu. On syytä muistaa, että identiteetinhallinta ja sen mahdollisuudet ovat laajempia kuin pelkkää käyttäjätunnushallintaa, mutta pienissä ohjelmistoyrityksissä jo keskitetty käyttäjätunnushallintaa on todella hyvä lähtölaukaus kun identiteetin- ja pääsynhallinnan menetelmiä aletaan ottaa käyttöön. Käytännössä pienemmissä yrityksissä keskitetty identiteetin hallinta on monesti alussa toteutettu siten että aluksi keskeisimmät ja lopulta kaikki mahdolliset palvelut käytännössä autentikoidaan Active Directorya vasten [22].

Identiteetin todentaminen eli autentikointi

Identiteetin todentaminen tarkoittaa, että identiteetin ja sitä tosielämässä vastaavan henkilön välille rakennetaan kytkös. Tietojärjestelmä siis varmistaa tavalla tai toisella, että järjestelmään kirjautuu sisään sama henkilö, jolle tietty järjestelmään luotu identiteetti kuuluu. Kytkös on yleensä voimassa istunnon ajan, ja istunto suojataan yleensä kryptografisesti, kuten symmetrisellä istuntoavaimella. Istunto päättyy, kun käyttäjä kirjautuu ulos [22].

Identiteetin todentamiseen on lukuisia erilaisia luotettavuudeltaan vaihtelevia menetelmiä ja välineitä, jotka yleensä jaetaan kolmeen kokonaisuuteen.

- Jotain, mitä henkilö tietää tai muistaa kuten salasana tai PIN-koodi.
- Jotain, mitä henkilöllä on hallussaan, kuten toimikortti (engl. smart card), toimiavain (engl. token), pankkitunnukset, matkapuhelin tai kertakäyttösalasanalista tai -laite.
- Jotain, mitä henkilö on tai kuinka hän käyttäytyy eli biometrinen tunnistus.

Suomessa laki vahvasta sähköisestä tunnistamisesta määrittelee tunnistamisen vahvaksi, jos vähintään kaksi yllämainituista tavoista on käytössä yhtä aikaa. Esimerkiksi pankkiautomaatin käyttöön tarvitaan sekä pankkikortti että PIN-koodi, jolloin on kyse vahvasta tunnistamisesta. Vahva tunnistamisen vastakohta on heikko tunnistaminen kuten esimerkiksi pelkkä salasana, jonka voi urkkia käyttäjän olan yli [22].

Jotta vahva tunnistaminen todella toimii, niin myös ensitunnistus on tehtävä vahvasti. Ensitunnistus on se hetki, jolloin käyttäjä saa haltuunsa tunnistukseensa tarvittavan välineen kuten ensimmäisen käyttäjätunnus-salasana-parinsa tai toimikorttinsa ja sen PIN-koodin. Usein tämä tapahtuu silloin kun esimerkiksi uusi työntekijä aloittaa uudessa työpaikassaan. Usein ensitunnistusta pidetään vahvana, jos käyttäjän täytyy kuitata tunnistusvälineensä kasvotusten rekisteröintipisteestä. Tällöin myös hänen tulee esittää henkilöllisyydestään luotettava asiakirja, joina Suomessa on totuttu pitämään passia, poliisin myöntämää henkilöllisyystodistusta ja yleensä myös ajokorttia. Näin pyritään kasvattamaan rekisteröityjen tietojen luotettavuutta. Vaikka käytössä olisikin vahva tunnistaminen, on huomioitava että vahvasta tunnistuksesta ei ole mitään hyötyä jos tunnukset luovutetaan kolmannelle osapuolelle. Tämän vuoksi esimerkiksi pankit ovat

erityisen tarkkoina siitä kenen tiedossa ja hallussa kunkin pankkitunnukset ovat. Esimerkiksi ulkoisiin palveluihin toisen pankkitunnuksilla tunnistaminen ei onnistu edes valtakirjan välityksellä, sillä kolmannen osapuolen on pystyttävä luottamaan siihen että tunnistettava käyttäjä on juuri se samainen henkilö keneksi hänet tunnuksilla tunnistetaan [22][23].

Käyttövaltuuksien hallinta eli auktorisointi

Käyttäjän tunnistamisen jälkeen on ratkaistava kysymys, onko käyttäjällä valtuus eli oikeus suorittaa hänen pyytämänsä toiminto. Tätä kutsutaan pääsynvalvontapäätökseksi, johon huipentuu käyttövaltuuksien hallinnaksi eli auktorisoinniksi kutsuttu prosessi. Yhdessä käyttäjän tunnistuksen kanssa pääsynvalvontapäätös muodostaa pääsynvalvonnaksi kutsutun toimintosarjan, joka tapahtuu sillä hetkellä kun käyttäjä kirjautuu palveluun. Pääsynvalvonta on keskeinen osa pääsynhallintaa, joka sisältää lisäksi käyttäjien käyttövaltuuksien hallinnan. Auktorisoinnin lähtökohtana on, että käyttäjä on tavalla tai toisella tunnistettu. Se millä tavalla tämä tunnistaminen on toteutettu, ei ole auktorisoinnin kannalta tärkeää. Joissakin suojattavissa kohteissa pääsynvalvontapäätökseen voi kuitenkin vaikuttaa tunnistuksen vahvuus, jolloin tunnistuksen vahvuuden voidaankin sanoa olevan pääsynvalvontapäätöksen attribuutti. Esimerkiksi joissain ympäristöissä normaaleille käyttäjille sallitaan heikko käyttäjätunnus-salasana-pariin pohjautuva tunnistaminen, mutta ylläpitäjän oikeuksien käyttöön ottoon vaaditaan tunnistaminen käyttäen toimikorttia [22].

Pääsynvalvontamatriisi on klassinen tapa esittää käyttäjän käyttövaltuudet suojattavaan kohteeseen. Suojattavat kohteet esitetään matriisin sarakkeissa, ja käyttäjät riveillä. Matriisin soluihin kirjataan toiminnot, jotka kullekin käyttäjälle on sallittu kyseiseen suojattavaan kohteeseen. Yksinkertainen pääsynvalvontamatriisi on esitetty taulukossa 1.

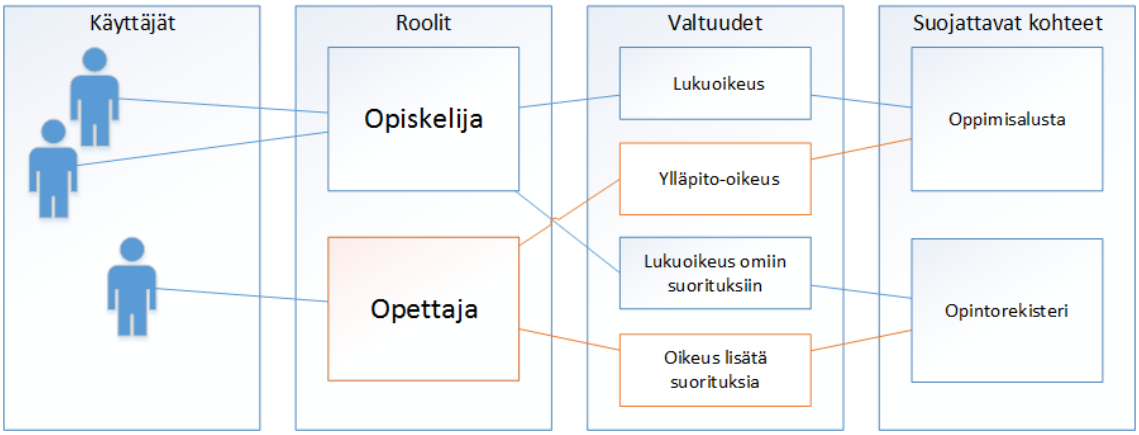
Taulukko 1: Esimerkki pääsynvalvontamatriisista [22].

	/home/timo/foo	/tmp/bar	/etc/passwd
Timo	read, write	read	-
Paula	read	read,write	-

Oheinen esimerkkitaulukko on tässä tapauksessa perinteinen Unix-tiedostojärjestelmä. Timo-nimisellä käyttäjällä on kotihakemistonsa kaikkiin tiedostoihin luku- ja kirjoitusokeus. Paula-nimiselle käyttäjälle Timo on antanut lukuoikeuden, mutta ei kirjoitusoikeutta, kotihakemistonsa tiedostoon foo. Kummallakaan käyttäjistä ei ole luku- eikä kirjoitusoikeutta tiedostoon /etc/passwd. Pääsynvalvontamatriisin huono puoli on, että sen koko kasvaa hyvin nopeasti hallitsemattomaksi, kun käyttäjien ja suojattavien kohteiden määrä kasvaa. Tämän vuoksi pääsynvalvontaa varten on kehitetty runsaasti myös muita pääsynhallintamalleja. Pääsynvalvontamatriisi on kuitenkin hyvin käytännöllinen ja yksinkertainen pääsynvalvontamalli monissa ohjelmistoyrityksissä tarvittavissa pääsynvalvontatoimissa kunhan sen rajoitukset ja käyttötarkoitus muistetaan ottaa huomioon [22].

Rooliin perustuva pääsynvalvonta eli RBAC (engl. Role Based Access Control) on toinen yleinen pääsynvalvontamalli. RBAC:ssa käyttäjän ja käyttövaltuuden väliin on luotu abstraktio: käyttäjälle annetaan rooleja, jotka kuvaavat esimerkiksi hänen työtehtäviään organisaatiossa. Käyttövaltuudet annetaan puolestaan näille rooleille, joiden kautta yksittäiset käyttäjät saavat lopulta käyttövaltuutensa. Yksittäisen käyttäjän käyttövaltuudet saadaan selville kun selvitetään, mitkä roolit hänelle on asetettu. Yksityiskohtaisemmat käyttövaltuudet saadaan selville kun tarkastellaan kyseiselle roolille annettuja oikeuksia.

Esimerkkinä voidaan ottaa tilanne, jossa korkeakoulussa pidetään kurssi. Tässä yksinkertaisessa roolijaossa kurssilla on olemassa opettajat ja opiskelijat. Opettajan tehtävänä on pitää ja valmistella kurssiin liittyvät tapahtumat kuten luennot ja tentti sekä arvostella opiskelijoiden suoritukset ja lopulta tallentaa suoritukset opintorekisteriin. Opiskelijan tehtävänä on opiskella ja käydä kurssin tentissä. Kurssin rooliperusteinen käyttövaltuusmalli on esitetty kuvassa 20 [22].



Kuva 20: Yksinkertainen rooliperustainen käyttövaltuusmalli [22].

Kuvasta 20 nähdään että käyttäjät ovat periaatteessa samanlaisia, mutta annettu rooli valtuuttaa käyttäjälle erilaisia valtuuksia. Esimerkiksi opettajan rooli valtuuttaa käyttäjän tekemään oppimisalustalla ylläpitotoimia ja opiskelijan rooli antaa lukuoikeuden omiin suorituksiin opintorekisteristä. Tarkoituksenmukaisten roolien tunnistaminen eli roolien louhinta (engl. role mining) on keskeinen osa rooliperustaisen käyttövaltuusmallin suunnittelua ja käyttöönottoa. Käyttäjillä voi olla useita rooleja ja rooleille voidaan määritellä lapsirooleja jolloin rooleja voidaan periyttää. Rooliperusteinen käyttövaltuusmalli on hyvin yleisesti käytössä tietojärjestelmissä ja se soveltuu myös yleisemmin ohjelmistoyritysten käyttöön [22].

Rooliperustaisen käyttövaltuusmallin ohella käytetään myös jäljitettävyyteen perustuvaa pääsynvalvontaa (engl. accountability based access control), joka oikeastaan ei ole pääsynvalvontamalli vaan sen filosofiassa todetaan aukottoman pääsynvalvonnan toteuttamisen olevan mahdotonta. Tämän vuoksi käyttäjävaltuuksia rajataan hyvin väljästi tai esimerkiksi erilaisia rooleja luodaan vain vähän. Tämän vastapainona jokainen käyttäjän toimi kirjataan erilliseen lokiin, josta mahdollisia väärinkäytöksiä voidaan myöhemmin tarkastella. Tietojärjestelmän, jonka pääsynvalvonnassa käytetään jäljitettävyyteen perustuvaa pääsynvalvontaa, käyttäjillä on yleensä suuri vastuu tekemisistään

ja sitä käytetään esimerkiksi väestö- ja potilastietojärjestelmissä. Jäljitettävyyteen perustuvan pääsynvalvonnan ongelmana on kuitenkin se että se harvoin noudattaa seuraavaksi esiteltävää vähimmän käyttövaltuuden periaatetta, jolloin esimerkiksi järjestelmien suojaaminen oikeudettomalta käytöltä tai erehdyksiltä jää toteutumatta [22].

Vähimmän käyttövaltuuden periaate (engl. least privilege) on eräs tietoturvallisuuden keskeisistä periaatteista. Sen mukaan käyttäjällä saa olla käytössään vain niin laajat käyttövaltuudet kuin mitkä hän tarvitsee hänelle kuuluvien tehtävien hoitamiseen. Koko käyttövaltuuksien mallintamisen ja hallinnan problematiikka rakentuukin tämän periaatteen ympärille; muutoinhan kaikille käyttäjille voitaisiin antaa aina kaikki valtuudet, ja sen kummempaa käyttövaltuuksien hallintaa ei tarvittaisi. Vähimmän käyttövaltuuden periaatetta rikotaan, jos organisaatio ei vaivaudu mallintamaan suojattavien kohteidensa käyttövaltuuksia riittävästi, vaan antaa käyttäjille ylläpitoa helpottaakseen tarpeettoman suuret käyttövaltuudet. Organisaatioiden toiminta ja käyttäjien käyttövaltuudet organisaatiossa ovat toki tosielämässä monimutkaisia, mutta edes keskeisimmät suojattavat kohteet olisi syytä tunnistaa ja suojata osana organisaation riskienhallintaa. Esimerkiksi ohjelmistokehittäjillä on monissa tapauksissa syytä olla ylläpito-oikeudet omaan työasemaansa, mutta yrityksen bisneskriittisten palveluiden ylläpito-oikeuden pitäisi olla vain sitä tarvitsevien hallussa [22].

Vaarallisten työtehtävien eriyttämisessä (engl. Segregation of Duties, SOD) on kyse taloushallinnon jo vuosisatoja tietämästä ongelmasta, jossa kassaa ja kirjanpitoa ei saa hoitaa yksi ja sama henkilö. Yritysmaailmassa tämä näkyy useimmin siinä että matkalaskua ei saa luoda sekä hyväksyä sama henkilö. Tällä ei ehkäistä pelkästään väärinkäytöksiä vaan myös tärkeidenkin tehtävien hoidossa tapahtuu jatkuvasti erehdyksiä ja virheitä. Ohjelmistoyrityksien kannalta tämä koskee erityisesti tietojärjestelmien käyttövaltuuksia. Kuvan 11 kuvaamaa tilannetta miettien vaarallinen yhdistelmä syntyisi, jos opettaja voisi olla omalla kurssillaan myös opiskelijanroolissa ja siten antaa kurssista opintosuorituksen itsellensä. Kun edes valtuuksien myöntämisen hyväksyminen ja toteuttaminen on eriytetty kahdelle eri henkilölle, joutuu käyttövaltuutusta pyytävä henkilö vakuuttamaan myös hyväksyjän siitä. että hänelle on turvallista myöntää haettu käyttövaltuus. Tällä toimenpiteellä vältetään erehdyksiä ja virheitä [22].

Organisaation koosta ja toimialasta riippuu, missä määrin vaarallisten työtehtävien eriyttämiseen kiinnitetään huomiota käyttövaltuushallinnassa. Pienessä organisaatiossa kuten monissa ohjelmistoyrityksissä kaikkia vaarallisiksi tunnistettuja työyhdistelmiä ei pystytä eriyttämään eri henkilöille, koska mahdolliset henkilöt loppuvat viimeistään lomakauden koittaessa. Tällöin riski joudutaan vain toteamaan ja ottamaan, ja mahdollisesti rakentamaan korvaavia kontrolleja riskin hallitsemiseksi [22].

3.3 Organisaation riskienhallinta

Ohjelmistoyritysten ollessa usein pieniä syntyy riskienhallinnallisesta näkökulmasta ongelma: mihin yrityksessä luotetaan ja kuinka suojaudutaan sisäisiä uhkia vastaan? Uhkakuvina voidaan pitää pahantahtoista entistä tai nykyistä työntekijää, jolle on suotu

aikojen kuluessa suuret käyttövaltuudet. Myös omasta mielestään hyväntahtoinen työntekijä voi osoittautua organisaation kannalta uhaksi. Esimerkiksi Yhdysvaltain tiedustelutietoja vuotanut Edward Snowden toimi mielestään oikein vuotaessaan tietoja, organisaation kannalta kuitenkin tapahtui suurta vahinkoa. Varsinkin pienissä organisaatioissa täytyy kuitenkin keskeisiin henkilöihin pystyä luottamaan ilman että heitä rajoitetaan liikaa tietoturvapolitiikkaan tai muuhun ohjeistukseen nojaten. Jotkin riskit on siis vain hyväksyttävä.

Organisaation riskien hallintaan on olemassa NIST:n kehittämä riskienhallinnan viitekehys RMF (engl. Risk Management Framework). RMF on riskiperustainen viitekehys, jonka kuusivaiheinen ohjeistus auttaa erilaisia organisaatioita paremmin arvioimaan, hallitsemaan ja minimoimaan tietojärjestelmistä syntyviä riskejä [24].

Ensimmäisenä vaiheena RMF:ssä on kategoriointi, jossa tietojärjestelmät kategorioidaan niiden aiheuttaman riskien perusteella [24].

Toisena vaiheena valinta, jossa eri kategorioille valitaan turvallisuustaso joka siihen kuuluvilta tietojärjestelmiltä vaaditaan [24].

Kolmantena vaiheena on implementointi, jossa järjestelmien sisäiset turvakontrollit luodaan ja dokumentoidaan [24].

Neljäntenä vaiheena on arvostelu (engl. assess), jossa järjestelmiin toteutetut turvakontrollit arvioidaan. Arvioinnissa käydään läpi esimerkiksi se toimivatko kontrollit niin kuin niiden pitää, ovatko kontrollit toteutettu oikein sekä tuottavatko ne sen lopputuleman tietoturvan kannalta, joka järjestelmälle on asetettu tavoitteeksi [24].

Viidentenä vaiheena on valtuutus (engl. authorize) tai hyväksyminen, jossa tietojärjestelmä ensin arvioidaan ja sitten hyväksytään kokonaisuutena tietojärjestelmään toteutetut turvakontrollit sekä tietojärjestelmään kohdistuvat riskit [24].

Kuudentena vaiheena on valvonta, jossa järjestelmään kohdistetaan sen ollessa käytössä, valvontaa. Valvonnalla pyritään selvittämään kuinka hyvin turvakontrollit toimivat, mahdolliset muutokset dokumentaatioon, muutoksista aiheutuvat vaikutukset järjestelmään sekä toteuttamaan raportointi organisaation johdolle [24].

3.4 Valvonta, jäljitettävyys ja raportointi

Modernin käsityksen mukaan yrityksen tietoverkko ei ole koskaan valmis vaan se kasvaa ja kehittyy yrityksen tarpeiden mukana. Kehityksen ja ongelmien ratkaisun kannalta on tärkeää tietää kuinka hyvin tietoverkko tai tietojärjestelmä nykyisellään toimii ja mistä löytyvät pahimmat pullonkaulat. Loppukäyttäjän kannalta valvonta, jäljitettävyys ja raportointi ovat usein huomaamattomia toimintoja, joilla on kuitenkin tärkeä rooli niin tietojärjestelmienkehityksessä kuin myös identiteetin- ja pääsynhallinnan täydentävinä kontrolleina. Esimerkiksi potilastietojärjestelmissä käytetylle jäljitettävyyteen perustuvalle käyttäjien hallinnalle asetetaan yleensä tiukat vaatimukset siitä mitä kaikkea on pystyttävä jäljittämään myöhemmin [22][25][26].

Verkonvalvonnan (engl. network monitoring) tarkoituksena on kerätä tietoa yksittäisiltä verkon osilta keskitettyyn verkon valvontaan. Verkonvalvonta on osa laajem-

paa kokonaisuutta jota kutsutaan verkon hallinnaksi (engl. network management). Samaan tapaan kuin verkon osia voidaan valvoa myös palvelimia tai kokonaisia tietojärjestelmiä, jolloin puhutaan tietojärjestelmä valvonnasta. Verkkolaitteilta kerätään esimerkiksi tietoja siirretyistä datamääristä ja laitteiden toimintakyvystä. Palvelimilta kerätään tietoja kuten vapaan levytilan määrä ja prosessorikuorma. Näistä tiedoista voidaan muodostaa pitkältä aikaväliltä trendejä, joiden avulla voidaan paremmin ennustaa tulevaisuuden päivitystarpeita koko tietojärjestelmän osalta [22][25][26].

Jäljitettävyyteen liittyvällä valvonnalla tarkoitetaan luotettavan kirjausketjun (engl. audit trail) muodostumista identiteetin- ja pääsynhallintaan liittyvistä tapahtumista. Tämän kirjausketjun tehtävä on pitää luotettavasti kirjaa siitä mitä ja milloin kukin käyttäjä on tehnyt. Kirjausketjua voidaan käyttää myöhemmin esimerkiksi ongelmien selvittämiseen tai jopa todistusaineistona tapauksen niin vaatiessa. Käytännössä jäljitettävyyteen liittyvä valvonta perustuu yleensä keskitettyyn lokien hallintaan, joita kerätään käyttäjien tunnistamisesta ja heidän käyttövaltuuksiensa myöntämisestä ja käyttämisestä. Lokien hallinta, kerääminen ja eheys on hoidettava tietoturvan hyvien käytäntöjen mukaan, jotta kirjausketju pysyy luotettavana. Täytyy kuitenkin muistaa että jäljitettävyyden toteutumisessa tunnuksien täytyy aina liittyä kiistämättömästi jonkun identiteettiin. Ryhmätunnuksista luopuminen onkin jäljitettävyyden kannalta pakollista, joka tarkoittaa henkilökohtaisten tunnusten käyttöä kaikissa järjestelmissä [22].

Informaation keräämiseen verkkolaitteilta tai palvelimilta on monia tapoja. Standardoituja tapoja on esimerkiksi laajalti käytössä oleva SNMP (engl. Simple Network Management Protocol) sekä uudempi NETCONF (engl. network configuration protocol). Perinteisiä tapoja ovat myös verkkotekniikan perustyökalut kuten ping ja traceroute. Erityisesti palvelimiin voidaan asentaa myös erilaisia asiakaspuolen valvontaohjelmistoja, joita kutsutaan agenteiksi. Monilla valvontaohjelmistoilla on omat agenttinsa eri alustoille. Lokien keräämiseen käytetään yleensä perinteistä syslog-formaattia. Lisäksi laitteita ja palvelimia voidaan valvoa myös täysin itsetehdyillä skripteillä. Konkreettisiin ohjelmistoyritysten tietojärjestelmien valvonta toimenpiteisiin, keskitettyyn lokitukseen ja niihin liittyviin toimenpiteisiin tutustutaan luvussa 4 [25][26][27].

4 TIETOJÄRJESTELMÄN KEHITYSRATKAISUT ERÄÄSSÄ OHJELMISTOYRITYKSESSÄ

Tässä luvussa tutustutaan siihen miten tiettyjä menetelmiä on hyödynnetty tai voidaan hyödyntää ohjelmistoyrityksessä. Aliluvuissa käsitellään identiteetin- ja pääsynhallinnan, verkko- ja tietojärjestelmävalvonnan sekä virtualisoinnin ja pilvipalveluiden kehitystä ja ratkaisuja esimerkki-organisaation kautta. Esimerkki-organisaatiomme on muutaman kymmenen hengen ohjelmistokonsulttiyritys, jolla on oma ylläpito. Asiakkaiden tietoturvavaatimusten takia suoraan julkisten pilvipalveluiden käyttö ei ole mahdollista vaan ratkaisujen on lähtökohtaisesti pysyttävä sisäisinä.

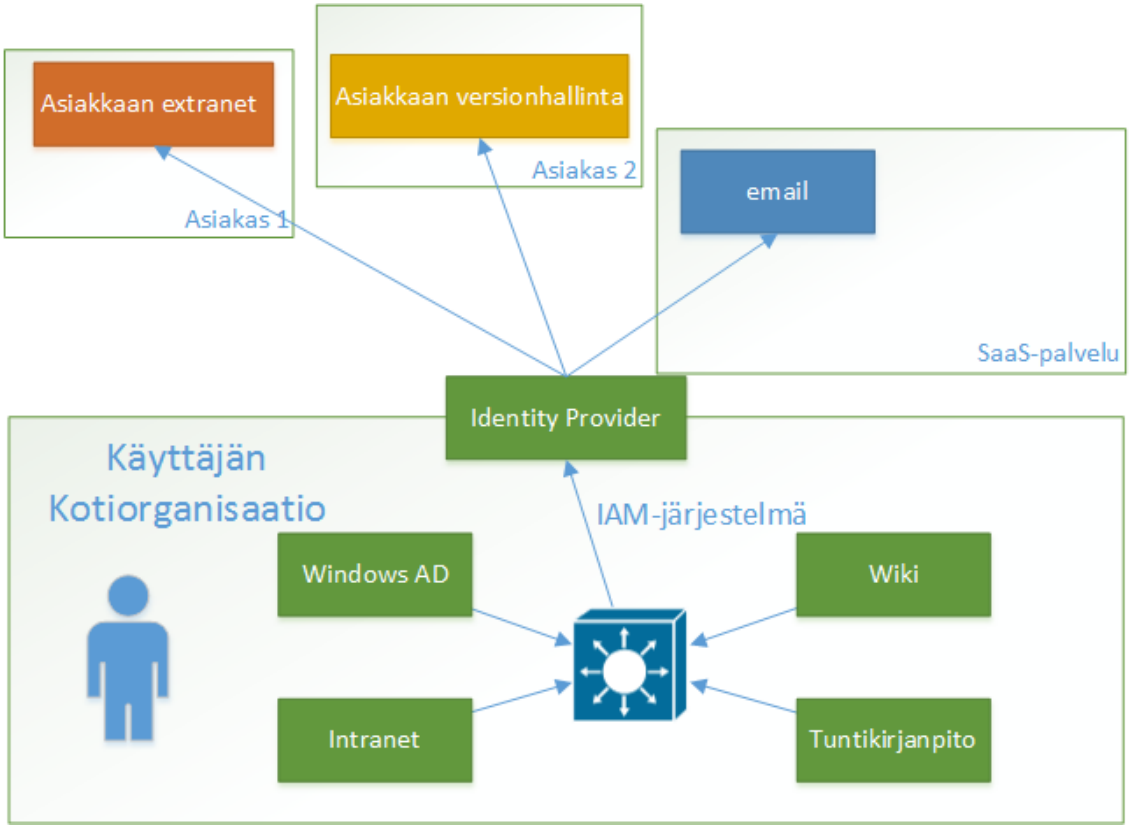
4.1 Identiteetin ja pääsynhallinnan kehitys

Identiteetin- ja pääsynhallinnan menetelmät eivät ole yleensä se ensimmäinen asia mihin ohjelmistoyrityksissä panostetaan. Ohjelmistoyrityksen tarinan alussa koko henkilökunta mahtuu monesti samaan huoneeseen ja heidän verkkoaan toimittaa yksi kytkin tai kotireititin keskellä pöytää, johon kaikki kytkeytyvät. Käyttäjätunnukset työkoneisiin ovat paikallisia eikä erilliselle infrastruktuurille nähdä vielä varsinaista hyötyä.

Yrityksen kasvaessa kuvaan astuu Windows-verkkojen tapauksessa Microsoft Active Directory (AD), jonka käyttö keskittää ja sujuvoittaa käyttäjätunnusten hallintaa vähänkään suuremmassa organisaatiossa. AD:ta vasten voidaan autentikoida myös muita tietojärjestelmiä käyttäen Kerberos- tai LDAP-protokollia. Lisäksi autentikointi voidaan toteuttaa käyttäen SSO:ta (engl. Single Sign On) eli kertakirjautumista. Tällöin esimerkiksi käyttäjän ei tarvitse erikseen kirjautua sisään intranetin käyttöä varten. Organisaatioissa on yleensä käytössä myös muita kuin Windows-käyttöjärjestelmiä ja myös näihin kirjautuminen voidaan toteuttaa AD:ta vasten käyttäen esimerkiksi Linuxissa SSSD:tä tai WINBIND:iä. Käyttäjän kirjautuessa ensimmäistä kertaa sisään haluttuun järjestelmään hänelle luodaan kotihakemisto sekä paikallinen tunnus, joka autentikoidaan aina kirjautuessa AD:ta vasten. Toinen mahdollisuus on säilyttää kotihakemistot erillisellä verkkolevyllä ja hakea ne sieltä kirjautessa, mutta ainakaan testausvaiheessa tähän ei ole vielä siirrytty. AD ei varsinaisesti ole vielä IAM-järjestelmä vaan suuremmissa organisaatioissa se sidotaan osaksi suurempaa identiteetinhallintajärjestelmää eli IAM-järjestelmää. Kuitenkaan pienemmässä organisaatiossa ei ole nähty tarvetta varsinaiselle IAM-järjestelmälle vaan on tyydytty sitomaan palveluita kiinni olemassa olevaan infrastruktuuriin [22].

AD:n vastatessa sisäisestä autentikoinnista ja auktorisoinnista ongelmaksi jäävät vielä samat asiat ulkoisten palveluiden ja yhteistyötahojen kanssa. Tähän tarpeeseen

vastaa federoitu identiteetinhallinta. Tyypillisessä tilanteessa organisaation työntekijä joutuu kirjautumaan esimerkiksi asiakas organisaation extranet-palveluun. Kuvassa 21 on esitetty yksinkertainen federoitu identiteetinhallinta [22].



Kuva 21: Federoitu identiteetinhallinta. Vihreät laatikot kuvaavat yhden identiteetin kattavuutta [22].

Kuvan 21 mukaisesti tunnistautuminen kotiorganisaation ulkopuolisiin palveluihin voidaan tehdä federoidun identiteetinhallinnan kautta. Federointi tarkoittaa käytännössä sitä että tunnistusta kaipaava palvelu luottaa tunnistusta tarjoavaan palveluun. Tarkemmin identity provider on palvelu, joka tarjoaa tunnistus palvelun tunnistukseen nojaavalle palvelulle. Federoinnin hyödyiksi voidaan laskea esimerkiksi se että käyttäjätunnuksen sulkeminen kotiorganisaatiossa sulkee pääsyn myös siihen nojaaviin palveluihin. Lisäksi federoitu käyttäjätunnistus saattaa osaltaan luoda edellytyksiä organisaation toimintojen järjestämiseen uudella tavalla kun esimerkiksi uuteen SaaS-palveluun ei tarvitse erikseen luoda tunnuksia käyttäjille [22].

Kaikkien näiden identiteetin- ja pääsynhallintatapojen käyttö vaatii suunnitelmallisuutta. Tulevaisuudessa uusia järjestelmiä hankittaessa täytyy varmistua siitä että ne tukevat tai niihin on mahdollista toteuttaa valittu keskityttyyn identiteetinhallintaan nojaava identiteetin- ja pääsynhallinta. Tällöin puhutaan teknologia-arkkitehtuurista, joka onkin organisaation kannalta tärkeää määritellä, että turhilta kömmähdyksiltä vältytään [22].

4.2 Valvonnan kehitys

Kuten aikaisemmin aliluvussa 3.4 todettiin kehityksen ja ongelmien ratkaisun kannalta on tärkeää tietää kuinka hyvin tietoverkko tai tietojärjestelmän alustana toimivat palvelimet nykyisellään toimivat. Tällöin ilmenevät ongelmat voidaan havaita ja mahdollisesti jo korjata ennen niiden todellista syntymistäkään. Loppukäyttäjälle tämä näkyy palveluiden parempana toimintavarmuutena. Valvonta tulisi toteuttaa lisäksi keskitettynä, jotta näkyvyys koko organisaation alueella saadaan maksimoitua. Lisäksi erillisiin järjestelmiin kirjautuminen tietojen saamiseksi on hyvin vaivanloista. Tämä luku on jaettu kahteen eri alilukuun, joista ensimmäisessä käsitellään tietojärjestelmien osien valvontaa ja jälkimmäisessä käsitellään lokien keskitettyä keräämistä.

4.2.1 Tietojärjestelmä valvonnan kehitys

Ohjelmistoyrityksissä ajetaan yleensä moninaisia erilaisia tietojärjestelmiä kuten laskutuksen järjestelmät, erilaiset versionhallinnat ja tehtävien hallinnan työkalut. Nämä työkalut ovat monesti osa jokapäiväistä työntekemistä ja niiden käyttö on turvattava. Esimerkki-organisaatiossa alkuun ei järjestelmiä valvottu mitenkään. Ongelmien ilmetessä aloitettiin selvitystyö käytännössä nollasta, jolloin se oli hidasta ja vaivanloista. Esimerkiksi jonkun kytkiessä väärin konfiguroidun verkkolaitteen tuotantoverkkoon koko verkko jumiutui. Tämän ongelman ratkaisemiseksi otettiin käyttöön ensimmäinen verkon valvontaa suorittanut sovellus eli Arpalert. Arpalert käytännössä tarkkailee verkkoa ja pitää kirjaa Ethernet-verkosta löytyvistä verkkolaitteista MAC-osoitteen perusteella. Uuden ennen näkemättömän laitteen liittyessä verkkoon, Arpalert lähettää hälytyksen sähköpostiviestinä ylläpitäjälle [42].

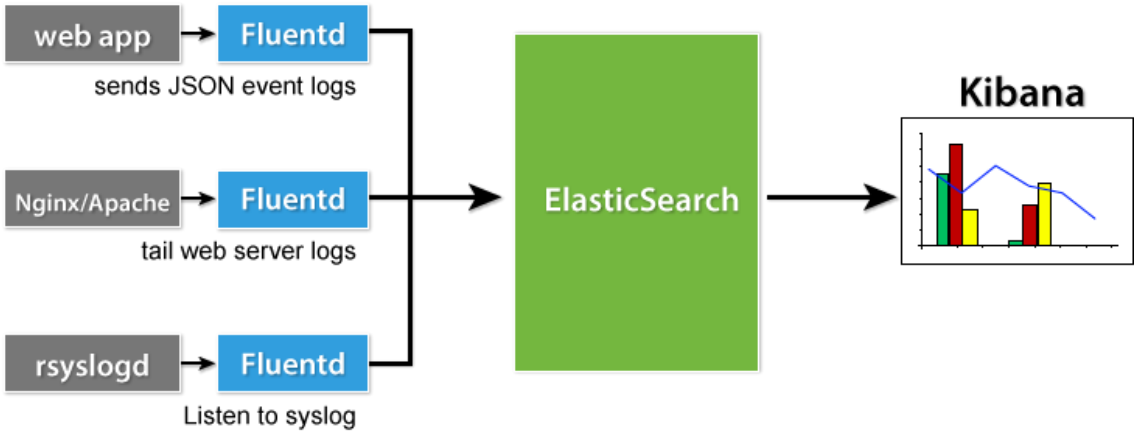
Arpalert ei kuitenkaan ollut varsinainen IDS-järjestelmä vaan sillä torjuttiin lähinnä huolimattomuudesta johtuneita virheitä ja varsinaiseen valvontaan tarvittiin jotain parempaa. Avoimeen lähdekoodiin perustuva Snort vastasi tähän tarpeeseen. Snort on niin sanottu NIDS-järjestelmä (Network Intrusion Detection System) ja se pyrkii tunnistamaan reaaliaikaisesti IP-liikenteestä sisältöä ja protokollia tutkien haitallista liikennettä. Snort toimii nykyään osana organisaation IDS-järjestelmää vastaten verkkoliikenteen seurannasta [43].

Pelkkä verkkoliikenteen valvonta ei kuitenkaan ole riittävää tietojärjestelmävalvonnan kannalta, sillä verkko on vain yksi osa tietojärjestelmää. Itse järjestelmien valvonta voidaan toteuttaa joko käyttäen standardeja protokollia kuten SNMP tai itse tehtyjen skriptien avulla tai asentamalla järjestelmään sen tilaa valvova ohjelma eli agentti, joka raportoi tuloksensa omaan järjestelmäänsä. Esitutkimuksen jälkeen organisaatiossa otettiin käyttöön Zabbix-järjestelmä. Zabbix pystyy valvomaan agentittomasti tai agentillisesti verkkolaitteita, palvelimia tai oikeastaan mitä vain laitetta, joka voidaan tavoittaa IP-verkon yli. Myös Snort:n hälytykset saadaan putkitettua Zabbixille, jolloin keskitetty valvonta saadaan toteutettua tehokkaasti. Zabbixiin voidaan määritellä monimutkaisiakin hälytyksiä riippuen eri järjestelmien antamasta reaaliaikaisesta, historiallisesta tai laskennallisesta tilannetiedosta [44].

4.2.2 Keskitetty lokitus

Edellisessä aliluvussa käsitelty valvonta keskittyi nimenomaan reaaliaikaiseen tietojärjestelmien valvontaan eli siihen mitä nyt tapahtuu. Lokit sen sijaan kertovat sen mitä tapahtui. Miltei kaikki nykyisin ajettavat palvelut ja järjestelmät generoivat lokia tapahtumistaan ja ne oletuksena tallennetaan kyseisen järjestelmän paikalliselle levylle. Kun järjestelmien määrä nousee lokien hallinnointi ja lukeminen käy hankalaksi tai jos kyseiseen järjestelmään liittyvä paikallinen tallennustila esimerkiksi laiterikon takia on menetetty häviävät samalla myös järjestelmään liittyvät lokit. Lisäksi useista sadoista lokitiedostoista halutun virheen etsiminen on todella työlästä ilman kunnon työkaluja.

Yleinen tapa on keskittää lokien kerääminen keskitetylle lokipalvelimelle. Esimerkki-organisaatiomme keskitti lokien keräämisen ja tutkimisen avoimen lähdekoodin ohjelmistoihin Fluentd, Elasticsearch ja Kibana, joilla saatiin organisaation käyttöön kaupallista Splunk-ohjelmistoa vastaava toiminallisuus. Kuvassa 22 on esitetty järjestelmän toiminallisuus periaatteellisella tasolla [45].



Kuva 22: Keskitetyn lokijärjestelmän toiminta periaatteellisella tasolla [45].

Kuvan 22 mukaisesti keskitettyyn palveluun saadaan kerättyä lokit verkkolaitteilta, palvelimilta ja käytännössä miltä tahansa palvelulta, joka saadaan lähettämään JSON-muotoista (engl. JavaScript Object Notation) dataa. Elasticsearch avulla lokeista saadaan etsittyä haluttu tieto ja Kibanan avulla lokeja voidaan visualisoida tarvittaessa.

4.3 Virtualisoinnin ja pilvipalveluiden käyttöönoton kehitys

Virtualisoinnin ja pilvipalveluiden käyttöönoton kehitys lähti esimerkki-organisaatiossamme liikkeelle yksittäisten ohjelmistokehittäjien tarpeista testata kätevästi ohjelmistojaan mahdollisella puhtaalla kohdekäyttöjärjestelmällä. Kakkostyypin hypervisoreina käytettiin sekä KVM:ää että Virtualboxia ja myöhemmin VMWare Workstationeita. Tarpeiden kasvaessa sekä testaustyön jakautuessa usealle kehittäjälle projekteille alettiin tarpeen vaatiessa perustaa omia VMWaren vSphere-ympäristön päälle pystytettyjä virtuaalihosteja. Samalla saatiin lisää luotettavuutta ympäristön siirtyessä yksittäisiltä työasemilta konesalin syövereihin ja nopeille peilatuille levyalustoil-

le. Samalla käytännössä koko lähiverkon palvelinympäristö virtualisoitiin ylläpidon helppouden ja toimintavarmuuden vuoksi.

VSphere-ympäristö toimi hyvin, mutta se oli kuitenkin kallis projektien testiympäristöjen käytettäväksi. Tarvittiin joku kevyempi ja halvempi ratkaisu, joten organisaatiolle pystytettiin oma yksityinen pilvipalvelualusta, jonka alustaksi valikoitui OpenStack. OpenStack mahdollisti organisaation projekteille itsepalveluperiaatteen mukaisen toiminnan tarpeiden mukaan. Projektit pystyivät ilman ylläpidon toimia saamaan käyttöönsä tarvittavan määrän virtuaalikoneita. OpenStack mahdollisti myös hybridipilvipalvelun sitomisen palveluun, jolloin projektit pystyivät lyhyisiin tarpeisiinsa kuten rasitustesteihin ostamaan suoritustehoa myös julkisesta pilvestä.

5 CASE-TUTKIMUKSET

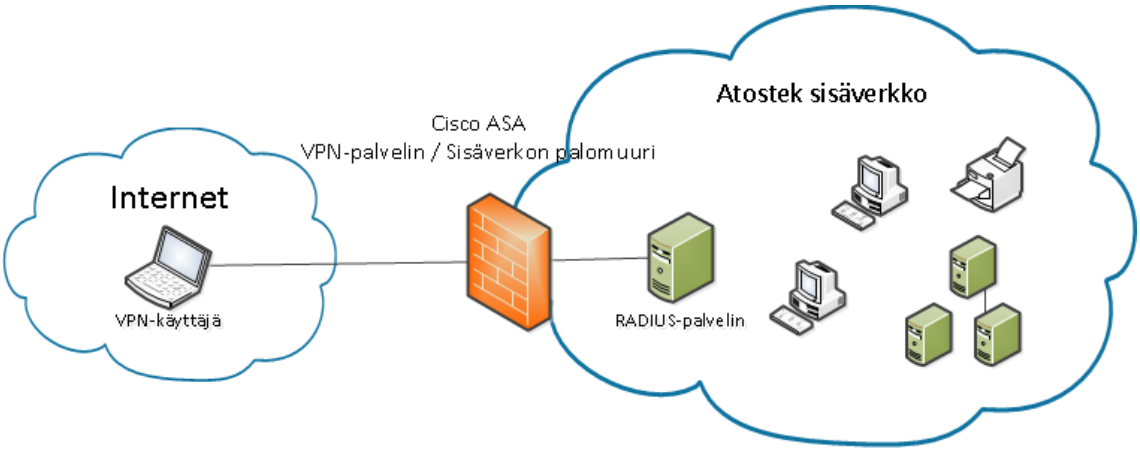
Tässä luvussa käydään läpi kaksi erilaista case-tutkimusta, jotka tehtiin diplomityön tilaajalle eli Atostek Oy:lle vuoden 2014 aikana. Atostek Oy on noin 50 hengen ohjelmistoalan yritys, joka on perustettu vuonna 1999. Yrityksessä ei ole vuosien aikana ollut erillistä henkilökuntaa ylläpitämässä verkkoa tai palvelimia vaan muutama ohjelmistosuunnittelija on hoitanut ylläpidollisia toimenpiteitä oman työnsä ohella vuoden 2012 joulukuuhun asti, jolloin palkattiin ensimmäinen päätoiminen ylläpitäjä. Varsinaisen ylläpidon puuttumisen johdosta myös varsinainen olemassa olevan verkon dokumentaatio on puutteellinen tai sitä ei käytännössä ollut ollenkaan. Seuraavissa aliluvuissa kuvatuissa caseista ensimmäisessä käydään läpi Atostek Oy:n VPN-palvelun päivitys kokonaisuudessaan. Toisessa casessa käydään läpi Atostek Oy:lle perustetun sisäisen pilvipalvelun esitutkimus ja pystytys. Molemmissa caseissa kuvataan ensin nykytila, käydään läpi syyt muutokselle, esitellään eri vaihtoehdot ja käydään läpi valittu vaihtoehto.

5.1 Case study 1: Atostekin VPN-päivitys

Tässä casessa on tarkoitus käydä läpi Atostek Oy:n työntekijöiden etäyhteydet mahdollistavan VPN-palvelun päivitys. Yrityksen kasvaessa varsinkin senior-tasolle päässeet ohjelmistosuunnittelijat viettävät usein suuren osan viikon työtunneista muualla kuin perinteisesti toimistolla työskennellen. Toimiva, turvallinen ja sujuva etäyhteyksien käyttö on noussut todella tärkeäksi osaksi yrityksen toimintaa. Aiemmissa luvuissa läpikäytyjä parhaita käytäntöjä sekä tietoturvan määritelmiä sovellamme nyt käytäntöön.

5.1.1 Nykytila

Nykyään Atostekilla on käytössä etäyhteyksiä varten olemassa oleva VPN. VPN-palvelimena toimii toimiston Cisco ASA palomuuri ja asiakasohjelmana toimii Cisco VPN-client. Käyttäjien todentaminen tapahtuu siten että jokaisella VPN-käyttäjällä on henkilökohtainen tunnus. Lisäksi käyttäjille on jaettu RSA:n valmistama SecurID-dongle, johon he ovat valinneet henkilökohtaisen pin-koodin. Donglelta saatu kertakäyttösalasanan ja PIN-koodin yhdistelmä muodostaa kertakäyttöisen vahvan tunnistamisen tunnusmerkit täyttävän tunnuslauseen. VPN-palvelin välittää tunnuksen ja tunnuslauseen RADIUS-palvelimelle (engl. Remote Authentication Dial In User Service), joka joko hylkää tai hyväksyy tunnistamisen VPN-palveluun. VPN-palvelun nykytila on kuvattu kuvassa 23.



Kuva 23: VPN-palvelun nykytila.

5.1.2 Syyt muutokselle

Suurin syy muutokselle on että nykyinen IPsec-pohjainen (engl. Internet Protocol Security) VPN-asiakasohjelmiston tuki on loppunut eikä siihen tule enää päivityksiä. Myöskään Microsoftin uusin käyttöjärjestelmä eli Windows 8 ei ole enää tuettu vanhassa asiakasohjelmistossa. Asiakasohjelmiston tekniikasta johtuen myöskään yrityksen käytössä olevien kannettavien tietokoneiden sisäänrakennetun mobiiliverkkomodeemin käyttö ei ole mahdollista VPN-yhteyksien muodostamiseen. Käyttäjämäärä oli myös rajattu eikä sitä voitu nostaa. Vanhalla järjestelmällä jatkaminen ei ollut mahdollista. Kuten kuvasta 23 havaitaan vanha järjestelmä sisältää myös heikoimman lenkin ongelman eli sekä VPN-palvelu että sisäverkon palomuuri sijaitsevat samalla laitteella – laitteen vikaantuessa sekä VPN-palvelu että sisäverkon yhteydet ulkomaailmaan ovat poissa käytöstä. Lopullisena liikkeelle panijana VPN-järjestelmän päivitykselle oli SecurID-donglejen sopimuskauden päättymisen läheneminen.

Käytännössä uuden VPN-järjestelmän pitää ratkaista kaksi ongelmaa. Ensinnäkin tarvitaan VPN-ohjelmisto, jonka asiakasohjelmistolla että palvelinohjelmistolla on riittävä tuki myös tulevaisuudessa. Lisäksi tunnistamista varten tarvitaan tapa, joka täyttää vahvan tunnistamisen tunnusmerkit johtuen sekä omista että erityisesti asiakkaiden etäyhteyksiä koskevista vaatimuksista.

5.1.3 Erilaiset vaihtoehdot

Alustavissa tutkimuksissa VPN-palvelun toteuttamiseen tuli esille useita erilaisia vaihtoehtoja. Vaihtoehdot luokiteltiin sen mukaan tarjoaako tuote pelkän VPN-ohjelmiston, ratkaisun vahvan tunnistamisen ongelmaan, vai molemmat.

VPN-ohjelmistovaihtoehtoja oli markkinoilla tarjolla suuri määrä ja varsinaiseksi harkittaviksi vaihtoehdoiksi nousivat lopulta Cisco Anyconnect, Microsoft Direct Access ja avoimen lähdekoodin OpenVPN. Vahvan-tunnistamisen ratkaisuiksi harkittavina vaihtoehtoina oli Yubico Yubikey, toimikortti sekä RSA SecurID.

Cisco Anyconnect on käytännössä nykyisen VPN:n korvannut järjestelmä. Erona on Anyconnectin käyttämä SSL/TLS-protokolla vanhemman ohjelmiston käyttäessä IPsec-protokollaa yhteyden salaukseen [32].

Microsoft Direct Access on Microsoftin Active Directory ympäristöön liittyvä VPN-ratkaisu. Direct Accessin kantavana ideana on että Direct Accessia käyttävä laite yhdistetään automaattisesti organisaation verkkoon. VPN-yhteyttä ei siis erikseen tarvitse muodostaa sisäverkon ulkopuolella toimiessa vaan käytännössä Direct Access käyttäjä toimii aina organisaation sisäverkosta käsin. Vahvaa tunnistamista varten Direct Accessin lisäksi Active Directory ympäristöön implementoitaisiin toimikortti kirjautuminen [33].

OpenVPN on avoimen lähdekoodin VPN-tuote. OpenVPN:n pohjana on käyttää SSL/TLS-protokollaa VPN-yhteyksien salaamiseen. OpenVPN:n asiakasohjelmisto on tarjolla miltei kaikille alustoille sen avoimuuden takia. OpenVPN:stä on tarjolla myös täysin ilmainen versio, mutta lopulta päädyttiin vaihtoehdoksi ottamaan OpenVPN:n tarjoama access server appliance sen helppouden ja ylläpitotyökalujen takia [34].

Yubicon Yubikey on käytännössä usb-liitäntöinen yhden painikkeen näppäimistö, joka toteuttaa kertakäyttösalasanan syötön nappia painamalla. Yubikey toteuttaa vahvan tunnistamisen ja se on käytössä laajalti niin suurissa kuin pienissä yrityksissä ja organisaatioissa ympäri maailmaa [35].

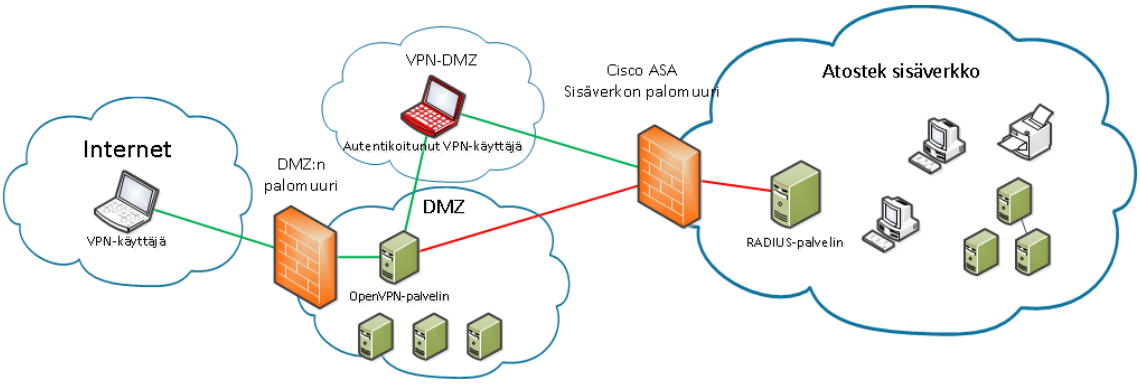
RSA SecurID dongle on laite, joka generoi laitteen näytölle uuden kuusi numeroisen numerosarjan tyypillisesti minuutin välein. Numerosarja generoidaan käyttäen algoritmia, joka saa syötteeksi jokaiseen dongleen erikseen kovakoodatun satunnaisen siemenluvun. Sama siemenluku on myös SecurID:hen kuuluvan RADIUS-palvelimen tiedossa, jota vastaan tällä luvulla lopulta autentikoidaan. SecurID toteuttaa vahvan tunnistamisen vaatimukset kun generoidun luvun lisäksi käytetään henkilökohtaista PIN-koodia [36].

Toimikortti (engl. smart card) on muovinen kortti, jolla on tallennettu tunnistamista varten käyttäjän henkilökohtainen PIN sekä sertifikaatti, jonka on luonut luotettu taho, johon myös VPN-palvelun tunnistuspalvelu luottaa. Toimikortin lukija lukee kortin ja kysyy käyttäjältä PIN-koodia, jolloin vahvan tunnistamisen vaatimukset toteutuvat [37].

Lisäksi harkittiin myös koko VPN-palvelun ja mahdollisesti myös ylläpidon ulkoistamista, mutta tämä todettiin hyvin nopeasti liian kalliiksi siihen nähden että tarvittava tietotaito löytyi nykyisellään myös organisaation sisältä palvelun pystyttämiseen ja ylläpitoon.

5.1.4 Valittu vaihtoehto

VPN-ohjelmiston valintaa tehtäessä aika uuden järjestelmän käyttöön saamiselle alkoi olla vähissä. VPN-ohjelmistoksi valittiin OpenVPN, koska siitä oli ylläpidolla jo valmiiksi kokemusta. Lisäksi kustannustehokkuus oli tälle ympäristölle OpenVPN:llä kaupallisia valmistajia parempi. Uusi järjestelmä on esitetty kuvassa 24.



Kuva 24: Uusi VPN-järjestelmä.

Kuvan 24 mukaisesti VPN-palvelin sijoitettiin verkon DMZ-alueelle, jolloin se ei suoraan ole yhteydessä julkiseen verkkoon vaan välissä on vielä erillinen palomuuri. Tunnistamisen yhteydessä OpenVPN-palvelin ottaa yhteyttä sisäverkossa sijaitsevaan RADIUS-palvelimeen, jota vasten VPN-käyttäjä tunnistetaan. Tunnistamisen yhteys on ilmaistu kuvassa punaisella. Tunnistamisen jälkeen VPN-käyttäjä liitetään osaksi OpenVPN-palvelimen sisällä olevaa VPN-DMZ-verkkoa, jolta voidaan liikennöidä tietyin rajoituksin sisäverkkoon. Tämä on ilmastu kuvassa 24 punaisella VPN-käyttäjällä.

Vahvan tunnistamisen ollessa vaatimuksena ja ajan loppuessa valittiin käyttöön RSA SecurID, sillä sitä varten tarvittava ympäristö oli jo vanhan järjestelmän ajalta pystyssä. SecurID ei ollut kustannuksiltaan pienin, mutta tässä tapauksessa uuteen järjestelmään siirtymisestä johtuva työaika oli niin lyhyt että kokonaiskustannukset pysyivät pieninä. Erään asiakkaan tiukoista vaatimuksista johtuen vahvan tunnistamisen muutokset olisivat myös voineet käynnistää tietoturvaan liittyvän auditoinnin, jota haluttiin välttää pitäytymällä vähintään samassa tunnistamistasossa kuin ennenkin. Lisäturvaominaisuutena uusi järjestelmä toi käyttöön käyttäjäkohtaiset sertifikaatit, jotka samalla helpottivat ylläpitäjien työtä manuaalisen asiakasohjelmistokonfiguroinnin vähenemisen muodossa. Lisäksi VPN-käyttäjien yhteydet verkkoon saatiin parempaan valvontaan, kun yhteyksiä ei enää oteta suoraan organisaation sisäverkon palomuurin kautta. VPN-palvelinohjelmiston siirryttyä pois sisäverkon palomuurista ratkaistiin myös VPN-palvelun ja sisäverkon palomuuriin liittynyt heikoimman lenkin ongelma, jossa jommankumman palvelun vikaantuessa tai vaatiessa suurempia konfiguraatiomuutoksia myös toinen palvelu olisi poissa käytössä.

5.2 Case study 2: Yksityinen pilvipalvelu toteutus Atostekin omaan käyttöön

Tässä casessa käydään läpi Atostek Oy:n omaan käyttöön tuleva yksityinen pilvipalvelu. Organisaation kasvaminen sekä projektien muuttuneet vaatimukset ovat johtaneet siihen että virtualisoituja palvelimia tarvitaan yhä enemmän ja yhä nopeammalla aikataululle. Asiakkaiden tiukkojen tietosuojavaatimusten takia julkisen pilven käyttö ei ole mahdollista suurimmassa osassa toimintaa. Organisaation sisäisellä yksityisellä pilvipalvelulla koitetaan tuoda ratkaisut lähemmäs tarvitsijaa eli projektia ja sen tekijöitä.

5.2.1 Nykytila

Nykyisellään Atostek Oy:llä ei ole käytössä omaa varsinaista pilvipalvelua, mutta sisäinen tietojärjestelmäympäristö on laajalti virtuaalisoitu. Käytännössä virtuaalikoneisiin liittyvät tarpeet ja toiveet vaativat aina projektin ja ylläpidon välistä kanssakäymistä.

5.2.2 Syyt muutokselle

Pilvipalvelu-konseptin on tarkoitus vähentää ylläpidon työtaakkaa hallinnan siirtyessä laitteistotasolta enemmän virtualisointitasolle eli käytännössä ylläpidolta projektien henkilöstölle. Pystytettävän pilvipalvelun tulee olla käytettävissä itsepalveluperiaatteella eli käytännössä tarjottavasta palvelusta on pystyttävä käynnistämään lisää virtuaalikoneita niitä tarvittaessa. Pilvipalvelun alla toimivien hypervisoreiden tulisi olla vapaasti valittavissa, jottei sitouduta liikaa yhteen valmistajaan. Lisäksi mahdollisuus ulkoisten pilvipalveluiden käyttöön oman pilven jatkeena olisi tervetullut ominaisuus, kun jossakin projektissa vaaditaan suhteellisen lyhyen aikaa todella suurta laskentatehoa.

5.2.3 Erilaiset vaihtoehdot

Alustavien tutkimusten jälkeen pilvipalvelualustaksi Atostekin omalle yksityiselle pilvelle löytyi avoimeen lähdekoodiin perustuvat OpenNebula ja OpenStack sekä VMWare vCloud, joista vCloud hylättiin ainakin konseptivaiheessa pois johtuen sen hinnasta sekä lukkiutumisesta liikaa yhteen valmistajaan.

Sekä OpenNebula että OpenStack ovat käytössä pilvialustoina nykypäivänä ympäri maailmaa ja niiden kehitys on aktiivista. Kumpikaan alustoista ei sisällä varsinaista virtualisointiympäristöä vaan niillä ohjataan ja hallinnoidaan eli orkestroidaan erilaisia hypervisioreita kuten VMWare, KVM tai Xen [20].

OpenNebula on avoimen lähdekoodin projekti, jonka tarkoituksena on rakentaa standardi pilvipalvelutyökalu suurien kompleksisten ja heterogeenisten ympäristöjen hallintaan. Sen avulla voidaan rakentaa niin yksityisiä, julkisia kuin myös hybridipilviä. OpenNebula myös tukee yleisimpiä hypervisioreita kuten Xen, KVM sekä VMWare. OpenNebulaa voidaan myös käyttää VMWaren oman pilvenhallintatuotteen vCloudin vaihtoehtona [20].

OpenStack on kokoelma alun perin Yhdysvaltain ilmailu- ja avaruushallinto NASA:n ja Rackspace nimisen yrityksen kehittämiä ohjelmistoja, jotka yhdistetty kokonaiseksi pilvialustaksi. OpenStackin tarkoituksena on olla yksinkertainen ja avoin ympäristö, jota erilaiset organisaatiot voivat käyttää omien pilvipalveluidensa pystyttämiseen. OpenStack tukee laajalla skaalalla erilaisia hypervisoreita. Lisäksi oman OpenStackin liittäminen hybridipilveksi esimerkiksi maailman suosituimman pilvipalvelun eli Amazon AWS:n kanssa on vaivatonta [20].

5.2.4 Valittu vaihtoehto

Toteutettavaksi yksityiseksi pilvipalvelukonseptiksi valittiin OpenStack. Valintaa puolsi erityisesti alkuvaiheessa tärkeänä pidetty nopea pystyyn saaminen, joka yksinkertaisimmillaan tapahtui käyttämällä OpenStackin kehittäjille tarkoitettua valmista pakettia eli DevStackia.

Yksityisen pilvipalvelun käyttöönotto jaettiin neljään eri vaiheeseen:

1. Atostekin ylläpito asentaa ja ottaa OpenStackin testikäyttöön johonkin omaan projektiinsa ja luo samalla tarvittavat virtuaalikoneiden levykuvat.
2. Ylläpidon avustuksella testikäyttöön valittu projekti pystyttää tarvittavat virtuaalikoneet OpenStackiin ja alkaa käyttää sitä osana normaalia toimintaansa.
3. Ylläpito standardoi pilvipalvelusta löytyvät levykuvat ja prosessit kuinka projektit ottavat palvelun käyttöönsä.
4. Uusien projektien alkaessa virtuaalikoneet toteutetaan itsepalveluna omaan pilveen.

Tällä hetkellä projektissa ollaan vaiheessa numero yksi. Atostekissa on meneillään useampia projekteja, jotka hyötyisivät palvelun käyttöönotosta. Mikään projekti ei kuitenkaan seiso tämän pilvipalvelu puuttumisen johdosta, joten palvelu voidaan pystyttää rauhassa ja kunnolla.

6 ARVIOINTI JA TULOKSET

Tässä luvussa käydään läpi työn tulokset ja arvioidaan aikaisemmissa luvuissa esiteltyjen osa-alueiden hyödyllisyyttä. Rakenteellisesti tämän luvun alaluvuissa käydään läpi tietoverkot, virtualisointi, pilvipalvelut ja tietoturvasta esitellyt asiat. Jokaista aihealuetta arvioidaan siltä kannalta että mitä hyötyä tilaajaorganisaatiolle on kyseisen aihealueen tutkimisen myötä esille tulleista asioista nyt ja mahdollisesti tulevaisuudessa. Toisena asiana arvioidaan sitä, helpottuuko ylläpitäjien työ aihealueen ratkaisuja hyödyntämällä.

6.1 Tietoverkot

Tietoverkkoihin liittyen työssä tutustuttiin tietoverkkojen rakenteeseen sekä arkkitehtuurimalleihin. Työn tilaajan ympäristö sijoittuu asiakasverkkoon, joten suuressa mittakaavassa tietoverkoista kerätyn tiedon käyttö jää matalalle tasolle. Tietoverkon hierarkista suunnittelumallia saatiin kuitenkin sovellettua myös tämän kokoiseen verkkoympäristöön. Verkkoympäristö rakennettiin uudelleen luhistuneen rungon mallina, joka toi lisää toimintavarmuutta verkkoon. Samalla verkko saatiin dokumentoitua, mikä säästää ylläpitäjiltä turhaa päänvaivaa muutoksia tehdessä tai mahdollista verkkovikaa selvittäessä.

Lehti-ranka-mallin hyödyntäminen tässä ympäristössä ei ole järkevää, mutta organisaation asiakkaiden konsultointityössä siitä voi olla hyötyä, sillä useisiin hankkeisiin liittyy nykyään myös konesalipalveluiden käyttö, joiden syvällisempi ymmärrys monesti asiakkaalta uupuu. Lisäksi diplomityön aikana kertyi lisätietoa myös paljon erilaisista verkkoprotokollista kuten STP, TRILL ja SPB.

Kokonaisuutena tietoverkoista saatiin diplomityön aikana hyvä kokonaiskuva, joka tulevaisuudessa auttaa organisaation omaan verkkoon ja mahdollisten asiakasprojektien verkkoihin liittyvässä työssä.

6.2 Virtualisointi

Virtualisointiin liittyen työssä käytiin läpi erilaisia virtualisointeja sekä virtualisointiin liittyviä konsepteja kuten SDDC. Tilaajan ympäristössä virtualisointi oli jo vahvasti käytössä ennen työn aloittamista. Ohjelmistoyrityksen kannalta saatiin irti kuitenkin paljon tietoa, sillä ohjelmistojen jakelu on siirtymässä enemmän ja enemmän pilvipalveluihin. Esimerkiksi sovelluskonttivirtualisointia päästiin kokeilemaan jo käytännössä muutaman ohjelmistosuunnittelijan voimin. Toisena esimerkkinä työpöytävirtualisoinnin tarkempi tarkastelu auttoi erästä asiakasprojektia eteenpäin. Palvelinvirtualisointi

sen sijaan oli jo niin laajalti käytössä että sen tutkimisesta saadut hyödyt menivät suoraan ylläpidon avuksi jokapäiväiseen työhön. Sen sijaan laajemman verkkovirtualisoinnin käyttöönoton ei ainakaan vielä todettu tuovan mitään parannusta ympäristöön. Organisaation kasvaessa ja verkkovirtualisoinnin tullessa paremmin tarjolle tekniikka koettiin mahdollisesti tulevaisuuden kannalta lupaavaksi. SDDC konseptiin tutustuminen antoi ylläpidolle suuntaviivoja siitä mihin suuntaan tallennustila-, verkko- ja virtualisointikehitystä tulisi viedä tulevaisuudessa.

Kokonaisuutena diplomityön aikana saatu tietämys virtualisoinnista koettiin hyödyllisenä. Tästä tietämyksestä saadaan organisaatiolle sekä lyhyen aikavälin että pitemmän aikavälin käytännön hyötyjä.

6.3 Pilvipalvelut

Pilvipalveluista työssä tutustuttiin erilaisiin pilvipalveluiden jakelu- ja palvelumalleihin. Tilaajan ympäristössä pilvipalveluita on ollut jonkun verran käytössä, mutta niitä ei ole otettu käyttöön laajamittaisesti lähinnä asiakkaiden tietosuoja- ja tietoturvavaatimusten takia. Myöskään kokonaiskuva erilaisista pilvipalveluista ei ollut alun perin kovin hyvin hallussa.

Diplomityön aikana kertyneestä tietämyksestä pilvipalveluihin liittyen tullaankin toivottavasti saamaan tulevaisuudessa irti paljon ja käyttöönotettavilla pilvipalveluilla saadaan tehostettua ja laskettua kustannuksia joissakin projekteissa. Ylläpidon kannalta prosesseja ja toimintaa on tarkasteltava uudelta kannalta, sillä arviointi esimerkiksi siitä, mitä julkiseen pilveen voidaan viedä, on tehtävä huolella. Pilvipalveluista kertynyt tieto myös käynnisti organisaatiossa projektin, jossa kokeillaan organisaation yksityisen pilvipalvelun käyttöönottoa.

Kokonaisuutena diplomityön aikana saatu tietämys erilaisista pilvipalveluista hälvensi ainakin osittain pelkoa pilvipalveluihin liittyen. Todennäköisesti organisaatio alkaakin hyödyntää pilvipalveluita jatkossa entistä enemmän ja tästä pitäisi seurata käytännön helpotusta ylläpidon työtaakkaan.

6.4 Tietoturva

Tietoturvasta diplomityössä käsiteltiin verkon tietoturvaa, identiteetin- ja pääsynhallintaa sekä valvontaa ja jäljitettävyyttä. Organisaatiossa tietoturvan taso oli kokonaisuutena hyvällä mallilla, mutta parannettavaakin löytyi. Diplomityön kanssa ajallisesti sattunut erään projektin tietoturva-auditointi tarjosi hyviä mahdollisuuksia hyödyntää myös tässä diplomityössä esille tulleita teorioita.

Verkon tietoturvassa esillä ollut verkon segmentointi oli organisaatiossa jo käytössä ja segmentointikäytäntöjä myös täydennettiin diplomityön aikana. Organisaation etäyhteyskäytännöt tarkistettiin ja päivitettiin ajan tasalle. Verkon tietoturva kokonaisuutena organisaatiossa oli diplomityössä kertyneiden havaintojen perusteella hyvällä tasolla.

Identiteetin- ja pääsynhallinnan tutkimisen tuloksista saatiin irti parannuksia hallinnolliseen tietoturvaan. Organisaatiossa suoritettiin rooliperusteisen pääsynhallintamallin mukainen roolien louhinta, josta välitön hyöty välittyi organisaation hallinnolle ja roolien pahin rämettyminen saatiin karsittua pois. Tästä hyötyi myös ylläpito, koska organisaatioon saatiin tarjolle selvemmät roolit ja ryhmät joilla pääsynhallintaa voidaan suorittaa. Identiteetin- ja pääsynhallinta oli suurilta osin jo keskitetty, mutta diplomityön aikana saatiin aikaan pieniä parannuksia esimerkiksi sitomalla Linux-ympäristö mukaan olemassa olemaan keskitettyyn käyttäjienhallintaan.

Valvonnan ja jäljitettävyyden kehitys diplomityön aikana oli ehkä suurin yksittäinen edistysaskel tilaajaorganisaation ympäristössä. Miltei kaikki laitteet ja palvelimet tuottivat jo aikaisemmin lokeja ja varoituksia, mutta niiden keskittäminen erilliseen järjestelmään toi hyötyjä ja tehokkuutta koko organisaatiolle. Ylläpidon työ helpottui siinä määrin, että esimerkiksi verkko-ongelmien ilmetessä ylläpito on yleensä jo tietoinen viasta ja käynnistänyt korjaustoimenpiteet ennen kuin varsinaiset käyttäjät huomaavat koko ongelmaa. Toisena uudistuksena lokien keskitetty kerääminen toi organisaatioon mahdollisuuden kokonaiskuvan helpompaan hahmottamiseen ja mahdollisti jäljitettävyyden.

Kokonaisuutena diplomityön aikana saatu tietämys tietoturvasta ja sen pohjalta suoritetut parannustyöt olivat organisaation kannalta suurin yksittäinen osa-alue, johon tämä diplomityö toi helpotusta. Identiteetin- ja pääsynhallinnan menetelmät auttoivat diplomityön aikana monia eri projekteja, organisaation hallintoa sekä yksinkertaistivat ja helpottivat ylläpitäjien töitä.

6.5 Tulevaisuuden näkymät

Tulevaisuudessa organisaation tulisi kehittää tietojärjestelmiään siten, että ylläpitotaakkaa saataisiin entisestään kevennettyä ja projekteille saataisiin niiden tarvitsemat palvelut toimitettua mahdollisimman ketterästi. Ketteryyden tulisi näkyä esimerkiksi siten että projekti voisi itsepalveluperiaatteen mukaisesti pystyttää omat tietojärjestelmänsä. Projektin alkaessa määriteltäisiin, voidaanko käyttää julkista pilveä vai organisaation omaa yksityistä pilveä, vai onko mahdollista tai hyödyllistä toteuttaa hybridipilviratkaisu. Tätä varten organisaation oma pilvipalveluprojekti täytyy lähitulevaisuudessa viedä maaliin asti. Myös erilaisten SaaS-palveluiden käyttöä projekteissa kannattaa suosia entistä enemmän, jos se vain on mahdollista.

Tietoturvan kannalta keskitettyä pääsyn- ja identiteetinhallintaa tulisi kehittää jatkossakin. Tätä palvelisi esimerkiksi federoidun tunnistautumisen käyttöönotto laajemmalti, jotta myös eri organisaatioiden piirissä olevat tunnukset saataisiin paremmin hallintaan. Hallinnollisen tietoturvan kannalta käyttöön voisi ottaa erilaisia vuosikelloja, jotta erilaiset tietoturvaan liittyvät toimet tulisi varmasti aina hoidettua.

7 YHTEENVETO

Tässä diplomityössä oli tarkoituksena saada ohjelmistoja tuottavalle konsulttiyritykselle lisää tietoa, jotta sen omia tietojärjestelmiä voitaisiin ylläpidon näkökulmasta kehittää paremmin. Diplomityön aikana tuli esille paljon teoriaa ja käytännön ratkaisuja niin verkkotekniikasta, virtualisoinnista, pilvipalveluista kuin myös tietoturvasta. Osa vastaan tulleista teknologioista tai tekniikoista koettiin vasta lapsenkengissä oleviksi tai muuten sopimattomiksi kyseiseen ympäristöön, mutta osa otettiin myös oikeasti käyttöön. Virtualisoinnin ja pilvipalveluiden kokonaiskuva selkeytyi selvästi organisaatiolle. Ohjelmistoyrityksen ydinosaamisen osalta virtualisointitutkimuksessa vastaan tullut sovellussäiliövirtualisointi koettiin hyödyllisenä ja luultavasti osa projekteista tuleekin hyödyntämään kyseistä tekniikkaa.

Diplomityön aikana jo valmiiksi käytössä ollut Microsoftin Active Directory otettiin laajemmin käyttöön, jolloin myös nykyiset ja tulevat Linux-koneet saatiin samojen keskitettyjen käyttäjätunnusten alle. Tämän lisäksi identiteetin- ja pääsynhallinnan toiminnoista ympäristössä suoritettiin myös johdon mukainen roolien louhinta. Tämä koettiin erityisesti hallinnon puolella hyödylliseksi, joten diplomityö ei tuottanut ainoastaan lisäarvoa ylläpidolle vaan myös hallinto hyötyi tähän työhön käytetystä työpanoksesta.

Diplomityön edetessä eräänä tärkeänä tavoitteena tunnistettiin tietoturvaan liittyvät vaatimukset ja niiden aiheuttama työmäärä ylläpidolle. Erityisesti tietoturvasta koettiin ylläpidon kannalta tarpeelliseksi identiteetin- ja pääsynhallinnan menetelmät sekä verkkoympäristön suojaaminen ja valvonta eli verkon tietoturva. Identiteetin- ja pääsynhallinnasta haluttiin entistä keskitetympää, sillä organisaation kasvaessa erillisten tunnusten hallinta esimerkiksi työntekijän poistuessa aiheutti ylimääräistä työtä. Lisäksi tietoturvaa ajatellen verkko- ja palvelinympäristön suojaaminen koettiin tärkeäksi. Tärkeimpänä ei ollut niinkään ulkopuolisten hyökkääjien torjuminen vaan tehokas ja toimiva verkko-ongelmien havaitseminen, tunnistaminen ja korjaaminen, sillä nykyään ohjelmistoyrityksen työnteko rakentuu vahvasti erilaisten verkkoyhteyksien takana olevien tietojärjestelmien ja palveluiden käyttöön.

Tässä diplomityössä ei pohdittu millaisia itse käytettävien tietojärjestelmien tai niiden sisällön tulisi olla vaan tarkoituksella keskityttiin ulkoisien edellytysten pohdiskeluun sekä kerättiin tietojärjestelmien ylläpitäjille tärkeää tietoa eri alueilta jatkokehitystä varten. Kokonaisuutena diplomityö kokosikin ylläpidolle tietotaitoa useista eri osa-alueista. Ohjelmistoyrityksen kannalta ylläpito tai IT-osasto on se taho minkä puoleen käännytään useasti, kun ohjelmistosuunnittelija joutuu oman ydinosaamisalueensa ulkopuolelle kuten hallinnollinen tietoturva, tietojärjestelmän pystyttäminen sekä verkko-

yhteyksiin liittyvät toimet tai ongelmat. Diplomityöstä karttunut tietotaito ei pelkästään anna mahdollisuuksia nykyisten toimintatapojen arviointiin ja kehittämiseen vaan yleensäkin mahdollistaa aivan uudenlaisten palveluiden tuottamisen organisaatiolle. Tulevaisuudessa ylläpidon tuleekin kehittää toimintaansa enemmän siihen suuntaan, että se standardoi toimintamallejaan ja onnistuu tuottamaan projekteille yhä enemmän palveluita itsepalveluperiaatteen mukaisesti. Tällöin ylläpidolle jää enemmän aikaa keskittyä oikeisiin ongelmiin, ja tähän suuntaan vietävällä kehityksellä myös diplomityölle asetetut tavoitteet saadaan täytettyä.

LÄHTEET

[1] Karri Huhtanen, Verkon suunnittelu 17.1.2008, Verkkotekniikan jatkokurssi 2008, TTY, viitattu 14.11.2013, [WWW]. Saatavissa: http://www.cs.tut.fi/~karrih/material/verkon-suunnittelu-muistiinpanoineen-current.pdf

[2] IP framework – A framework for convergence of telecommunications network and IP network technologies, ITU Recommendation Y.1001, International Telecommunication Union, 11/2000. Saatavissa: http://www.itu.int/rec/T-REC-Y.1001-200011-I/en

[3] Marja Saarikko, Pilvipalvelu säästää kustannuksia, 1/2011, viitattu 18.4.2014, [WWW]. Saatavissa: http://www.luonnontieteilijalehti.fi/artikkelit/2011/1/Pilvipalvelu_saastaa_kustannuksia

[4] Mark Dye, Rick McDonald, Antoon Rufi, Network Fundamentals: CCNA Exploration Companion Guide, Cisco Press. Saatavissa: http://ptgmedia.pearsoncmg.com/images/9781587132087/samplepages/1587132087.pdf

[5] Kotimaisten kielten keskus, lyhenneluettelo 2012, viitattu 8.5.2014, [WWW]. Saatavissa: http://www.kotus.fi/index.phtml?s=2149

[6] Understanding the Cloud Computing Stack: SaaS, PaaS, IaaS, Rackspace, viitattu 14.10.2014, [WWW]. Saatavissa: http://www.rackspace.com/knowledge_center/whitepaper/understanding-the-cloud-computing-stack-saas-paas-iaas

[7] Richard Kissel, NISTIR 7621. Small Business Information Security: The Fundamentals, National Institute of Standards and Technology, 2009. Saatavissa: http://csrc.nist.gov/publications/nistir/ir7621/nistir-7621.pdf

[8] Daniel Proch, Using OpenFlow protocol to control network flow, EE Times-Asia, 2011. Saatavissa: http://www.eetasia.com/STATIC/PDF/201202/EEOL_2012FEB17_NET_TA_01.pdf?SOURCES=DOWNLOAD

[9] Brian Underdahl, Robert Novak, Software Defined Data Centers (SDDC) for Dummies, Wiley Brand 2014. Saatavilla: https://www.nexenta.com/SDDCForDummies

[10] Jayshree Ullal, Evolutionary Designs for Cloud Networking, Arista Networks, viitattu 13.6.2014, [WWW]. Saatavissa: http://www.arista.com/blogs/?p=49

[11] Garret West, Cisco Spine and Leaf Architecture Discussion – Nexus 5500 vs 6001, 1.10.2013, viitattu 13.6.2014, [WWW]. Saatavissa: http://thenetworksurgeon.com/cisco-spine-and-leaf-architecture-discussion-nexus-5500-vs-6001/

[12] Massively Scalable Data Center (MSDC) Design and Implementation Guide, Cisco Press, 18.1.2013. Saatavissa: http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/MSDC/1-0/MSDC_Phase1.pdf

[13] Compare and Contrast SPB and TRILL, Avaya, 2011, viitattu 14.6.2014, [WWW]. Saatavissa: http://www.avaya.com/uk/resource/assets/whitepapers/SPB-TRILL_Compare_Contrast-DN4634.pdf

[14] Zhang Yandong, Zhang Yongsheng, Cloud computing and cloud security challenges, Information Technology in Medicine and Education (ITME), 2012 International Symposium on Hokodate, Hokkaido, Volume:2, IEEE 2012. Saatavissa: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6291488&queryText%3Dpublic+cloud+definition

[15] Jan Gabrielsson, Ola Hubertsson, Ignacio Más, Robert Skog, Cloud computing in telecommunications, Ericsson Review 1/2010. Saatavissa: http://www.ericsson.com/res/thecompany/docs/publications/ericsson_review/2010/cloudcomputing.pdf

[16] Brad Hedlund, Inverse Virtualization for Internet Scale Applications, 16.3.2011, viitattu 5.8.2014, [WWW]. Saatavissa: http://bradhedlund.com/2011/03/16/inverse-virtualization-for-internet-scale-applications/

[17] Jarmo Harju, Supermatrix Virtualizes Your Desktop Computer, Trex workshop 2010, viitattu 6.8.2014, [WWW]. Saatavissa: http://www.trex.fi/2010/Supermatrix_Virtualizes_Your_Desktop_Computer_-_Jarmo_Harju_170210.pdf

[18] Cisco Global Cloud Index: Forecast and Methodology, 2012–2017, Cisco 2013, viitattu 6.8.2014. Saatavissa: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.pdf

[19] Marcos Hernandez, The Future For Network Engineers, 27.6.2013, viitattu 6.8.2014, [WWW]. Saatavissa: http://blogs.vmware.com/networkvirtualization/2013/06/the-future-for-network-engineers.html

[20] Xiaolong Wen, Genqiang Gu, Qingchun Li, Yun Gao, Xuejie Zhang, Comparison of Open-Source Cloud Management Platforms: OpenStack and OpenNebula, 2012. Saatavissa: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6234218

[21] Jorge Carapinha, Peter Feil, Paul Weissmann, Saemundur E. Thorsteinsson, Cagri Etemoglu, Ólafur Ingpórsson, Selami Ciftci, Márcio Melo, Network Virtualization – Opportunities and Challenges for Operators, Future Internet – FIS 2010, Third Future Internet Symposium Berlin, Germany, September 20-22 2010. Saatavissa: http://link.springer.com/chapter/10.1007%2F978-3-642-15877-3_15

[22] Mikael Linden, Identiteetin- ja pääsynhallinta, luentomoniste, 2012. Saatavissa: http://www.cs.tut.fi/~linden/iam-pruju.pdf

[23] Heini Makkonen, Mummon verkkopankkiin ei ole muilla asiaa, Savon Sanomat 10.8.2014 s. 26.

[24] Risk Management Framework (RMF) OVERVIEW, NIST, viitattu 14.8.2014, [WWW]. Saatavissa: http://csrc.nist.gov/groups/SMA/fisma/framework.html

[25] J.Case, M.Fedor, M. Schoffstall, J. Davin, A Simple Network Management Protocol (SNMP), RFC 1157, IETF 5/1990. Saatavissa: https://www.ietf.org/rfc/rfc1157.txt

[26] Edmund Wong, Network Monitoring Fundamentals and Standards, 14.8.1997, viitattu 22.8.2014, [WWW]. Saatavissa: http://www.cse.wustl.edu/~jain/cis788-97/ftp/net_monitoring.pdf

[27] Suman Pandey, Mi-Jung Choi, Young J. Won, James Won-Ki Hong, SNMP-based enterprise IP network topology discovery, 20.5.2009, Int. J. Network Mgmt 2011, 21, s.169–184. Saatavissa: http://young.hanyang.ac.kr/blog/paper/2011.ijnm1.pdf

[28] Network Virtualization Using Shortest Path Bridging and IP/SPB, Avaya, 2013. Saatavissa: http://www.avaya.com/usa/documents/network_virtualization_using_spb_white_paper_-_white_paper.pdf

[29] Kapil Bakshi, Secure Hybrid Cloud Computing: Approaches and Use Cases, Cisco System 2014. Saatavissa: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6836198

[30] Prashant Shenoy, Distributed Operating Systems Fall 2009, lecture 5, UMass Computer Science, viitattu 27.8.2014, [WWW]. Saatavissa: http://lass.cs.umass.edu/~shenoy/courses/spring11/lectures/Lec05.pdf

[31] Andrew Lambeth, Shudong Zhou, Distributed Virtual Switch for Virtualized Computer Systems, VMWare 2008. Saatavissa: http://worldwide.espacenet.com/publicationDetails/biblio?CC=US&NR=2009292858A1&KC=A1&FT=D

[32] Cisco Anyconnect VPN Client product documentation, viitattu 28.8.2014. [WWW]. Saatavissa: http://www.cisco.com/c/en/us/support/security/anyconnect-vpn-client/tsd-products-support-series-home.html

[33] Microsoft Direct Access Overview, Microsoft Corporation, 2012, viitattu 28.8.2014, [WWW]. Saatavissa: http://technet.microsoft.com/en-us/library/dd759144.aspx

[34] OpenVPN Access Server Overview, OpenVPN Technologies Inc, 2014, viitattu 28.8.2014, [WWW]. Saatavissa: https://openvpn.net/index.php/access-server/overview.html

[35] YubiKey Security Evaluation: Discussion of security properties and best practices, Yubico Inc, viitattu 28.8.2014, [WWW]. Saatavissa: https://www.yubico.com/wp-content/uploads/2012/10/Security-Evaluation-v2.0.1.pdf

[36] RSA SecurID, EMC^2, viitattu 28.8.2014, [WWW]. Saatavissa: http://finland.emc.com/security/rsa-securid.htm

[37] Toimikortit terveydenhuollolle ja julkishallinnolle, Väestörekisterikeskus, viitattu 28.8.2014, [WWW]. Saatavissa: http://www.fineid.fi/julkaisut/VRK_toimikortit/VRK_Toimikortit.pdf

[38] Patrick Galbraith, Docker and Ansible: Container management made easy, luentokalvot 2012, viitattu 1.10.2014. Saatavissa: http://www.slideshare.net/PatrickGalbraith/docker-ansible-34909080

[39] TLT-3301 Verkon tietoturva kevät 2011, Luento 3. Luentokalvot. TTY. Viitattu 1.10.2014, [WWW]. Saatavissa: https://moodle.tut.fi/file.php/2882/luennot-2011/TLT-3301-2011-luento3.pdf

[40] Understanding IPS and IDS: Using IPS and IDS together for Defense in Dept, SANS Institute 2004. Saatavissa: http://www.sans.org/reading-room/whitepapers/detection/understanding-ips-ids-ips-ids-defense-in-depth-1381

[41] TLT-3301 Verkon tietoturva kevät 2011, Luento 9, Luentokalvot, TTY, viitattu 2.10.2014, [WWW]. Saatavissa: https://moodle.tut.fi/file.php/2882/luennot-2011/TLT-3301-2011-luento9.pdf

[42] ArpAlert, viitattu 7.10.2014, [WWW]. Saatavissa: http://www.arpalert.org/arpalert.html

[43] Snort, viitattu 7.10.2014, [WWW], Saatavissa: https://www.snort.org/

[44] Zabbix, viitattu 7.10.2014, [WWW]. Saatavissa: http://www.zabbix.com

[45] Fluentd, Fluentd project, viitattu 7.10.2014, [WWW]. Saatavissa: http://www.fluentd.org/