

Palvelun saatavuuden varmistaminen monitoimittajaympäristössä

Tero Strakh

Tampereen yliopisto

Informaatiotieteiden yksikkö

Tietojenkäsittelyoppi

Pro gradu –tutkielma

Ohjaaja: Mikko Ruohonen

Huhtikuu 2015

Tampereen yliopisto

Informaatiotieteiden yksikkö

Tietojenkäsittelyoppi

Tero Strakh: Palvelun saatavuuden varmistaminen monitoimittajaympäristössä

Pro gradu -tutkielma, 52 sivua, 8 liitesivua

Huhtikuu 2015

Yhä useampi yritys on siirtymässä homogeenisestä keskitetysti toimitetusta siilomaisten järjestelmien arkkitehtuurista palvelupohjaiseen (SOA, Service Oriented Architecture) palveluna hankittuun arkkitehtuuriin. Palvelut ovat hajautettuina useille eri toimittajille, jotka yhteistyössä monitoimittajaympäristössä tuottavat asiakasyritykselle heidän tarvitsemansa IT-palvelut. Homogeenisessä arkkitehtuurissa palvelun toimitusketjun häiriöiden paikallistaminen on yksinkertaista. Toipumis- ja häiriönhallinnan prosessit ovat yleisesti yhden käyttöpalvelutoimittajan koordinoitavissa.

Monitoimittajaympäristössä toipumis- ja häiriönhallinnan prosessit ovat hajautettu eri toimijoille ja yrityksen oma käyttöpalvelutoimittaja koordinoi ainoastaan yrityksen omistamia räätälöityjä järjestelmiä sekä palveluita. Koska monitoimittajaympäristö on hajautettu eri toimijoiden hallinnoitavaksi, on häiriötilanteiden selvittäminen haastavampaa kuin homogeenisessä arkkitehtuurissa.

Tässä tutkielmassa selvitetään kirjallisuuskatsauksen sekä tapaustutkimuksen kautta eri vaihtoehtoja monitoimittajaympäristön häiriöiden paikallistamiseen, korjaamiseen ja ennakointiin. Sekä tutkitaan erään yrityksen olemassa olevaa monitoimittajaympäristöä ja tämän häiriönhallinnan ratkaisua. Tutkielman tavoitteena oli selvittää millaisella integraatioarkkitehtuurilla voidaan valvoa ja ennakoita palvelun saatavuuteen vaikuttavia häiriöitä. Tutkielman lopputuloksen tarkoituksena ei ole vähentää esiintyvien häiriöiden määrää vaan varmistaa palvelun saatavuus loppukäyttäjille ja näin tukea liiketoimintaa.

**Avainsanat ja -sanonnat:** ITIL, monitoimittajaympäristö, häiriönhallinta, järjestelmäintegraatio, palvelukeskeinen arkkitehtuuri, SOA, liiketoimintaprosessien hallinta, BPM, palveluväylä.

## ALKUSANAT

Tämä Pro gradu -tutkielma on tehty opinnäytetyöksi Tampereen yliopiston tietojenkäsittelytieteisiin. Haluaisin kiittää kaikkia, jotka ovat olleet myötävaikuttamassa pro gradu -tutkielmani valmistumiseen.

Haluan kiittää työni ohjaajaa, professori Mikko Ruohosta asiantuntevasta ohjauksesta, kommenteista sekä lähdevinkeistä.

Lisäksi haluaisin kiittää professori Pirkko Nykästä alustavan suunnitelman tekemisestä ja ohjeista työni aloittamisen suhteen.

Kiitos kuuluu myös vaimolleni, pojalleni, sekä minun että vaimoni vanhemmille läpi opintojeni jatkuneesta korvaamattomasta tuesta ja kannustuksesta. Erityisesti haluan kiittää vaimoani Annaa tuesta sekä kannustuksesta työni tekemisessä.

Tampereella 21.4.2015

Tero Strakh

## Sisällys

1. Johdanto .....	1
2. Tutkimus .....	3
2.1 Tutkimusongelma .....	3
2.2 Tutkimusmetodi ja tutkimuksen rajaus .....	3
3. Järjestelmäintegraatio ja monitoimittajaympäristö lyhyesti .....	6
3.1 Järjestelmäintegraation syntyhistoria .....	6
3.2 Mikä järjestelmäintegraatio on? .....	7
3.2.1 Integraation hyödyt .....	7
3.2.2 Yleisimmät integraatiotavat .....	8
3.3 Monitoimittajaympäristö .....	11
3.3.1 Monitoimittajaympäristön hyödyt .....	12
3.3.2 Monitoimittajaympäristön haasteet .....	12
4. Häiriönhallinta monitoimittajaympäristössä .....	13
4.1 Häiriönhallintaan ja monitoimittajaympäristöön liittyviä yhteisiä haasteita .....	13
5. Palvelun laatu .....	14
5.1 Palvelun laatu tietojärjestelmien onnistumisen näkökulmasta .....	15
6. Palvelukeskeinen arkkitehtuuri .....	18
6.1 Palvelukeskeisen arkkitehtuurin piirteet .....	19
6.2 Palvelukeskeisyys .....	20
6.3 Palveluväylä .....	20
6.3.1 Palveluväylän tehtävät .....	21
6.3.2 Palveluväylän käytön hyödyt .....	22
7. Liiketoimintaprosessit .....	23
7.1 Liiketoimintaprosessien hallinta .....	23
7.2 BPM:n hyödyt .....	24

7.3	SOA:n ja BPM:n yhdistäminen .....	25
8.	Tapaustutkimus – integraatiopalveluita sekä konsultointia tarjoava yritys .....	27
8.1	Yrityksen alkutila .....	27
8.1.1	Aiemman arkkitehtuurin vaikutus yrityksen liiketoimintaan.....	28
8.1.2	Aiempaan arkkitehtuurin liittyvät ongelmat .....	28
8.1.3	Monitoimittajaympäristöön liittyvät ongelmat .....	28
8.2	Yrityksen tavoittila .....	30
8.2.1	Liiketoiminnan tavoitteet .....	32
8.3	Siirtymävaihe uuteen arkkitehtuuriin eli migraatio.....	32
8.3.1	SOA-Valvonta.....	33
8.3.2	ICT-Häiriönhallintaratkaisu .....	33
8.3.3	Portaaliratkaisu.....	34
8.4	Toteutunut arkkitehtuuri.....	35
8.4.1	Palveluarkkitehtuuri .....	35
8.4.2	Palvelutaso, muutoksenhallinta ja ennakoitua huoltotoimenpiteet.....	38
8.4.3	Tiedottaminen palveluketjussa.....	39
8.4.4	Palveluiden valvonta ja hallintamalli .....	41
8.4.5	Yhtenäinen CMDB .....	42
8.4.6	Palvelukeskeinen arkkitehtuuri keskiössä.....	43
8.5	Mitä häiriönhallinnan arkkitehtuurilla saatiin aikaan?.....	43
8.6	Ennakoimattomat vaikutukset ja mahdolliset parannusehdotukset.....	44
9.	Yhteenveto.....	47
	Viiteluettelo.....	49
	Liitteet .....	53
	Taulukko 1. Monitoimittajaympäristön haasteet ja niiden ratkaisut .....	53
	Taulukko 2. Tyypillisiä ICT-palvelujen laatumääreitä .....	55
	Taulukko 3. Palvelukeskeisyyden periaatteet .....	58

Taulukko 4. ESB:n tärkeimmät ominaisuudet .....	59
Kuva 17. Projektissa toteutunut valvonta-arkkitehtuuri.....	60

## Sanasto

**Actional.** Ohjelmistoyhtiö Progress:n kehittämä integraatioiden monitorointiin soveltuja agenttiperusteinen työkalu, jonka avulla yksittäisen pyynnön kulku palvelukeskeisen arkkitehtuurin prosessissa on mahdollista selvittää.

**BPM (Business Process Management).** Liiketoimintaprosessien hallinta on yrityksen operatiivisen hallinnan osa-alue, joka keskittyy yrityksen liiketoimintaprosessien kehittämiseen sekä optimointiin.

**CMDB (Configuration Management Database).** Konfiguraatietietokanta, joka sisältää IT-yrityksen komponenttien väliset konfiguraatitiedot CI:t.

**CI (Configuration Item).** Konfiguraatietietokannan tietue, joka voi olla esimerkiksi dokumentaatio, tieto ohjelmistosta, prosessien ja palveluiden kuvaus, säännöstö tai reititystieto.

**ESB (Enterprise Service Bus).** Arkkitehtuurimalli, jonka avulla järjestelmät keskustelevat keskenään palvelukeskeisessä arkkitehtuurissa.

**Integraatio.** Vähintään kahden järjestelmän liittäminen toisiinsa, esimerkiksi rajapintojen avulla.

**ITIL (Information Technology Infrastructure Library).** Globaalisti jo 20 vuotta tunnustettu kokoelma erilaisia käytäntöjä IT-palveluiden hallintaan ja johtamiseen.

**Legacy-järjestelmä.** Vanha järjestelmä, joka pystyy edelleen suorittamaan tehtävänsä, mutta on toteutettu vanhanaikaisella tekniikalla eikä sitä välttämättä pysty päivittämään tai se ei olisi kustannustehokasta.

**Oracle BAM (Business Activity Monitoring).** Ohjelmistoyhtiö Oraclen työkalu, jolla voidaan valvoa reaaliajassa liiketoimintaprosesseja. BAM mahdollistaa erilaisten hälytysrajojen ja –sääntöjen asettamisen valvonnan parantamiseksi.

**Palvelutasosopimus, SLA (Service Level Agreement).** Asiakkaan ja palveluntarjoajan välinen sopimus, jossa on määriteltyinä palvelulle halutut vaatimustasot. Palvelutasoa mitataan erilaisin mittarein ja sen alittamisesta seuraa sopimuksessa kuvattu sanktio.

**Rajapinta** (API, Application Programming Interface). Määritelmä, jonka kautta eri järjestelmät voivat vaihtaa tietoja keskenään, esimerkiksi XML-viestien avulla.

**SOA** (Service-Oriented Architecture). Palvelukeskeinen arkkitehtuuri, löyhään sidokseen perustuva ohjelmistoarkkitehtuurien suunnittelumalli.

**XML** (Extensible Markup Language). Rakenteellinen kuvauskieli, jota käytetään formaattina eri järjestelmien kommunikointiin sekä dokumenttien kuvaukseen.



## 1. Johdanto

Eri tietojärjestelmien yhteistoiminnallisuutta ja integrointia ajatellaan usein synonyymeinä. Termit usein sekoittuvat keskenään, koska ajatukset niiden taustalla ovat hyvin lähellä toisiaan.

Tietojärjestelmien yhteistoiminnallisuudesta voidaan puhua silloin, kun eri tietojärjestelmät kykenevät vaihtamaan informaatiota keskenään ja osaavat hyödyntää vaihdettua informaatiota [Geraci *et al.*, 1991]. Tietojärjestelmien integrointi tarkoittaa tiivistä liitosta tietojärjestelmien prosessien välillä [Pollock, 2001]. Integraatiossa tietojärjestelmät kykenevät vaihtamaan ja välittämään dataa, viestejä sekä tapahtumia keskenään. Järjestelmäintegraatio käsitteenä antaa mielikuvan yksittäisestä tuotteesta tai sovelluksesta, mutta järjestelmäintegraatio on todellisuudessa kokonaisuus, joka muodostuu erilaisista ajatus- ja suunnittelumalleista [Tähtinen, 2005].

Yrityksillä on käytössään useita itsenäisiä tietojärjestelmiä, joita on kehitetty erilaisiin tarpeisiin ja eri konteksteihin. Samaan aikaan on olemassa tarve kehittää uusia, usein yhä monimutkaisempia tietojärjestelmiä, jotka ainakin osaltaan perustuvat olemassa olevien sovellusten hyödyntämiseen. Järjestelmien käyttämän informaation integroimiseksi on yrityksissä tehty paljon töitä, jonka ansioista eri ohjelmistojen on mahdollista jakaa käyttämäänsä dataa keskenään. XML-kielestä on muodostunut standardi väline jakaa dataa ohjelmistojen välillä. XML-kielen suosio perustuu sen soveltuvuuteen rakenteisen data esittämiseen. XML-kieltä on jo pitkään hyödynnetty järjestelmien yhteisenä kielenä informaation siirtoon heterogeenisten tietojärjestelmien välillä. Vaikka XML:n avulla voidaan ratkaista syntaktinen yhteensopivuus tietojärjestelmien välillä, tarvitaan usein ihmisen ymmärtämystä päättämään esimerkiksi käytetäänkö kahta tietokannan kenttää samassa semanttisessa merkityksessä.

Tässä tutkielmassa tuodaan esille yksi ratkaisu monitoimittajaympäristön ja sen integraatioiden häiriöiden selvittämiseen sekä niistä toipumiseen. Tutkielmassa haluttiin selvittää onko tällainen häiriöiden käsittely ylipäättään mahdollinen ja millaisia ratkaisuja kirjallisuudesta löytyy toteutettavaan integraatiomalliin. Tämän lisäksi haluttiin selvittää, tukeeko tällainen ratkaisu liiketoiminnan kasvua sekä lisääkö se yrityksen tehokkuutta ja kilpailukykyä. Tämä tutkielma koostuu kahdesta osasta, kirjallisuuskatsauksesta ja tapaustutkimuksesta.

Tutkielman kirjallisuuskatsaus koostuu luvuista 3. – 7. Kolmannessa luvussa käsitellään järjestelmäintegraatioiden historiaa sekä esitellään erityyppisiä integrointitapoja. Lisäksi luvussa

käsitellään integraatioiden hyötyjä yrityksille sekä monitoimittajaympäristöä. Neljäs luku käsittelee ITIL:n mukaista häiriönhallintaa monitoimittajaympäristössä. Viidennessä luvussa käsitellään palvelun laatua käsitteenä sekä sen muodostumista. Kuudennessa luvussa käydään läpi palvelukeskeisen arkkitehtuurin toteuttaminen ja rakenne. Seitsemäs luku käsittelee liiketoimintaprosesseja, niiden hallintaa sekä palvelukeskeisen arkkitehtuurin ja liiketoimintaprosessien hallinnan yhdistämistä sekä tästä saatavaa hyötyä. Kahdeksas luku keskittyy tutkielman tapaustutkimukseen, joka käsittelee palvelukeskeisen arkkitehtuurin, liiketoimintaprosessien sekä häiriönhallinnan haasteita monitoimittajaympäristössä yrityksen näkökulmasta. Tapaustutkimuksessa käydään läpi yrityksen häiriönhallinnan tavoitetila, lähtötila, migraatio sekä lopputila. Tapaustutkimus koostuu yrityksen häiriönhallinnan migraation jälkeisestä lopputuloksesta, johon peilataan tavoitteita ja tarkastellaan saavutettiin ko järjestelmälle asetetut tavoitteet. Tutkimuksen kohteeksi rajataan ainoastaan ohjelmistopuolen häiriönhallinta. Yhdeksänten lukuun on kirjattu lyhyt yhteenveto tutkielman teemoista ja tapaustutkimuksesta. Tutkielman lähdeaineisto koostuu pääosin Google Scholar, ACM- ja IEEE –tietokannoista, alan kirjallisuudesta sekä tieteellisistä artikkeleista ja konferenssijulkaisuista.

## 2. Tutkimus

### 2.1 Tutkimusongelma

Järjestelmäintegraatiot ovat lisääntyneet viimeisen kymmenen vuoden aikana huomattavasti, koska yritysten tietojärjestelmät ja tieto ovat hyvin hajautettua. Järjestelmäintegraation avulla voidaan yhdistää vanhat legacy-järjestelmät ja uudemmat tietojärjestelmät yhteen. Tämä mahdollistaa reaaliaikaisen- sekä on-demand tiedonsiirron. Järjestelmäintegraatio voidaan toteuttaa omana integraatiojärjestelmänään tai pilvipalveluna tilattuna. Integraatiojärjestelmä soveltuu parhaiten suurten yritysten toimenkuvaan korkeiden kustannustensa vuoksi. Pilvipalveluna tilattu integraatiojärjestelmä on kustannustehokkaampi ja yritys voi tilata sen toiselta toimittajalta. Yleisesti sekä integraatiojärjestelmissä että pilvipalveluissa on monia eri toimijoita. Palvelimet ovat usein eri toimijan hallinnassa, verkot ovat eri toimijan ylläpidossa ja integraation toteutus ja ylläpito toisen toimijan hallinnassa. Tällaista hajautettua järjestelmää kutsutaan monitoimittajaympäristöksi. Tutkimuksen aiheena on ITIL:n (Information Technology Infrastructure Library) monitoimittajaympäristön häiriönhallinta IT-yrityksessä ja sen ratkaisun arviointi. Tutkimukseni lähtökohtana on selvittää kuinka erään yrityksen jatkuva palvelun saatavuus voitaisiin varmistaa monitoimittajaympäristössä ja kuinka palvelu voidaan asiakkaalle varmistaa ilman, että asiakas itse huomaa häiriön syntyä ja ilmoittaa siitä toimittajalle. Varsinainen tutkimukseni pyrkii vastaamaan, onko mahdollista toteuttaa nykyaikaisilla työkaluilla reaaliaikaista palveluiden valvontaa niin, että häiriöt ovat reaaliajassa tunnistettavissa.

### 2.2 Tutkimusmetodi ja tutkimuksen rajaus

Tutkielman tutkimusmenetelmänä käytetään sekä tapaustutkimusta, koska olin mukana tutkimuksen teon aikana täysin uudenaikaisessa ja merkittävässä integraatiohankkeessa. Tapaustutkimusta voidaankin ajatella tutkimusstrategiana, koska se pystyy antamaan vastauksen siihen mitä tutkitaan [Cassel and Symon, 2004]. Tämä tutkimusmenetelmä sopii erinomaisesti tutkimusaiheeseen, josta on olemassa vähän empiiristä tutkimusta [Cassel and Symon, 2004]. Tapaustutkimus tuottaa yleisesti paljon muuttujia, mutta hyvin vähän havaintoarvoja. Tästä syystä tilastollisia ja kvantitatiivisia päätelmiä on vaikea tehdä [Järvinen ja Järvinen, 1996].

Tutkimusmetodiksi valikoitui reflektioiva käytäntö (Reflective practice) [Schön, 1983]. Koska osallistuin itse integraatioprojektin toteutukseen, oli luontaista reflektoida projektin aikana

tapahtuneita asioita ja peilata kokemuksia tuleviin tilanteisiin. Refleктоivan käytännön keskeisenä ajatuksena on, että uuden tiedon oppimista hankitaan käytännössä sekä peilaamalla tuloksia tai kokemuksia tuleviin tapahtumiin [Schön, 1983]. Refleктоivan käytännön käyttäminen tutkimusmetodinä dynaamisessa ympäristössä oli loogista, koska se tukee oppimista, uuden omaksumista sekä ymmärrystä ilman rajoittavampia tutkimusasetelmien määrittelyjä [Schön, 1983].

Tapaustutkimus on luonteeltaan empiirinen tutkimustapa, jonka tarkoituksena on tutkia reaaliaikaisia ilmiöitä niiden todellisessa kontekstissaan [Yin, 1989]. Tutkimustavalle on tyypillistä, että ilmiön ja kontekstin yhteinen rajapinta ei ole täysin selvä ja käytettävissä olevia lähteitä on lukuisia.

Tapaustutkimus keskittyy tarkastelemaan nykyajan ilmiötä todellisessa käyttöympäristössä.

Yin [1989] puolestaan toteaa, että tapaustutkimus on empiirinen tutkimustapa, jossa tutkitaan reaaliaikaisia ilmiöitä niiden todellisessa kontekstissa. Tästä johtuen tapaustutkimukselle on tyypillistä, että ilmiön ja kontekstin rajapinta ei ole selvä ja käytettävissä olevia tiedon lähteitä on paljon. Tapaustutkimuksessa siis tutkitaan nykyajan ilmiötä sen todellisessa käyttöympäristössä. Yin [1989] myös erottelee tapaustutkimuksen muista tutkimusotteista tutkimuksen kysymyksen muodon, tutkittavien käyttäytymisen kontrollin ja ajanhetken perusteella. Jotta tapaustutkimus olisi riittävän laadukasta, on keskeistä kerätä riittävästi laadukasta tietoa [Yin, 1989]. Tässä tutkielmassa on hyödynnetty projektin aikana muodostuneita dokumentteja, osallistuvalla havainnoinnilla sekä projektin jäsenten haastatteluilla.

Tämän tutkielman tapaustutkimuksessa hyödynnetään suunnittelutieteellisen tutkimusotteen peruseriaatteita. Suunnittelutieteellisen tutkimuksen ajatuksena on pyrkiä tuottamaan hyödyllisiä toteutuksia todellisiin liiketoiminnan ongelmiin [Hevner *et al*, 2004]. Suunnittelutieteellinen tutkimus kuuluu innovaation hyödyllisyyttä painottaviin tutkimuksiin. Tutkimuksessa rakennetaan ja arvioidaan innovaatio, jolla pyritään ratkaisemaan jokin kohdeympäristön ongelma.

Tutkimuksella on mahdollista tuottaa uutta tietämystä, jota voidaan hyödyntää suunnittelu- ja konstruointiongelmien ratkaisemisessa [Järvinen ja Järvinen, 2004].

March ja Smith [1995] luokittelevat neljä toimintoa suunnittelutieteelliselle tutkimukselle: *rakentaminen*, *arviointi*, *teorian luonti* ja *teorian testaus*, sekä neljä tuotosta: *konstrukt*, *malli*, *menetelmä* ja *toteutus*. Tämän tutkielman tapaustutkimuksessa rakennetaan artefakti, jonka toiminnan hyvyttä arvioidaan soveltuvilla välineillä. Yksi tärkeimmistä asioista, on tarkastella miksi rakennettu artefakti toimii tai ei toimi, koska sen pohjalta on mahdollista rakentaa ja testata uusia teorioita. Tutkimuksella pyritään luomaan uusia innovaatioita ja artefakteja. Näillä

artefakteilla pyritään ratkaisemaan kohdeympäristön ongelma, esimerkiksi huonosti toimiva tietojärjestelmä, jota kehittämällä tehostettaisiin yrityksen toimintaa [Järvinen ja Järvinen, 2004].

Soveltamalla suunnittelutieteellistä tutkimusta, voidaan parantaa oikeaa liiketoimintaympäristöä sekä kasvattaa tietämyskantaa tarkastelua ja arvioinnin kautta. Tutkimuksena perustana ovat teoriat, viitekehykset, käsitteistöt, työkalut, mallit, metodit ja toteutukset. Työvälineet voidaan ajatella tutkimuksen teon apuvälineinä, kuten analyyseinä, tekniikoina, mittareina sekä arviointikriteereinä [Hevner *et al.*, 2004].

### 3. Järjestelmäintegraatio ja monitoimittajaympäristö lyhyesti

#### 3.1 Järjestelmäintegraation syntyhistoria

Järjestelmäintegraatio syntyi tiettävästi 1950–1960-lukujen vaihteessa, jolloin yritysten tietojärjestelmät jouduttiin räätälöimään käyttötarkoitukseensa sopiviksi. Järjestelmätoteutusten ollessa ei-yhteensopivia, törmättiin vaikeuksiin tilanteissa, joissa järjestelmien olisi pitänyt kyetä kommunikoidaan keskenään [Tähtinen, 2005]. Ensimmäinen askel kohti järjestelmäintegraatiota voidaan katsoa alkaneen kahdesta suuresta projektista, jotka olivat nimeltään SAGE (Semi-Automatic Ground Environment) ja SABRE (Semi-Automatic Business Research Environment) [Tähtinen, 2005]. SAGE oli Yhdysvaltojen vuonna 1949 alkaneen selvitystyön myötä syntynyt ilmavoimien projekti, jonka päämääränä oli Yhdysvaltojen ilmailuväyläjärjestelmän automatisointi kylmän sodan tarpeisiin. Projektin reaaliaikaiseen tiedonkäsittelyyn käytettiin 250 tonnia painavaa IBM:n (International Business Machines) kehittämää AN FSQ-7-tietokonetta, jonka ohjelmiston parissa työskenteli yli 700 ohjelmoijaa. Kun SAGE projekti valmistui, reaaliaikaisesti kommunikoidavia ilmailuväylätietokoneita asennettiin yhteensä 23 kappaletta eri puolille Yhdysvaltoja [Tähtinen, 2005]. SABRE –projekti sai alkunsa American Airlines –lentoyhtiön tarpeesta hoitaa lentojen paikanvaraus ja lipunmyynti reaaliajassa, jotta päällekkäisvaraukset eivät olisi mahdollisia. Tämä projekti vaati yli kymmenen vuoden suunnittelu- ja kehitystyön ennen käyttöönottoa vuonna 1964.

Toinen järjestelmäintegraatioiden syntyyn vaikuttanut tekijä syntyi kun Neuvostoliitto laukaisi vuonna 1957 ensimmäisen avaruusraketin, Sputnikin, avaruuteen. Yhdysvallat reagoi raketin laukaisuun ja perusti seuraavana vuonna ARPA-projektin (Advanced Research Project Agency). ARPA-projektin tarkoituksena oli edesauttaa korkean teknologian kehitystä. Projektin yhtenä tuloksena syntyi vuonna 1969 ARPANET-kommunikaatiojärjestelmä. Aluksi se oli Yhdysvaltojen puolustusministeriön ja yliopistojen käyttämä tutkimus- ja yliopistoverkko, joka laajentui vuosi vuodelta. Puolustusministeriö alkoi huolestua turvallisuusriskeistä ja siitä johtuen ARPANET jaettiin käyttötarkoituksiensa mukaan kanteen osaan, MILNETiin ja ARPAINETiin. MILNET oli ainoastaan puolustusviranomaisten ja ARPA-INTERNET yliopistojen käytössä. Vuonna 1990 aloitettiin ARPA-INTERNETin yksityistäminen, mikä johti Internetin syntymiseen. WWW:n (Word Wide Web) myötä ja yksittäisten tietokoneiden yleistyttyä Internet kohtasi

räjähdysmäisen kasvun. Yleisesti voidaan todeta, että Internet on muodostunut maailman suurimmaksi järjestelmäintegraatioksi.

Monitoimittajaympäristö muodostuu useista eri toimijoiden IT-palveluista, jotka yhteistyössä tuottavat ja tukevat asiakkaan tarvitsemaa IT-palvelua, josta muodostuu yhtenäinen järjestelmäintegraatioprojekti [Bartolini *et al.*, 2010]. Monitoimittajaympäristön toimittajat ovat vastuussa oman erikoisalansa palvelusta tai sen osasta ja ovat jatkuvassa yhteistyössä palveluiden suunnittelussa, muotoilussa ja toimittamisessa asiakkaalle. Asiakas on vastuussa monitoimittajaympäristön koordinoinnista ja palveluiden kokoamisesta, mutta monitoimittajaympäristön rakentaminen voidaan ulkoistaa kolmannelle osapuolella, joka toimii palveluintegroijana [Bartolini *et al.*, 2012].

### 3.2 Mikä järjestelmäintegraatio on?

Integraatiokäsitteestä ei ole yhtä ainoaa yleisesti hyväksyttävää määritelmää, sillä se liittyy moniin eri yhteyksiin. Uuden suomen kielen sanakirjan (2007) mukaan integraatio tarkoittaa käsitteitä: yhdentyminen, eheyttäminen, integroiminen. Matematiikassa sillä tarkoitetaan integraalin laskemista. Sosiologiassa integraatio liitetään yhteiskunnan rakennekokonaisuuksiin. Lisäksi se on liitetty politiikkaan, talousasioihin, aluekäsitteisiin, rotukysymyksiin, maatalouden ekosysteemeihin. Nykypäivän integraatio on valikoima teknologioita ja toimintatapoja, joiden avulla muutoin ei-yhteensopivat tietotekniset sovellukset saadaan kommunikoimaan keskenään [Tähtinen, 2005]. Suppea määrittely kertoo vain teknisen näkökulman integraatioon. Integraation merkitys voidaan nähdä koko organisaation ydintoiminnan kehittämisen työkaluna.

Järjestelmien integroimiseen on olemassa lukuisia eri tapoja ja kaikkia järjestelmiä ei ole alun perin suunniteltu jakamaan informaatiota, jolloin niiden liittäminen integraatioon saattaa olla haastavaa. Yleisimpiä tapoja järjestelmien välisiin integraatioihin ovat siirtotiedostot, tietokantayhteydet, etäproseduurikutsut (RPC, Remote Procedure Call) sekä sanomat eri järjestelmien välillä. Nykyisin suosituin tapa tietojärjestelmäintegraatioissa on sanomapohjaiset ratkaisut, koska ne mahdollistavat etäkommunikoinnin, tiedonsiirron ja luotettavan sekä helpon tavan integroida järjestelmät yhteen.

#### 3.2.1 Integraation hyödyt

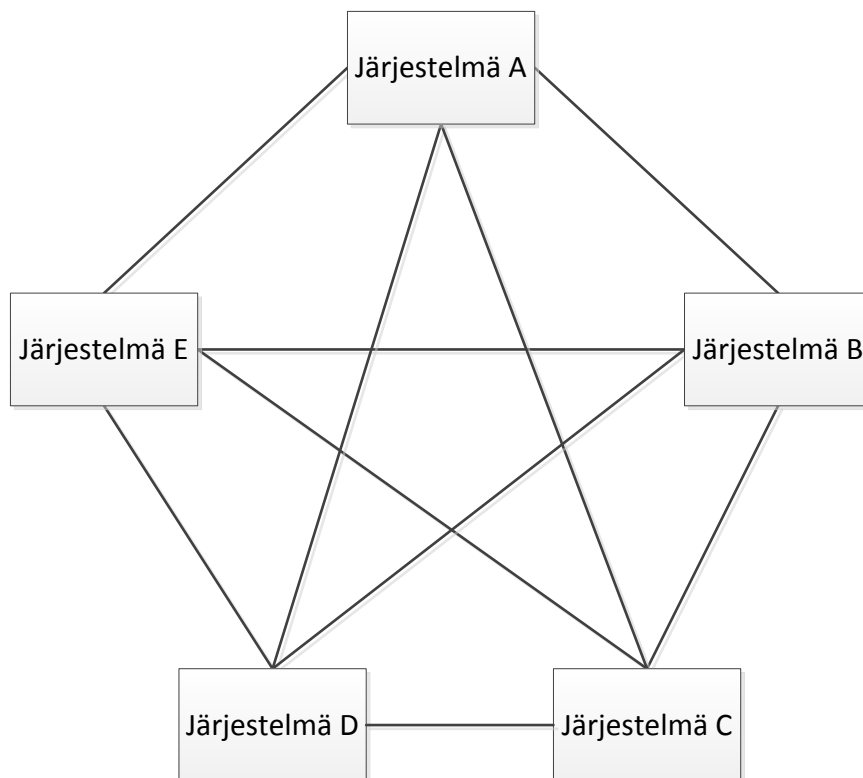
Metcalfen laki kuvaa tietojärjestelmän ja sen integroinnin hyödyllisyyttä. Lain mukaan verkon arvoksi muodostuu verkon käyttäjien lukumäärä neliöön korotettuna. Tämän seurauksena on

mahdollista päätellä, että esimerkiksi tietojärjestelmäverkko, jolla on kymmenen erillistä käyttäjää, on 13 kertaa arvokkaampi kuin verkko, jossa on neljä käyttäjää [Tähtinen, 2005].

Integraatioarkkitehtuurin suunnittelulla on mahdollisuus liiketoimintaprosessin sekä organisaation muutokseen. Hyvin suunnitellulla integraatioarkkitehtuurilla voidaan vähentää yrityksen riippuvuutta ohjelmistotoimittajista, joka mahdollistaa monitoimittajaympäristön luomisen ja sen tuomat edut [Tähtinen, 2005]. Monitoimittajaympäristön kuvaus sekä hyödyt on kuvattu tarkemmin luvussa 3.3.

### 3.2.2 Yleisimmät integraatiotavat

Integraatioiden toteuttamiseen on olemassa lukuisia tyylejä, joista yleisin on Point-to-point (kuva 1.) eli pisteestä pisteeseen integrointi tai Siltayhteistoiminta-malli. Point-to-point integraatiomallissa tiedon siirtäminen tapahtuu jonoilla (silta), joiden tehtävänä on siirtää tietoa järjestelmästä toiseen. Tällä tavalla integraation toteuttaminen on suoraviivainen, nopea ja helppo [Josuttis, 2007]. Point-to-point integraatiomallin heikkous piilee sen hallittavuudessa jos integraatiomallissa käytetään useampaa kuin kolmea järjestelmää, koska yhdistettävien järjestelmien määrän kasvaessa yli kolmen, yhteyksien määrä kasvaa vastaavasti neliöllisesti kun uusia järjestelmiä liitetään integraatiomalliin [Tähtinen, 2005].

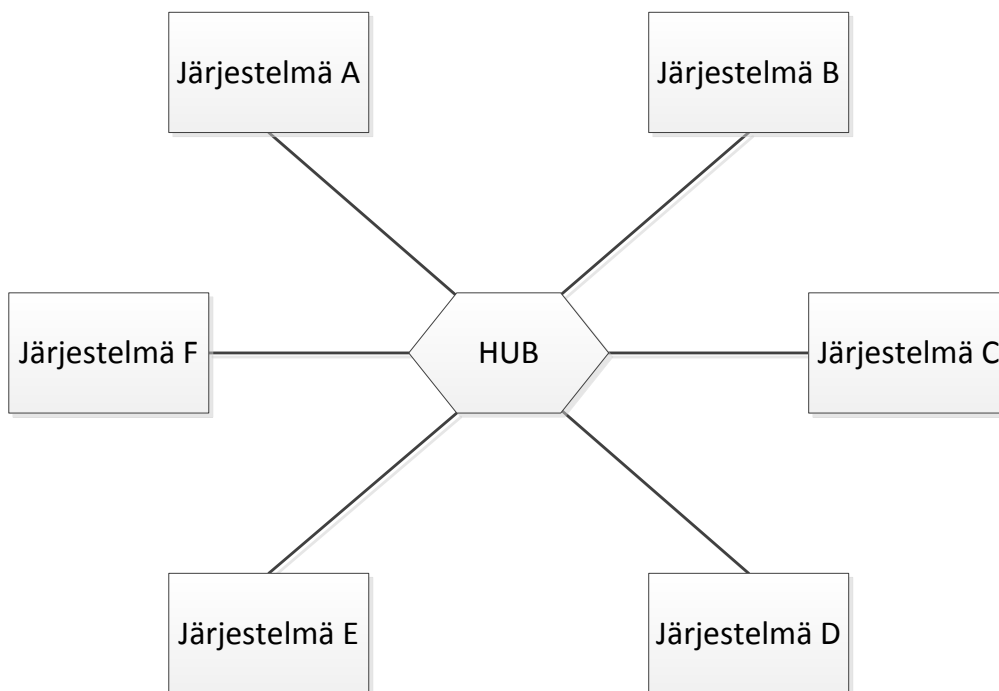


Kuva 1. Point-to-point integraatiomalli



*Point-to-point* –mallissa on hankaluutena myös tiedonsiirron automatisoiminen, koska jokainen järjestelmä käyttää omaa “kieltään”. Jotta järjestelmät voisivat suoraan keskustella keskenään, tarvitaan jokaista yhteyttä varten omia koodinmuuntimiaan [Mykkänen *et al.*, 2003]. Point-to-point integraatiomalli on hankalasti ylläpidettävä ja häiriöiden selvittäminen vaikeutuu. Koska Point-to-point integraatiomalli toimii usein tekemällä skriptejä sekä ohjelmapätkiä, on integraatiomallin keskitetty valvonta ja ylläpito mahdotonta. Integraatiomalli ei ole lainkaan skaalautuva, joka tekee uusien järjestelmien liittämisen integraatiomalliin mahdottomaksi.

*Hub-and-Spoke* –mallissa (kuva 2.) kaikki järjestelmät ovat kytköksissä keskitettyyn integraatiojärjestelmään, joka suorittaa viestin muunnokset ja viestien välittämisen järjestelmien välillä. Integraatiomalli oli vuosituhanen vaihteessa suosituin integraatoratkaisu. Hub-and-spoke integraatiomallia kutsutaan puhelinkeskusmalliksi keskitetyn integraatiojärjestelmän vuoksi, joka toimii samaan tapaan kuin puhelinkeskuksen hoitaja. Tässä integraatiomallissa on mahdollisuus useiden tietovirtojen hallintaan ja järjestelmän toiminnan valvontaan [Bussler, 2010].



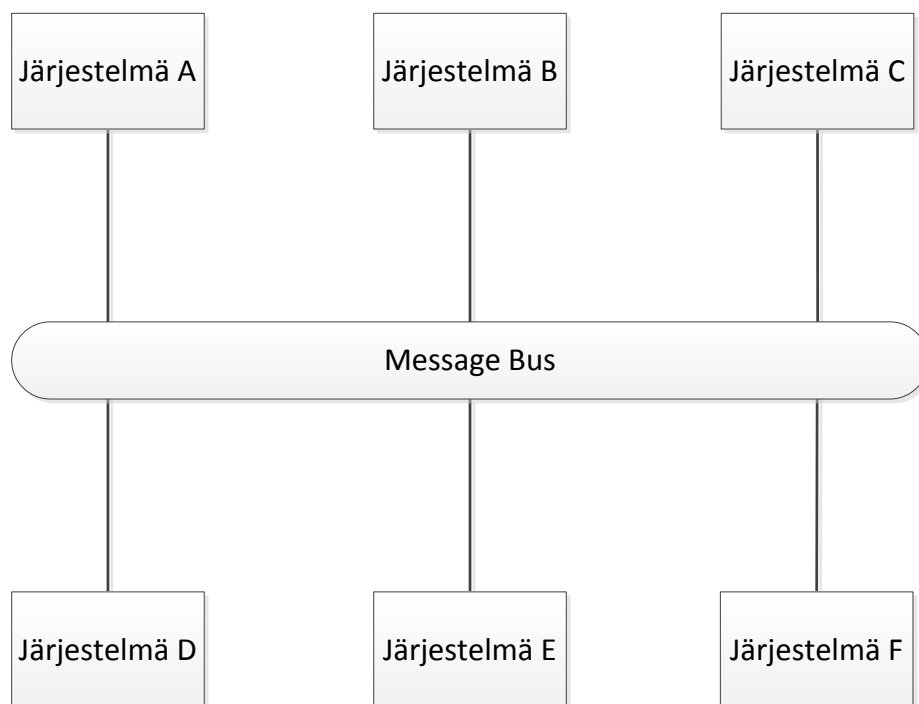
Kuva 2. Hub-and-spoke –integraatiomalli

Hub-and-spoke oli ensimmäinen integraatiomalli, jossa uuden järjestelmän lisääminen oli helppoa keskitetyn integraatiojärjestelmän vuoksi. Lisäksi arkkitehtuuri mahdollisti vanhempien

järjestelmien poistamisen tai korvaamisen koskematta toisiin järjestelmiin. Tällä tavalla voitiin hallita sitä, millaiset viestit ja kenellä on oikeus käyttää näitä resursseja [Tähtinen, 2005].

Hub-and-spoke mallin heikkoutena on sen haavoittuvuus, koska kaikki integraatiot on luotu keskitetyn järjestelmän kautta. Jos keskitetty järjestelmä hajoaa tai ulkopuolinen toimija pääsee siihen käsiksi, vaarantuu kaikki integraatioissa olevat järjestelmät ja niiden lähettämät viestit. Arkkitehtuurin keskusjärjestelmää kutsutaan single point of failureksi, koska yksi piste rikkoutuessaan katkaisee kommunikaation muiden järjestelmien väliltä [Hohpe, 2012]. Yritykset usein kahdensivat näitä integraatiomalleja, jotta viestinvälitys ei katkeaisi vaikka toisen kahdennetun ympäristön keskusjärjestelmä kaatuisi tai joutuisi hyökkäyksen kohteeksi. Tämä kuitenkin lisää huomattavasti integraatiomallin kustannuksia, koska kahdennetut ympäristöt ovat usein sijoiteltuina eri keskuspalvelimille.

Arkkitehtuurin heikkoutena voidaan myös ajatella sen anonymiutta, koska lähettävä ja vastaanottava järjestelmä eivät ole tietoisia toisistaan. Jos eri järjestelmien datat ovat ristiriidassa keskenään, on vastaanottavan järjestelmän mahdoton päätellä, kumpi data on relevanttia järjestelmälle.



Kuva 3. Message Bus –integraatiomalli

*Message Bus* (Kuva 3.) eli palveluväylä –integraatiomalli on nykyaikaisin järjestelmä, johon on helppo lisätä ja poistaa järjestelmiä. Palveluväylä on hajautettu malli, jolloin Hub-and-spoke –integraatiomallin suorituskyky- ja skaalautuvuusongelmat on helppo välttää. Hajautetussa palveluväylässä järjestelmät integroidaan palveluväylään ja palvelut ovat piilossa omien rajapintojen takana. Koska väylä mahdollistaa järjestelmien välisen sekä toimittaja -välisen riippumattomuuden, kutsutaan tällaista mallia monitoimittajaympäristöksi. Palveluväylässä integraatioiden toteuttaminen ja työmäärät ovat alhaisemmat kuin missään muussa integraatiomallissa [Trowbridge *et al.*, 2004]. Palveluväylä perustuu kanoniseen tietomalliin, jossa eri järjestelmien sanomat muutetaan väylän omaan formaattiin. Tällöin rajapintojen määrittely ja toteutus on helpompaa, koska kaikki järjestelmät käyttävät samaa kanonista tietomallia viesteissään.

Palveluväylä muistuttaa tekniseltä rakenteeltaan Enterprise Service Bus (ESB) –mallia, mutta ESB:n sisältyy muutakin kuin ainoastaan väylä viestien välitykseen järjestelmien välillä.

### 3.3 Monitoimittajaympäristö

Integraatioprojekteissa hyödynnetään usein montaa eri toimijaa, jotka koostuvat asiakkaasta, integraatiotoimittajasta sekä useista muista järjestelmätoimittajista ja muista tahoista, kuten esimerkiksi viranomaisista. Monitoimittajaympäristöllä tarkoitetaan sellaista tilannetta, jossa yritys on hankkinut IT-palveluita tai niiden osia useilta eri alan toimijoilta, jotka yhteistyössä muodostavat ja tuottavat asiakkaalle hänen tarvitsemansa IT-Palvelut [Bartolini *et al.*, 2012] [Bapna *et al.*, 2010]. Jokainen monitoimittajaympäristön toimittaja on vastuussa yhdestä tai useasta palvelusta tai palvelun osasta. Integraatioprojektissa toimittajat yhteistyössä vastaavat palveluiden suunnittelusta, muotoilusta ja toimittamisesta asiakkaalle. Tämän koordinointi on usein ulkoistettu kolmannelle osapuolelle, joka toimii samalla palveluintegroijana [Bartolini *et al.*, 2012].

Monitoimittajaympäristön ero perinteisiin kahden yrityksen välisiin ulkoistamisjärjestelyihin on, että useiden eri toimittajien tehtävät ovat toisistaan riippuvaisia ja toisten toimittajien työpanos vaikuttaa yksittäiselle toimittajalle ulkoistettuun tehtävän lopputulokseen [Bapna *et al.*, 2010]. Ilman toimittajien välistä riippuvuutta, monitoimittajaympäristö toimisi kuten perinteinen yhden toimittajan ulkoistamisjärjestely [Bapna *et al.*, 2010]. Tällaisten riippuvuuksien vuoksi toimittajat ovat pakotettuja panostamaan omaan tehtäväänsä sekä yhteistyöhön muiden toimittajien kanssa, jotta asiakkaalle saataisiin toimitettua saumaton ja integroitu palvelu monitoimittajaympäristössä.

### 3.3.1 Monitoimittajaympäristön hyödyt

Monitoimittajaympäristö tuo mukanaan monia etuja asiakasyritykselle ja parantaa samalla yrityksen kilpailukykyä. Yritykselle on mahdollista jakaa riskejä ulkoisten toimittajien kautta ja samalla kilpailuttaa palveluiden tuottamiseen kohdistuvia kuluja [Cohen and Young, 2006]. Yritykselle on mahdollista päästä asiantuntijaksi omalla sektorillaan ja helpottaa asiakasyritystä sulautumaan jatkuvasti kehittyviin toimialaedelleyksiin [Bapna *et al.*, 2010].

IT-palveluiden aiheuttamat kulut ovat yksi suurimmista yritysten maksuista ja yritys pyrkii vastaamaan vaatimukseen tehostamalla IT-palveluiden aiheuttamia kuluja, hankkimalla palveluita useilta ulkopuolisilta toimittajilta [Buxman and Kaiser, 2012]. Kun yritys ulkoistaa osan IT-palveluistaan ja tekee yhteistyötä usean eri toimittajan kanssa, on todennäköisempää että yritys pääsee osalliseksi uusiin liiketoimintamahdollisuuksiin sekä saavuttaa korkeamman suorituskyvyn [Bartolini *et al.*, 2012].

### 3.3.2 Monitoimittajaympäristön haasteet

Monitoimittajaympäristöön ja sen hallintaan tai johtamiseen liittyy paljon erilaisia haasteita. Monitoimittajaympäristössä odotetaan hyvää johtamista, koska asiakassuhteet ylittävät yritysraajat. Lisäksi haasteeksi muodostuu luottamuksen rakentaminen eri kilpailevien toimittajien ja asiakasyrityksen välille [Bartolini *et al.*, 2012]. Eri toimittajille ulkoistettujen tehtävien keskinäisestä riippuvuudesta johtuen, suhteiden johtamisesta tulee erityisen haastavaa [Bapna *et al.*, 2010]. Eri toimittajien motivoiminen nousee keskeiseksi tekijäksi asiakasyrityksen etua ajatellessa. Tällöin toimittajia kannustetaan tekemään yhteistyötä muiden toimittajien kanssa sekä auttamaan toisia toimittajia suoriutumaan omista tehtävistään. Kannustimien asettamista hankaloittaa toimittajien työpanosten arviointi, koska toisistaan riippuvaisten tehtävien laatua on vaikea havaita toimittajakohtaisesti [Bapna *et al.*, 2010].

Monitoimittajaympäristössä voi olla hankalaa tunnistaa ongelmien syytä, koska tehtävät ovat usein toisistaan riippuvaisia. Esimerkiksi yhden toimittajan työpanoksen taso voi olla pienempi kuin toisen toimittajan, jolloin syntyy tilanne, että asiakasyrityksen on hankalaa tunnistaa vastuullista toimittajaa joukosta. Monitoimittajaympäristössä tuloksen laatuun vaikuttaa usean eri toimittajan työpanos toisin kuin kahdenvälisessä ulkoistamisessa, jossa toisen toimittajan työpanos on suoraan suhteessa työtuloksen laatuun [Bapna *et al.*, 2010]. Taulukossa 1. on kuvattu monitoimittajaympäristön haasteita ja mahdollisia ratkaisuja näihin haasteisiin.

## 4. Häiriönhallinta monitoimittajaympäristössä

Nykypäivän yrityksen IT-palveluille on tärkeää palveluiden toiminnan jatkuvuus. Information Technology Infrastructure Library (ITIL) kuvaa häiriönhallintaa prosessina, jonka tarkoituksena on palauttaa järjestelmän toiminta normaaliksi häiriötilanteen jälkeen [ITIL, 2011]. ITIL sisältää kattavan kokoelman käytäntöjä IT-palveluiden hallintaan ja sitä pidetään IT-palveluiden hallinnan de facto –standardina [Bartolini et al., 2010].

### 4.1 Häiriönhallintaan ja monitoimittajaympäristöön liittyviä yhteisiä haasteita

Koska monitoimittajaympäristö koostuu useista eri toimittajien toiminnoista, on näiden koordinointi hankalaa. Lisäksi eri osapuolilla on usein puutteelliset tiedot ulkoistettujen tehtävien välisistä keskinäisistä riippuvuuksista. Häiriönhallinta monitoimittajaympäristössä on haastavaa, koska ulkoistettujen tehtävien välisiä riippuvuuksia on vaikea hahmottaa ja siten häiriöiden syiden on vaikea löytää. Häiriöiden alkuperän selvittäminen vaikeutuu, koska vastuu on usein jaettuna usealle eri toimittajalle ja työmäärät eri toimittajien välillä vaihtelevat suuresti [Bapna *et al.* 2010]. Tästä syystä palvelun palauttaminen normaaliin toimivaan tilaan saattaa pitkittyä merkittävästi.

Ulkoistettujen tehtävien keskinäinen riippuvuus vaikuttaa häiriönhallintaprosessin haastavuuteen myös siksi, että organisaatioiden eri komponenttien riippuvuus saattaa olla hyvin tiukkaa, jolloin normaalin toiminnan palauttaminen häiriötilanteen jälkeen on haastavaa. Jos komponentit ovat löyhemmin keskenään riippuvaisia, on organisaatiolla paremmat mahdollisuudet palautua häiriöstä ja löyhempi riippuvuus komponenttien välillä helpottaa vaikutussuhteiden ja siten häiriöiden ymmärtämistä [Bapna *et al.*, 2010].

Sekä monitoimittajaympäristö että häiriönhallintaprosessi ovat monimutkaisia, joka tekee häiriönhallinnasta monitoimittajaympäristössä haastavaa. Organisaation monimutkaisuus vaikuttaa implementoidun häiriönhallintastrategian ja sen suorituskyvyn arviointiin sekä ymmärtämiseen. Monimutkaisuudesta johtuen asiakasyrityksen tulisi tehdä yhteistyötä prosessitasolta saakka, jolloin kaikki osapuolet voidaan ottaa mukaan häiriönhallintaprosessin suunnitteluun ja toteuttamiseen.

## 5. Palvelun laatu

Palvelun laatu käsitteenä on hyvin haastava, koska sitä on erittäin vaikea mitata tai määritellä [Wisniewski, 2001]. ICT-palvelujen laatua pidetään subjektiivisena asiana, jonka mittarina usein on asiakas- tai käyttäjätyytyväisyys. Asiakas voi olla tyytymätön palvelun laatuun, vaikka palvelua koskevat palvelutasotavoitteet täytyisivätkin ja vaikka palvelun tuotannon laatumittarit olisivat hyvällä tasolla. Subjektiivisen asiakkaan tai käyttäjän laatumäkömyksen lisäksi ICT-palvelujen laatu voidaan jakaa sovittuihin palvelutasoihin ja ICT-palvelujen tuottamista koskevaan laatuun. Edellisten lisäksi laatuun vaikuttavat merkittävästi luonnollisesti palvelun sisältö ja asiakkaan ICT-palveluntuottajan välisen palveluyhteistyön sujuvuus. Palvelukuvauksiin ja palvelusopimuksiin liitettävistä palvelutasoista sekä laatumääreistä käytetään usein nimitystä SLA (Service Level Agreement) [ITIL, 2011]. Taulukossa 2. on kuvattu tyypillisiä palvelutasosopimuksiin kuvattavia ICT-palvelujen laatumääreitä.

IT-Palveluilla on yleisesti ottaen kolme yleisluonteista peruspiirrettä [Grönroos, 2010]:

1. Palvelut ovat prosesseja, jotka koostuvat toiminnoista tai joukosta toimintoja.
2. Palvelut tuotetaan ja kulutetaan ainakin jossain määrin samanaikaisesti.
3. Asiakas osallistuu ainakin jossain määrin palvelun tuotantoprosessiin kansatuottajana.

Grönroosin peruspiirteiden perusteella tavarat ovat arvoa tukevia resursseja ja palvelut puolestaan ovat arvoa tukevia prosesseja eli prosesseja, jotka tukevat asiakkaiden arvon tuottamista ja vaikuttavat IT-palvelun laatuun. Myös ITIL määrittelee palvelun keinoksi, jonka avulla tuotetaan asiakkaalle arvoa auttamalla asiakasta saavuttamaan halutut tulokset ilman riskien ja kustannusten omistajuutta [ITIL, 2011].

Erilaisten mittaustyökalujen avulla on pyritty helpottamaan palvelun laadun mittaamista, joista yksi tunnetuimmista on 1980-luvulla kehitetty SERVQUAL-malli. SERVQUAL-malli perustuu oletukseen, että palvelun laadun mittaaminen voidaan jakaa viiteen erilaiseen tekijään: luotettavuuteen (Reliability), vaikuttavuuteen (Assurance), empatiaan (Empathy), reagoimiskykyyn (Responsiveness) ja konkreettisiin tekijöihin (Tangibles) [Wisniewski, 2001].

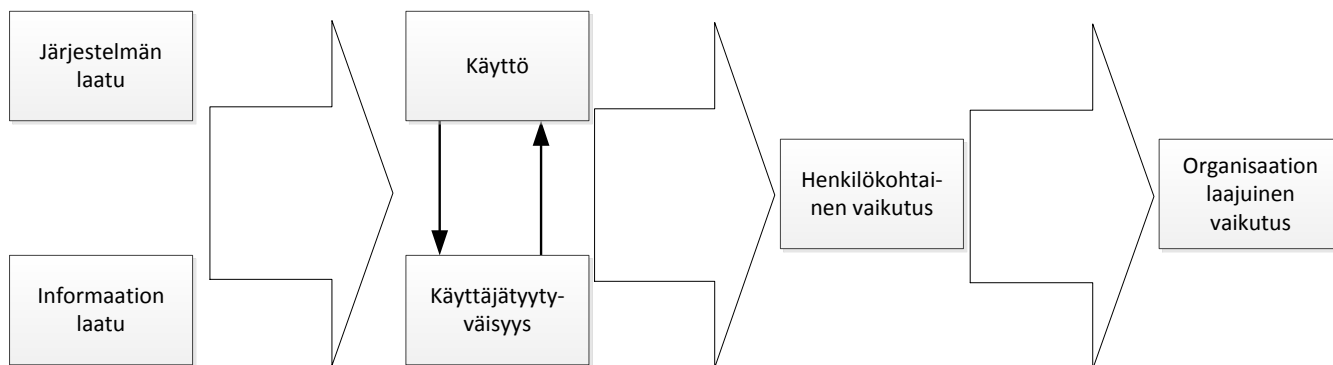
Kun tarkastellaan tietojärjestelmiä ja niiden tarjoamia palveluita, huomataan, että palvelun laadun määritelmät ja mittarit ovat osoittautuneet ongelmallisiksi tai jopa puutteellisiksi. Koska mallit ovat

syntyneet vanhanaikaisiin tarpeisiin, esimerkiksi haluan mitata asiakkaan näkökulmasta koetun palvelun laatua, ovat perinteiset mallit melko toimimattomia tietojärjestelmien kannalta. Palvelun määrittely tilanteesta riippuen on tärkeää, jotta oikeanlaisten työkalujen valinta helpottuisi. Esimerkiksi loppukäyttäjän kokemusta ei huomioida kun puhutaan tietojärjestelmien välisestä palvelusta, jolla tietojärjestelmät kommunikoivat. Palvelun laadun ja palvelun mittaaminen vaikeutuu, koska näkökulma ratkaisee, onko tarvittava palvelutaso tai palvelu saavutettu.

## 5.1 Palvelun laatu tietojärjestelmien onnistumisen näkökulmasta

DeLone ja McLean:n vuonna 1990 julkaisema tietojärjestelmien onnistumismalli auttoi ymmärtämään tietojärjestelmien onnistumisen mittaamista. Heidän mielestään tietojärjestelmien onnistumisen mittaamisen hajanaisesta tutkimuskentästä puuttui kokonaan konsensus [DeLone and McLean, 1992], jota onnistumismalli avaa (Kuva 4.). Vaikka tietojärjestelmien onnistumiseen vaikuttavista tekijöistä olikin tehty paljon tutkimuksia, oli niistä vaikea hahmottaa, mitkä tekijät vaikuttivat tietojärjestelmien onnistumiseen. Varsinaista yhtenäistä pohjaa tietojärjestelmien onnistumiseen tietojärjestelmien tutkimiseen ei ollut olemassa [DeLone and McLean, 1992].

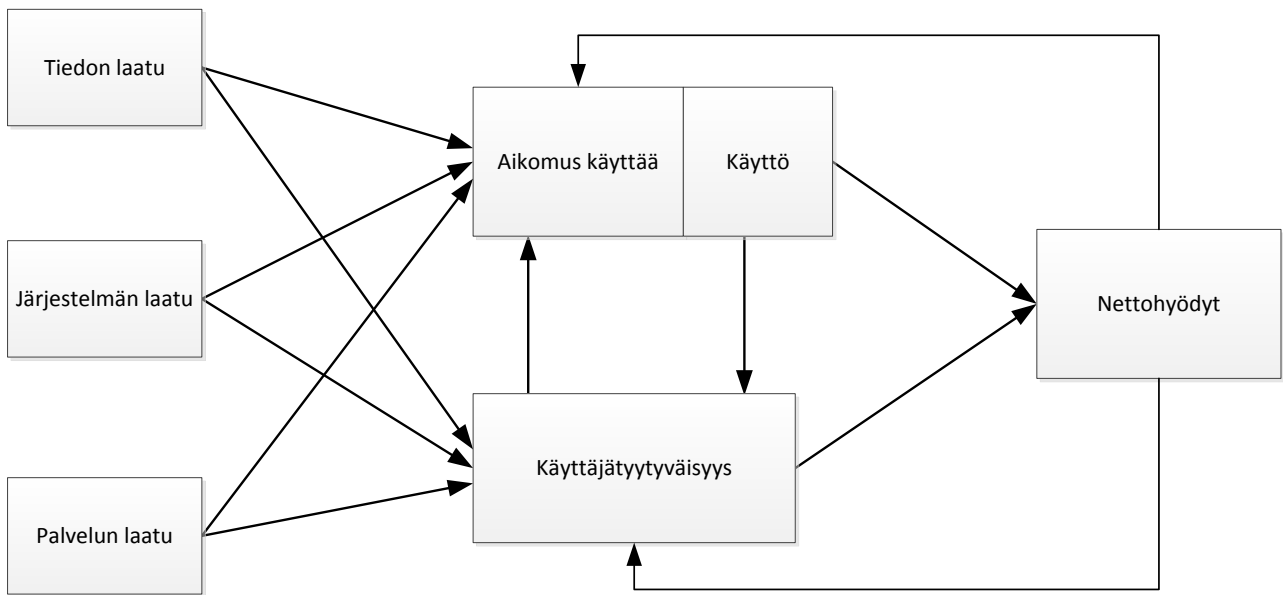
DeLonen ja McLean:n onnistumismallin pohjana on, että tietojärjestelmäprojektin onnistuminen on suhteessa toteutetun järjestelmän laatuun ja siinä hyödynnetyn informaation laatuun. Sekä järjestelmän käyttökokemus että käyttäjätyytyväisyys ovat suoraan suhteessa toteutetun järjestelmän ja siinä hyödynnetyn informaation laatuun. Vaikka käyttäjätyytyväisyys mielletään henkilökohtaisena kokemuksena, mielipiteet heijastuvat organisaatiosolle. Laadun mittareista käyttäjätyytyväisyyden merkityksen suhde muihin mittareihin on korkeampi silloin, kun tietojärjestelmä on saatu valmiiksi ja loppukäyttäjät ovat ottaneet järjestelmän käyttöön. Johdon tyytyväisyys ja muut organisaation asiat, vaikuttavat lopullisen tietojärjestelmän laatuun [DeLone and McLean, 2003].



Kuva 4. DeLonen ja McLeanin [1992] tietojärjestelmien onnistumismalli

Kymmenen vuotta ensimmäisen, vuonna 1992 ilmestyneen, onnistumismallin jälkeen DeLone ja McLean tekivät uuden päivitetyn version onnistumismallista (Kuva 5.) siihen kohdistuneen kritiikin, parannusehdotusten ja paikkaansapitävyyden vuoksi. Onnistumismallin päivittämiselle oli tarvetta, koska 1990-luvun aikana tietojärjestelmien kehitys ja niiden toimintaympäristö olivat merkittävässä muutoksessa. Internetin mukana tuoma toimintaympäristön muutos ja mahdollisuudet esimerkiksi elektroniseen kaupankäyntiin, vaikuttivat suuresti tietojärjestelmien onnistumisen mittaamiseen [DeLone and McLean, 2003]. Vaikka päivitetystä mallista peruslähtökohta on sama kuin alkuperäisessä onnistumismallissa, päivitettyyn malliin päätettiin ottaa mukaan yhdeksi muuttujaksi laatu. Uuden muuttujan avulla tietojärjestelmästä oli mahdollista arvioida palvelun, tietojärjestelmän ja tiedon laatua. Näiden kolmen osa-alueen avulla voidaan nähdä mikä on käyttäjättyvyisyys ja miten tietojärjestelmää käytetään. Käyttäjättyvyisyys ja tulevat käyttökerrat ovat nettohyötyjä, jotka saavutetaan positiivisen käyttäjättyvyyden ja tietojärjestelmien aktiivisen käytön seurauksena [DeLone and McLean, 2003].





Kuva 5. DeLonen ja McLeanin [2003] tietojärjestelmien onnistumismalli

Tämä tietojärjestelmien onnistumismalli sopii erinomaisesti sellaisten tietojärjestelmien ja niiden laadun arviointiin, joissa suorassa vuorovaikutuksessa ovat loppukäyttäjät. Tämä on nähtävissä palvelun laatu- ja käyttäjätyytyväisyyskäsitteiden kautta, koska niillä on helppo kuvata miten helppokäyttöinen esimerkiksi verkkokauppa on. Integraatioprojektien saavutetut vaikutukset tulee tarkastella koko organisaation tietojärjestelmien näkökulmasta, koska integraatioarkkitehtuureihin liittyvät toteutukset ja ratkaisut ovat usein ainoastaan operatiivisia järjestelmiä tukevia kokonaisuuksia. Hyödyt on nähtävissä siten, että liiketoimintaprosessit sekä niitä tukevat operatiiviset järjestelmät toimivat tehokkaammin.

## 6. Palvelukeskeinen arkkitehtuuri

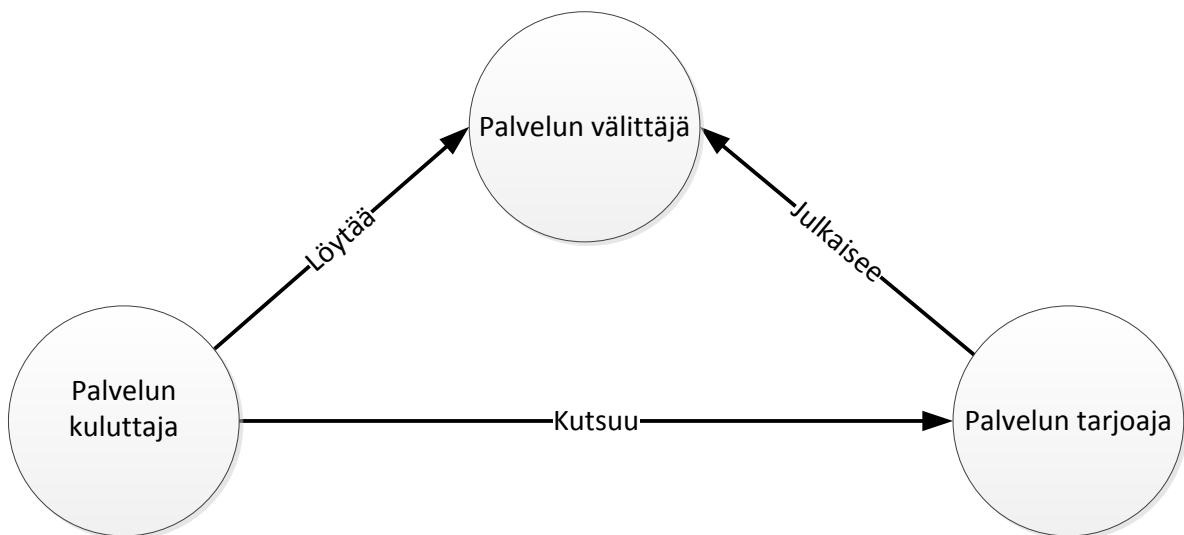
Service-Oriented Architecture (SOA) eli palvelukeskeinen arkkitehtuuri on suunnittelumalli, jonka lähtökohdaksi on luoda yleiskäyttöisiä ja uudelleenkäytettäviä palveluita. Kantava ajatus palvelukeskeisen arkkitehtuurin taustalla on päästä eroon vain yhteen käyttötarkoitukseen rakennetuista sillomaisista järjestelmistä, joiden integroiminen ja ylläpito toisten järjestelmien kanssa ovat kalliita monimutkaisten toteutusten vuoksi [Kajastila, 2012]. Palvelukeskeinen arkkitehtuuri voidaan kuvitella työkalupakkina, josta voidaan kokoamalla luoda uusia palveluita tukemaan liiketoimintaprosesseja [Josuttis, 2007]. Palvelukeskeinen tietotekniikan käsite sisältää myös paljon muuta kuin palvelukeskeisen arkkitehtuurin vaikka nämä usein ovatkin selitetty synonyymeinä. Palvelukeskeinen tietotekniikka sisältää myös käsitteen palvelukeskeisyydestä, palvelukeskeisen ratkaisulogiikan, palvelut, palveluiden yhdistämisen sekä palvelutarjoaman [Erl, 2008].

Palvelukeskeinen arkkitehtuuri on kehitetty lisäämään vanhojen järjestelmien käyttöikää sekä vähentämään IT-järjestelmien kustannuksia. Vanhat järjestelmät ovat kalliita ja vaikeita muuttaa, kun niitä olisi tarve jatkokehittää tai kun organisaation liiketoiminnan tavoitteet ja tarpeet muuttuvat. Vanhat järjestelmät sisältävät usein liiketoiminnalle tärkeitä tietoja ja toimintoja, jotka tekevät järjestelmien korvaamisesta vaikeita ja kalliita. Tämän vuoksi vanhoista järjestelmistä on järkevää muodostaa uudelleenkäytettäviä palveluja. Palveluiden standardoidut rajapinnat mahdollistavat, että eri sovellukset voivat käyttää kyseisiä palveluja riippumatta kummarkaan osapuolen toteutusteknologiasta [Hailstone *et al.*, 2007].

Tämän kappaleen tarkoituksena on avata palvelukeskeistä arkkitehtuuria. Koska palvelukeskeinen arkkitehtuuri ei ole yksittäinen tuote, joka on mahdollista organisaation ostaa, tulisi palvelukeskeistä arkkitehtuuria ajatella asenteena tai ajattelutapana, johon organisaation tulisi pyrkiä [Josuttis, 2007].

## 6.1 Palvelukeskeisen arkkitehtuurin piirteet

Palvelukeskeisen arkkitehtuurin kantavana ajatuksena on, että yrityksen kokonaisratkaisu syntyy erilaisista ohjelmistojen ja tietojärjestelmien tuottamista palveluista. Nämä palvelut ovat näkyvissä käyttäjälle erillisten rajapintojen välityksellä, jonka vuoksi palveluiden teknisen toteutuksen ei tarvitse olla yhtenäistä tai samalla tekniikalla toteutettua. Liiketoimintalogiikka muodostuu näiden palveluiden seurauksena joko itsekseen tai toisiin palveluihin yhdistettynä. Palvelukeskeinen arkkitehtuuri muodostuu joukosta komponentteja, jotka yhdessä hyödyntävät toistensa tarjoamia palveluita. SOA –ajattelun (Kuva 6.) keskeisin päämäärä on järjestelmien joustavuus.



Kuva 6. Palvelukeskeisen arkkitehtuurin toimintamalli

Uudelleenkäytettävyys on ollut pitkään yksi perinteisen ohjelmistokehityksen tavoitteista ja siksi SOA-ajattelu on tervetullut parannus. Palvelukeskeisen arkkitehtuuri tavoitteena on järjestelmäintegraation hyödyntämien hyväksi havaittujen käytäntöjen toteuttaminen ohjelmistokehitystä ylemmillä abstraktiotasoilla [Josuttis, 2007].

Tulevaisuudessa pilvi- ja verkkosovellusten lisääntyminen tulee lisäämään SOA-ajattelun hyödyntämistä. SOA:n yksi keskeisimmistä piirteistä on sen tuki heterogeenisille tietojärjestelmille, jolloin monien erilaisten tekniseltä toteutukseltaan toisistaan poikkeavien sovellusten integrointi on mahdollista [Josuttis, 2007].

## 6.2 Palvelukeskeisyys

Palvelukeskeisyys on yleistynyt termi, mutta sitä on mahdollista tarkastella myös yleisesti ilmiönä, joka näkyy erilaisissa ympäristöissä. Palvelukeskeisyyttä voidaan ajatella esimerkiksi yrityksinä, jotka ovat itsenäisiä toiminnallisia järjestelmiä, mutta tarjoavat uudelleenkäytettäviä palveluita toisille yrityksille. Yritykset käyvät vuorovaikutusta tiettyjen yhteisten standardoitujen välineiden, kuten kielen, valuutan tai kommunikaatiomuodon kautta. SOA:ssa palvelukeskeisyydellä on kuitenkin täsmällisempi merkitys ja sen tukena on palvelukeskeisiä mallinnusmenetelmiä. Palvelukeskeisyyttä on mahdollista soveltaa sekä liiketoimintalogiikassa että sovelluslogiikassa, koska liiketoimintaa voidaan mallintaa liiketoimintapalveluina, jotka implementoidaan vastaavina sovelluspalveluina [Erl, 2008]. Liiketoimintalogiikka voidaan kuvata prosessikuvauksina, liiketoimintasääntöinä tai tapahtumasääntöinä. Liiketoimintalogiikkaa hyödynnetään SOA:ssa palvelujen yhdistelemiseen tai ohjaamiseen.

Palvelukeskeisyys on usein määritelty palvelupohjaisten järjestelmien sekä palvelujen suunnittelun periaatteina. Taulukossa 3. on esitetty Thomas Erlin määrittämät ja laajasti omaksutut palvelukeskeisyyden periaatteet, jotka koskevat erityisesti ohjelmistojärjestelmien suunnittelua.

Palvelut voidaan jakaa toimintojensa mukaisesti kolmeen eri ryhmään: entiteetteihin, tehtäviin ja työkaluihin [Erl, 2008]. *Entiteettipalveluiden* tehtävänä on hallinnoida tietoja organisaation perustiedoista, kuten laskuista, asiakkaista ja lisensseistä. Entiteettipalveluihin lukeutuu myös perinteiset operaatiot kuten lisääminen, lukeminen, päivittäminen ja poistaminen

*Työkalut* voidaan ajatella tavallisina työkaluina, kuten rakentamisessa vasara ja saha. Nämä tarjoavat uudelleenkäytettäviä palveluita kuten esimerkiksi virheidenkäsittelyn ja tapahtumahistorian. *Tehtäväpalvelut* ovat käsitteenä monimutkaisempia, koska ne ovat vastuussa palveluiden organisoinnista. Tehtävät voidaan ajatella koostuvan erilaisista palvelutyypeistä, joiden vastuulla on liiketoimintasäännöistä huolehtimen [Erl, 2008].

## 6.3 Palveluväylä

Palvelukeskeisen arkkitehtuurin yksi keskeisimmistä toteutusteknologioista on palveluväylä (Enterprise Service Bus, ESB). ESB mahdollistaa yhtenäisen liitäntärajapinnan, jonka kautta palveluiden välinen kommunikaatio järjestetään ilman, että ne ovat täysin kytköksissä toisiinsa. Taulukossa 4. on lueteltu palveluväylän tärkeimpiä ominaisuuksia.

ESB:n yksi tärkeimmistä tehtävistä on mahdollistaa eri järjestelmien yhteensopivuus, joka tapahtuu välitetyn tiedon muuntamisella kohdejärjestelmälle ymmärrettäväksi [Josuttis, 2007]. Tämä muuntaminen vaatii järjestelmiltä *skaalautuvuutta* [Papazoglou and van den Heuvel, 2007]. Yleisesti tiedon muuntaminen tapahtuu määrittelemällä yksityiskohtaisesti formaatit, joita alustat ja rajapinnat pystyvät lukemaan ja käsittelemään.

Toinen ESB:n tärkeimmistä tehtävistä on *luotettava viestinvälitys*. Teknologiasta ja viestinvälityksen konfiguraatiomahdollisuuksista riippuen, tehtävä saattaa olla yksinkertainen tai erittäin vaativa. Luotettava viestinvälitys pitää sisällään ajatuksen, että lähettäjä voi olla varma, että lähetetty viesti tai palvelupyyntö menee vastaanottajalle perille. Vastaavasti lähettäjä voi olla varma, että saa vastaanottajalta vastauksen lähettämäänsä palvelupyyntöön [Josuttis, 2007].

SOA mahdollistaa *olemassa olevien toteutuksien hyödyntämisen*. Nämä perinnejärjestelmät usein muodostavat organisaation liiketoimintaprosessien ytimen, mutta näiden järjestelmien modifiointi on liian työlästä tai mahdotonta, mutta niitä ei myöskään voida hylätä tärkeytensä vuoksi. SOA mahdollistaa rakenteen, johon perinnejärjestelmät on mahdollista ”kääriä” (wrap) mukaan [Papazoglou and van den Heuvel, 2007].

*Palveluiden välinen kommunikointi* ja *dynaaminen liitettävyys* ovat hyvin lähellä toisiaan, koska ESB-implemantaatiossa palvelut kommunikoivat keskenään ja liittyvät toisiinsa. Tämä hyödyttää SOA:ssa tavoiteltavaa palveluiden välistä löyhää kytkentää [Josuttis, 2007].

### 6.3.1 Palveluväylän tehtävät

Palveluväylän eräs keskeisimmistä tehtävistä on toimia SOA:n ytimenä tarjoten mahdollisuuden organisaation tietojärjestelmien palveluiden tiedon hyödyntämiselle keskenään. Kunkin palveluväylään liitetyn järjestelmän hallinnassa on omat tiedot ja ne vastaavat siitä, että nämä tiedot ovat saatavissa palveluväylän kautta. *Yhdistettävyys* onkin on yksi tärkeimmistä palveluväylän tehtävistä.

Toinen tärkeä palveluväylän ominaisuus on *tiedon muunnos*, jotta palveluväylään kytketyt järjestelmät voivat kommunikoida keskenään. Nämä tietojen muunnokset ovat yleensä toteutettu sisäisenä palveluna ja näin ollen sovellukset voivat olla tietämättömiä toistensa sijainnista. Palveluväylään kytkeytyneet järjestelmät käyttävät erilaisia tietoja palveluväylässä kulkevista viesteistä. Jotta jokaiselle järjestelmälle ei tarvitsisi tehdä omaa viestiformaattia, käytetään kanonista tietomallia (Canonical model) tietojen siirtämisessä [Hohpe, 2012]. Kanoninen tietomalli on muista sovelluksista riippumaton viestiformaatti. Jokainen järjestelmä palveluväylässä toteuttaa

kanonisen tietomallin, jolloin järjestelmän tiedot ovat kaikkien muidenkin, mukaan lukien, uusien järjestelmien käytettävissä.

Edellisten tehtävien lisäksi palveluväylän tehtävänä on *tietojen reitittäminen*. Palveluväylään kytketyt järjestelmät hyödyntävät toistensa tietoja pyynnöin ja vastauksin. Jotta nämä pyynnot ja vastaukset menisivät järjestelmän haluamalle kohdejärjestelmälle, tulee viestit reitittää oikein, josta palveluväylä huolehtii. Tyypillisesti tämä tapahtuu XML-viesteihin asetetuilla otsikkotiedoilla (header), joissa on määriteltynä pääte piste (endpoint). Näitä pääte pisteitä voivat olla jonot (queue) tai suorat prosessikutsut (topic).

### 6.3.2 Palveluväylän käytön hyödyt

Palveluväylän yksi suurimmista hyödyistä skaalautuvuuden lisäksi on sen älykäs viestien välitys. Nämä viestit voidaan ”rikastaa” (enrich) niin, että otsikkotiedoissa on lisätietoja lähettävästä järjestelmästä tai kohdejärjestelmän päättelyyn käyttämiä parametreja. Nämä viestit on mahdollista tallentaa jonoihin, joista viestejä voidaan myöhemmin hyödyntää [Tähtinen, 2005]. Palveluväylän yhteyteen on mahdollista rakentaa virhejonoja, joihin välittämättömät tai virheelliset viestit ohjataan. Tällöin tieto virheellisestä vai toimittamattomasta viestistä säilyy jatkotutkimuksia varten [Josuttis, 2007].

Samalla palveluväylän yhteyteen voidaan rakentaa monitorointia, jolloin nähdään reaaliajassa viestien kulkeminen palveluväylässä ja virheviestien generoituminen. Tätä tekniikkaa hyödynnetään BAM-toteutuksessa (Business Activity Monitoring), jolloin liiketoiminnalle voidaan luoda tehtäviä (Work Flow tai Worklist). Liiketoiminta voi reaaliajassa hyväksyä tai hylätä sille esitettyjä pyyntöjä. Lisäksi liiketoiminnalla on mahdollisuus seurata liiketoiminnan prosesseja palveluväylässä erillisten raporttien kautta, jotka tuotetaan palveluväylän keräämän tietovaraston avulla.

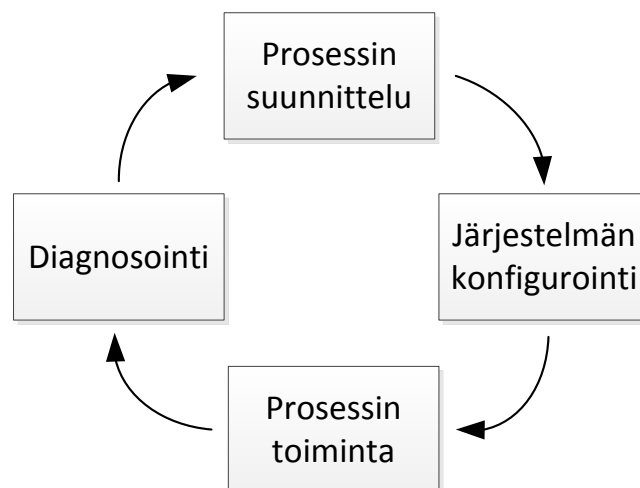
## 7. Liiketoimintaprosessit

Liiketoimintaprosessit koostuvat tietyistä, ihmisten ja järjestelmien suorittamista tehtävistä, joiden toteuttamistavan pyrkimyksenä on saavuttaa tietynlainen tuote tai tulos. Liiketoimintaprosessin käsite voidaan ajatella koostuvan resursseista, toiminnasta ja tuotoksesta, joka syntyy prosessin lopputuloksena [Tähtinen, 2005]. Liiketoimintaprosessin alku on asiakkaan tarpeessa ja prosessi päättyy asiakkaan tarpeen tyydyttämiseen. Liiketoimintaprosessi voidaan ajatella kokoelmaksi toimintoja, jolla asiakas saavuttaa halutun lopputuloksen. Prosesseja ja toimintoja, jotka eivät tuo asiakkaan kannalta mitään lisäarvoa, ei voida ajatella olevan liiketoimintaprosesseja. [Hyötyläinen, 2013].

### 7.1 Liiketoimintaprosessien hallinta

Liiketoimintaprosessien hallinnalla (Business Process Management, BPM) on merkittävä osa SOA:n onnistumisessa ja hyödyntämisessä, koska liiketoimintaprosessit ovat osa palvelukeskeistä arkkitehtuuria. Nykypäivän jatkuvasti kehittyvä yhteiskunta sekä liiketoimintaan liittyvät muutokset luovat paineita organisaatioiden liiketoimintaprosessien kehittämislle sekä hallinnalle.

Liiketoimintaprosesseilta vaaditaan jatkuvaa uusien toimintojen etsimistä sekä uusiutumista yrityksen kilpailukyvyyn ylläpitämiseksi [Hyötyläinen, 2013].



Kuva 7. Liiketoimintaprosessien hallinnan kehä [van der Aalst *et al.*, 2003]

Liiketoimintaprosessin hallintaa voidaan kuvata kehän (Kuva 7.) avulla, joka koostuu erilaisista vaiheista, jotka jokainen osaltaan tukevat liiketoimintaprosesseja [van der Aalst *et al.*, 2003]. *Prosessin suunnitteluvaiheessa*, liiketoimintaprosessit suunnitellaan tarvittaessa uudelleen. *Järjestelmän konfigurointivaiheessa* tuotetut suunnitelmat toteutetaan osaksi liiketoimintaprosessi-sovellusta, esimerkiksi työlistan hallinnointiin tarkoitettua tietojärjestelmää. Konfigurointivaiheen jälkeen siirrytään *prosessin toimintavaiheeseen*, jolloin liiketoimintaprosessit käynnistetään. *Diagnosointivaiheessa* liiketoimintaprosesseja analysoidaan virheiden tai ongelmien löytämiseksi ja selvitetään, voisiko prosessia vielä parantaa entisestään. Kehän avulla voidaan optimoida sekä tarvittaessa automatisoida liiketoimintaprosessit, jotta voidaan havaita sekä poistaa päällekkäisyydet tai hidasteet, parantaa tehokkuutta, selvittää nopeasti liiketoiminnan uudet määritykset ja prosessit sekä kyetään valvomaan ja hallitsemaan prosessien suorituskykyä.

BPM:ssa yhdistetään prosessit, tietotekniikka sekä informaatio, jotta voidaan tukea yrityksen pääresursseja: ihmisiä, informaatiota, teknologiaa sekä prosesseja. BPM:n avulla on mahdollista reaaliajassa saavuttaa tietämystä liiketoiminnan tilasta, resurssien käytöstä sekä tietojärjestelmien suorituskyvystä. BPM:n soveltamisen yksi tärkeimmistä ominaisuuksista on liiketoimintainformaation nopea saatavuus ja parempi reagointi muuttuviin markkinoihin, kilpailuedun saavuttaminen sekä liiketoiminnan kautta saavutettu parempi kannattavuus. BPM:n tavoitteena on aiemmin kuvatun kuvan 9. kehän tapaan luoda prosessien jatkuva kehittyminen määrittämällä, mittaamalla ja kehittämällä prosesseja, ja pyritään yrityksen liiketoimintaprosessien optimointiin sekä panostamaan yrityksen jatkuvan kasvun kehittymiseen [Hyötyläinen, 2013].

## 7.2 BPM:n hyödyt

Liiketoimintaprosessien hallinta mahdollistaa yrityksen muuntautuvuuden markkinoiden mukaan sekä samalla helpomman integroituvuuden liiketoimintatavoitteisiin. Tämän seurauksena, yrityksen on entistä helpompaa ja nopeampaa tuoda markkinoille uusia järjestelmiä tai palveluita.

Cumberlidge [Cumberlidge, 2007] nimeää viisi etuutta, jotka BPM tuo mukanaan:

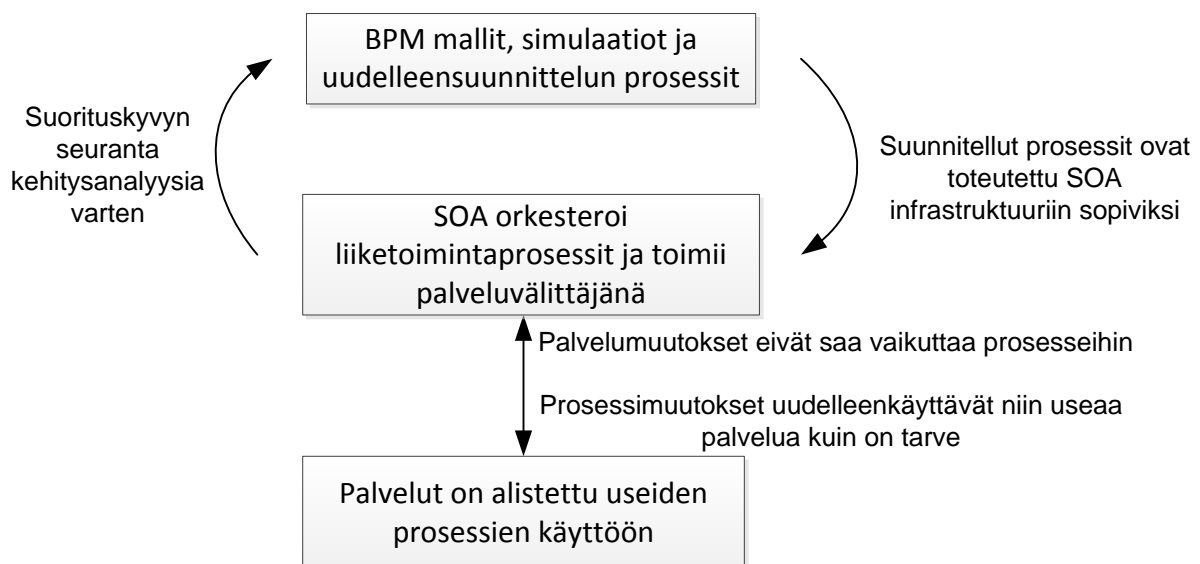
1. Tuottavuuden, tehokkuuden sekä hallinnan lisääntyminen. BPM huolehtii, että yritys työskentelee tärkeimpien tuotteiden parissa ja tuotanto on jatkuvan valvonnan alaisena. Tuloksena on tärkeimpien prosessien nopeampi läpivienti sekä parempi tehokkuus.
2. Mahdollisuuden joustavampaan liiketoimintaan, joka pystyy muotoutumaan liiketoimintavaatimuksiin nopeammin.



3. Parantaa resurssien hyödyntämistä, suunnittelua ja prosessien näkyvyyttä. Koska prosessitapahtumat ovat paremmin nähtävissä ja hallittavissa, on tilanteen seuraaminen ja raportointi helpompaa.
4. Tehokkuuden lisääntyminen ja samalla tuotantoaikojen lyhentyminen. BPM varmistaa prosessien sujuvuuden pienin voimavaroin ja valvoo samalla niiden laatua.
5. Parempi asiakastyytyväisyys sekä parempi palvelun laatu. BPM mahdollistaa palveluiden ja tuotteiden toimittamisen asiakkaiden tarpeisiin ja korkealla laadulla.

### 7.3 SOA:n ja BPM:n yhdistäminen

Liiketoimintaprosessit toimivat palvelukeskeisen arkkitehtuurin ytimessä, jonka päämääränä on luoda mahdollisimman ketterästi toteutettu ympäristö vastaamaan automatisoidusti muutoksiin [Erl, 2008]. BPM on luonnollinen tekijä täydentämään SOA:a kun tavoitteena on korkeantason liiketoiminta. BPM on mahdollista toteuttaa ilman SOA:a, mutta näiden yhdistäminen tuo molemmille etuja. SOA:n ja BPM:n yhdistäminen mahdollistaa entistä ketterämmän liiketoiminnan sekä paremman joustavuuden vastata tuleviin muutoksiin.



Kuva 8. SOA:n ja BPM välinen vuorovaikutussuhde.

SOA mahdollistaa joustavuuden, uudelleenkäytettävyyden sekä sopeutumiskyvyn yrityksen tarpeisiin, mutta yhdistämällä SOA vastamaan liiketoiminnan haasteita BPM:n avulla, yritys saavuttaa paremman vaikutuksen liiketoimintaan ja mahdollistetaan mitattavissa oleva tuotto investoinneille (Return On Investment) [Hyötyläinen, 2013]. Yhdistämällä SOA:n ja BPM:n yritys saavuttaa parempia tuloksia kuin vain toista käyttämällä. SOA parantaa uudelleenkäytettävyyttä ja tekee liiketoimintaprosesseista joustavampia, ja BPM tuottaa korkeamman tason liiketoimintaprosesseja palveluiden avulla. SOA soveltuu erityisesti pitkän elinkaaren hankkeisiin, jotka eivät ole suurissa muutoksissa lyhyellä aikavälillä tarkasteltuna. Kuvassa 8. on kuvattu miten SOA:n ja BPM:n välinen vuorovaikutus toimii [Erl, 2008].

## 8. Tapaustutkimus – integraatiopalveluita sekä konsultointia tarjoava yritys

Tapaustutkimuksen kohteeksi valikoitui projekti järjestelmäintegraatioiden palvelun varmistamiseen monitoimittajaympäristössä, jonka kehitystiimin jäsenenä toimin yhteensä vuoden ajan. Projektin tavoitteena oli luoda arkkitehtuuri, jonka ansiosta nykyisille ja tuleville asiakkaille voidaan tarjota paremmin saavutettavaa palvelua ja voidaan reagoida nopeammin häiriötilanteisiin palvelun saatavuudessa. Tavoitteena ei ollut häiriöiden vähentäminen tai ehkäiseminen vaan niiden reaaliaikainen paikantaminen ja eskalointi virheen korjaamiseksi. Projektin ansiosta yritys pystyy tarjoamaan asiakkailleen laadukkaampaa palvelua ja vähentämään asiakkaan yhteydenottoja häiriötilanteissa.

### 8.1 Yrityksen alkutila

Tapaustutkimuksessa käytetyn yrityksen lähtötilanne oli hyvin perinteinen integraatiohankkeissa nähtävä arkkitehtuuri, jota ei ole rakennettu häiriönhallintaa silmälläpitäen. Kyseisen arkkitehtuurin vuoksi häiriöiden paikallistaminen, eskalointi ja korjaaminen olivat pitkiä ketjuja, koska vastuut olivat hajautettuina monille eri toimittajille. Koska yrityksen järjestelmäintegraatioiden kysyntä ja resurssit olivat jatkuvassa kasvussa, haluttiin palvelun saatavuuteen kohdistaa resursseja entistä enemmän, jonka seurauksena tavoiteltiin korkeampaa kasvua tuottavuudessa.

Arkkitehtuuria ja palvelun saatavuutta lähdettiin uudistamaan ensin pienemmistä tai vähemmän aktiivisista integraatioista. Ensin integraatiot priorisoitiin herkkiin ja vähemmän herkkiin sekä aktiivisiin ja passiivisiin integraatioihin. Pienet ja vähän aktiiviset integraatiot eivät olleet herkimpiä vaikka häiriöitä syntyisi tai palvelu ei olisi hetkeen saatavilla, joten niiden uudistaminen toimimaan yhdessä häiriönhallinta-arkkitehtuurin kanssa olivat ensimmäisen työvaiheen prosesseja.

Tämän alkututkimuksen aikana päätettiin, että integraatioita ei haluta uudistaa alusta asti suuren työmäärän vuoksi, joten päädyttiin pohtimaan monitoroinnin lisäämistä olemassa olevaan ympäristöön. Samalla huomattiin että, rajapintojen lisääntyessä myös mahdollisia poikkeamatilanteita on enemmän ja ne ovat vaikeammin tunnistettavissa ja hallittavissa. Alkututkimuksen pohjalta löydettiin myös kaksi konkreettista ja tarpeellista kehitystarvetta ympäristössä: vikaviestien reitityslogiikka sekä käyttöliittymä vikaviestien hallinnointiin.

### 8.1.1 Aiemman arkkitehtuurin vaikutus yrityksen liiketoimintaan

Edeltävässä järjestelmäarkkitehtuurissa ei ollut mahdollisuutta valvoa integraatioprosesseissa ilmeneviä häiriöitä, joten se ei myöskään tukenut organisaation kasvua. Osa yrityksen integraatioista oli toteutettu vanhentuneina Point-to-Point sekä Hub-and-Spoke –integraatioina, joten näiden muuttaminen käytössä olevaan palveluväylään olivat ensisijaisen tärkeitä, jotta nämä integraatiot olisivat jatkossa helpommin hallittavia, niiden valvonta helpottuisi sekä niiden kustannukset olisivat alhaisemmat.

### 8.1.2 Aiempaan arkkitehtuurin liittyvät ongelmat

Yrityksen lähtötilan arkkitehtuuri oli rakennettu ainoastaan luotavia integraatioita silmälläpitäen eikä siinä ollut mahdollisuutta laajennukseen. Aiempi arkkitehtuuri koostui osittain vanhan mallisista Point-to-Point ja Hub-and-Spoke –integraatioista. Tätä oli yritetty laajentaa ottamalla käyttöön palvelukeskeinen arkkitehtuuri, johon uudet integraatiot luotiin, mutta vanhat jäivät käyttöön palvelukeskeisen arkkitehtuurin ulkopuolelle. Yrityksellä oli käytössään useita eri ratkaisuja integraatioiden luomiseen tietojärjestelmien välille. Tämä aiheutti ongelmia palvelun laadussa ja saatavuudessa, koska asiakkaan haluamaa palvelua ei voitu tukea riittävästi ilman valvontaa tai häiriöhallintaa. Asiakkaalle palvelun häiriöhallinta ei juurikaan näkynyt ja palvelupyyntöjen luominen yritykselle oli hankalaa. Koska arkkitehtuuri koostui useista eri integraatioista ja tietojärjestelmistä, oli häiriöilmoituksen selvittäminen aikaa vievää ja työlästä yritykselle, joka näkyi heikentyvänä asiakastyytyvyytenä. Useat arkkitehtuurissa käytetyt ratkaisut olivat asiakkaille täysin räätälöityjä, joten yleiskäyttöisten ratkaisujen käyttö arkkitehtuurissa oli mahdotonta ja tämä osaltaan heijastui ratkaisuihin käytettyihin työmääriin.

### 8.1.3 Monitoimittajaympäristöön liittyvät ongelmat

Palveluketjut (kuva 9.) muodostuvat monien eri toimittajien toteuttamista järjestelmistä tai niiden osista. Tämä aiheuttaa haasteita häiriötilanteiden selvittämiseksi sekä tiedottamiselle eri osapuolten kesken. Häiriöiden korjaaminen oli pääsääntöisesti kohtuullisen helppoa, mikäli häiriön sijainti ja kyseessä olevasta osasta palveluketjussa vastaava toimittaja oli selvillä. Ongelma muodostuikin siitä, että häiriöpaikkaa ei kyetty paikallistamaan täsmällisesti, mikä aiheutti ongelman siirtelyä toimittajalta toiselle ja viivästyksiä häiriöiden korjaamisessa.

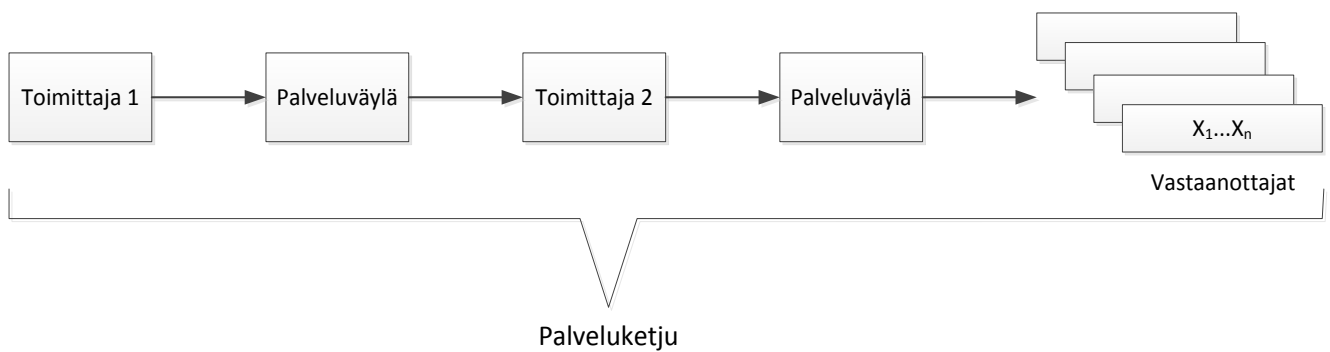
Lisäksi oli huomioitava, että palveluketjut saattavat käyttää samoja yhteiskäyttöisiä järjestelmiä/palveluita, jolloin niissä ilmenevät häiriöt heijastuivat kaikkiin niitä käyttäviin palveluketjuihin. Sama ongelma toistui verkkoliikenteen ja palvelinalustojen häiriöiden osalta.

Yksittäinen häiriötieto palveluketjussa oli pystyttävä paikallistamaan, priorisoimaan ja kohdistamaan oikealle vastaanottajalle toimenpidepyyntöä varten. Muille palveluketjun jäsenille riittäisi tiedonanto havaitusta ongelmasta.

Tiedottaminen vaatisi toimiakseen keskitetyn kauttakulkupisteen, jonka kautta toimittajat voisivat kommunikoida/informoida toisiaan häiriön syntymisestä, selvittelystä, korjaustoimenpiteistä aina häiriön korjaamiseen saakka.

Häiriön paikallistaminen palveluketjussa sekä aloitus- että korjaamisaika ovat olennaisia tietoja palvelutasosopimusten (SLA) kannalta. Alkutilanteessa häiriön alkamisaikaa ja häiriönaiheuttajaa oli vaikea todentaa, mikä aiheutti sopimusten palvelutason todentamisen lähes mahdottomaksi. Tämän vuoksi häiriötilanne oli voitava kohdistaa oikealle sopimukselle (toimittajalle) ja häiriön korjaamiseen kulunutta aikaa oli voitava mitata tarkemmin SLA:n vaatimusten vuoksi.

Myös tavanomaiset huoltokatkokset ja korjaukset tuottivat omat ongelmansa monitoimittajaympäristössä, koska katkoksista ei muistettu välttämättä informoida kaikkia palveluketjun osapuolia.



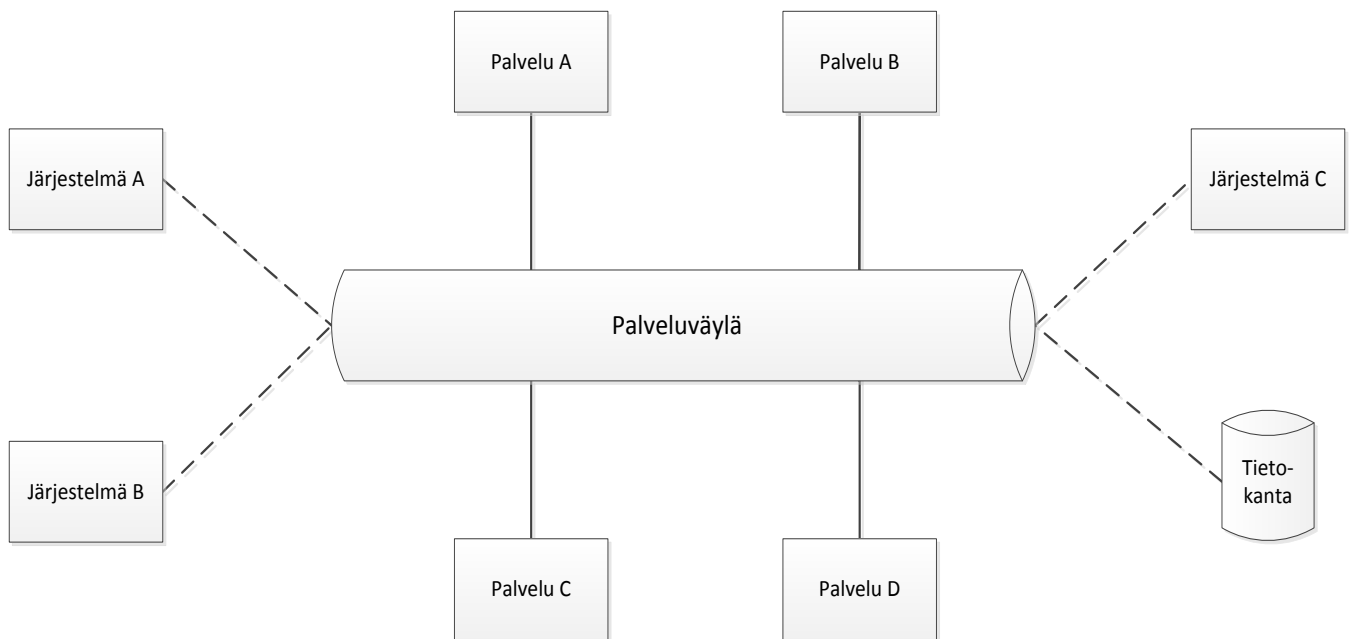
Kuva 9. Palveluketjun rakenne

## 8.2 Yrityksen tavoitetila

Yrityksellä oli selkeä tarve ja suunnitelma projektille, koska vastaavaa valvontajärjestelmää ei ollut vielä olemassa. Tämä tuo yritykselle selvää kilpailuetua muihin toimijoihin nähden ja samalla uuden mahdollisuuden lisämyyntiin yrityksen tarjoamille ja ylläpitämille integraatioille. Samalla yrityksen vanhat integraatiot päivitetään vastaamaan uutta arkkitehtuuria ja näin pystytään tarjoamaan myös olemassa oleville asiakkaille valvontapalveluita. Yrityksellä oli selkeä visio siitä, että se haluaa olla markkinajohtaja integraatioiden osalta ja tarjota asiakkailleen laadukkaampaa palvelua.

Yrityksen näkökulmasta projektin kantavana ajatuksena oli olemassa olevan arkkitehtuurin laajennus tai tarvittaessa siirto, jotta integraatioihin saadaan toteutettua häiriönhallintaa sekä valvontaa. Lisäksi tavoitteena oli liiketoiminnan kasvun tukeminen tarjoamalla laadukkaampaa ja varmempaa palvelua nykyisille sekä uusille asiakkaille. Jotta asiakkaalle voitaisiin tarjota parempaa palvelua, selvitettiin häiriöiden ennakointia niin, että häiriöilmoitukset eivät tulisi asiakkailta tai ne olisi jo yrityksen tiedossa kun asiakas reklamoi häiriöstä.

Häiriöiden ratkaiseminen ja niiden työmääräarviot eivät olleet yrityksen tiedossa, koska häiriöilmoitus saattoi olla eskaloituna väärälle toimijalle, joka oli siirtänyt ilmoituksen eteenpäin toiselle toimijalle. Tavoitteena oli selkeyttää ja keskittää häiriöistä johtuvien ilmoitusten eskalointia sekä ratkaisua esimerkiksi yhteiskäyttöisen portaalin avulla. Nykyisessä arkkitehtuurissa asiakas ilmoitti virheestä yhteiskäyttöisessä tiketöintijärjestelmässä, josta tuli ilmoitus järjestelmän ylläpitäjille. Ylläpitäjät lähtivät ratkaisemaan häiriötä, mutta koska toimijoita saattoi olla useita, oli oikean toimijan ja työmäärän arviointi hankalaa. Tavoitetilassa ylläpitäjät pystyvät valvomaan integraatioita ja pystyvät tunnistamaan oikean toimittajan, jonka järjestelmässä tai palvelussa häiriö on. Lisäksi ylläpitäjät näkisivät reaaliajassa, missä tilassa ilmoitus on ja millä toimijalla. Asiakas saa tarvittaessa tai halutessaan tiedon siitä, mikä kunkin ratkaisupyynnön tila on ja milloin se on ratkaistu.



Kuva 10. Yrityksen palveluväylän arkkitehtuuri integraatioissa

Projektin kautta yritys tavoittelee parempaa kommunikaatiota asiakkaan suuntaan sekä parempaa asiakaslähtöisyyttä integraatioiden suunnittelussa. Tätä kautta saavutetaan parempi asiakastyytyväisyys ja paremman yhteistyökumppanin status. Toteutuessaan projekti helpottaa yrityksen palveluiden myyntiä ja parantaa kilpailukykyä.

Esitetyn ratkaisun tavoitteena oli integroida toimittajien ja palveluntuottajien toimintamallit yhteensopiviksi yrityksen kanssa. Käytännössä tämä vaati yhdenmukaiset tavat lähettää ja vastaanottaa tietoa eri osapuolten kesken. SLA:n erilaisuus toi lisäongelman toimintamallien yhtenäistämiseksi, jonka vuoksi palveluketjussa toimivilla järjestelmillä ja palveluilla tuli olla lähtökohtaisesti samantasoiset palvelutasokuvaukset. Toimintamallien yhteensovittamisen lähtökohtana oli nopeuttaa koko ketjun toimintaa ja tällä tavoin taata parempaa ja nopeampaa reagoitua häiriöihin. Häiriöiden kohdalla tavoitteena oli niiden mahdollisimman hyvä paikallistaminen sekä priorisointi, jotta ratkaisuvastuu kohdistuisi oikealle taholle. Häiriöiden kohdistamisen oikealle taholle ja häiriön korjaamiseen kuluvan ajan kautta tavoitteena oli päästä mahdollisimman hyvään SLA:n seurantaan.

Tavoitteena oli myös lisätä häiriöinformaatiota palveluketjussa, jolloin päästään parempaan tiedotukseen ja ohjaukseen. Ratkaisun on mahdollistettava myös muun tyyppinen informaation levittäminen palveluketjussa oleville toimijoille kuten esimerkiksi huoltokatkot.

### 8.2.1 Liiketoiminnan tavoitteet

Liiketoiminnan asettamat tavoitteet olivat pääasiassa kustannusten tehostaminen, jotka syntyivät uuden asiakkuuden solmimisesta sekä palvelutason ylläpitämisestä. Liiketoiminnan tavoitetilassa palvelutasoa voidaan mitata sekä parantaa uuden arkkitehtuurin myötä. Yritys määrittelee palvelutason palvelukohtaisesti kunkin toimittajan kanssa erikseen. Projektin kautta häiriönhallinnan kautta saadaan tarkempaa tietoa kuinka toimittaja on häiriötiedon vastaanottamisesta korjaukseen asti toiminut. Jokaisesta toimenpiteestä jää järjestelmään aikaleima, jonka pohjalta palvelutason toimivuutta voidaan seurata toimittaja- ja palvelukohtaisesti. Tällä saavutetaan jatkossa kustannussäästöjä sekä saavutetaan kilpailuetua toimittaja- ja ratkaisuketjua tehostamalla.

### 8.3 Siirtymävaihe uuteen arkkitehtuuriin eli migraatio

Kun projektin kehitystyötä aloitettiin suunnittelemaan, otettiin asiakkaan edustaja mukaan projektiin. Asiakas oli jatkuvasti koko projektin ajan mukana, mutta ennen migraatiota ja suunnittelun aikana haluttiin myös asiakkaan edustaja mukaan, joka toimi teknisen konsultin roolissa. Ennen varsinaista migraatiota pyydettiin asiakkaita antamaan palautetta heille toteutetuista integraatioista ja niiden puutteista tai ongelmista. Tämä saatu palaute käytiin tarkasti läpi teknisen konsultin sekä yrityksen työryhmän kanssa ja luotiin lista vaatimuksista, jotka uuden arkkitehtuurin tulisi täyttää. Tämän lisäksi luotiin toimintasuunnitelma työvaiheista ja sovittiin projektin aikataulusta, jolloin uusi arkkitehtuuri olisi valmis.

Suunnitellun toteutuskokonaisuuden lähtökohtana lähdettiin siitä, että ICT-häiriönhallinta toteutetaan osaksi palveluväylää. Integraatioiden ja palveluiden valvonnan toteutus suunniteltiin kolmeen vaiheeseen alla olevan kuvan 11. mukaisesti.



Kuva 11. Toteutuksen päävaiheet



### 8.3.1 SOA-Valvonta

Migraation alkuvaiheessa haasteeksi tulivat kustannukset, jotka uhkasivat nousta korkeiksi uusien tuotteiden lisensoinnin kautta. Tämän vuoksi päädyttiin uudelleenkäyttämään mahdollisimman paljon jo olemassa olevia lisenssejä sekä palvelimia. Ennen varsinaista migraatiota uusille palvelimille luotiin virtuaalipalvelimia, jolloin lisenssien kustannuksia saatiin laskettua ja ylläpito oli yksinkertaista esimerkiksi huoltokatojen aikana. Lisenssien kustannusten noustessa, jouduttiin uudelleen arvioimaan valvontatyökalujen tarve ja mahdolliset päällekkäiset ominaisuudet. Tämän uuden kartoituksen seurauksena huomattiin, että kaikkia valvonta-työkaluja ei ollut järkevää lisensoida tai ottaa käyttöön. Uuden kartoituksen myötä huomattiin myös, että kaikki tuotteet eivät olleet yhteensopivia ja siksi niitä ei otettu käyttöön projektissa.

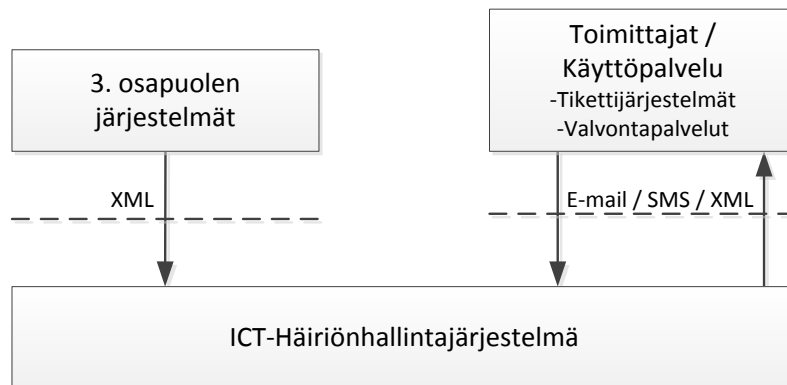
Yrityksen ensimmäinen vaihe oli siirtää vanhat integraatiot ja tärkeimmän prioriteetin integraatiot käyttämään palveluväylää, jolloin integraatiot saatiin valvonnan piiriin. Aiemmat Point-to-Point sekä Hub-and-Spoke –tekniikoilla toteutetut integraatiot siirrettiin käyttämään yhteistä palveluväylää. Tämä migraatio suoritettiin yhdessä asiakkaan teknisen konsultin kanssa, jolloin integraatioiden toimivuus voitiin testata toimivaksi ennen käyttöönottoa. Vanhat integraatiot toimivat vielä siirtymävaiheessa kun uutta järjestelmää testataan, mutta poistuvat käytöstä, kun uusi arkkitehtuuri on testattu toimivaksi.

### 8.3.2 ICT-Häiriönhallintaratkaisu

Projektin migraation toisessa vaiheessa uusi valvonta-arkkitehtuuri toteutettiin palveluväylän rinnalle, jolloin olemassa olevaa palveluväylää pystyttiin hyödyntämään häiriönhallinnassa. Palveluväylään luotiin standardit rajapinnat, joiden kautta voidaan vastaanottaa eri palveluiden tilaa kuvaavaa valvontatietoa. Rajapintaan voidaan lähettää virhetietoa sekä organisaation sisältä että myös ulkopuolelta. Yhtenäistä rajapintaa (Kuva 12.) voi käyttää sisäiset ja ulkoiset järjestelmät, toimittajan tikettijärjestelmät ja portaalikäyttöliittymä, jonka kautta käyttäjä voi raportoida manuaalisesti häiriöstä. Tämän lisäksi luotiin valvontatiedon integraatorajapinta osaksi palveluväylää, jonka avulla valvontatieto toimitetaan valvontatietokantaan raportointikäyttöön jalostettavaksi.

ICT –palvelurakenteiden valvontatavat toteutettiin kahdella tavalla, tuotepohjaisella (Actional ja sen Agentit sekä Oracle BAM) sekä räätälöitävällä osuudella. *Actional –valvontatuote* koostuu valvonta-agenteista, jotka valvovat nimetyn integraation toimintaa. *Oracle BAM:n* tehtävä on laajentaa valvontaa liiketoiminnalle sopivaksi. Agenttiperustaisessa valvonnassa käytetään

valvontajärjestelmän omia työkaluja (agentteja) mahdollistamaan eri palvelu- ja järjestelmärakenteiden toiminnan valvontaa. Agenttiperusteista valvontaa tullaan käyttämään yrityksen omassa tuotantoympäristössä sekä mahdollisuuksien mukaan ulkoisten palveluiden valvonnassa.



Kuva 12. ICT-häiriönhallinta ja yhtenäiset rajapinnat

### 8.3.3 Portaaliratkaisu

Migraation kolmannessa vaiheessa luotiin keskitetty valvontanäkymä portaaliin, josta voi seurata häiriötilannetta. Näkymä sisältää myös linkityksen palvelukohtaiseen näkymään. Käyttöliittymän käyttö on käyttöoikeuksin rajattu. Käyttöliittymään suunniteltiin niin sanotut liikennevalot, joiden avulla käyttäjä näkee integraatioiden sekä sen palveluiden reaaliaikaisen toiminnan. Liikennevalot koostuvat kolmesta väristä:

- Punainen, tarkoittaa että integraatiossa tai palvelussa on häiriö eikä viestiliikenne toimi. Tämä on kriittisin ympäristön häiriöistä.
- Keltainen, tarkoittaa että integraatiossa tai palvelussa on esiintynyt häiriöitä, mutta viestiliikenne toimii osittain tai normaalisti. Tämä voi johtua esimerkiksi havaituista virheistä ja katkoksista verkkoliikenteessä tai verkkoliikenteen ylikuormituksesta.
- Vihreä, tarkoittaa että integraatio tai palvelu toimii normaalisti eikä aiheuta toimenpiteisiin ole.

Liikennevalojen avulla käyttäjä pystyy tekemään tarkastukset ja korjaukset juuri oikeaan integraatioon tai palveluun, jolloin häiriön paikallistaminen ja korjaaminen on nopeampaa. Käyttäjää

pystyy tarvittaessa seuraamaan häiriön ratkaisua portaaliin rakennetun tiketöintijärjestelmän kautta tai kuittaamaan häiriön ratkaistuksi.

## 8.4 Toteutunut arkkitehtuuri

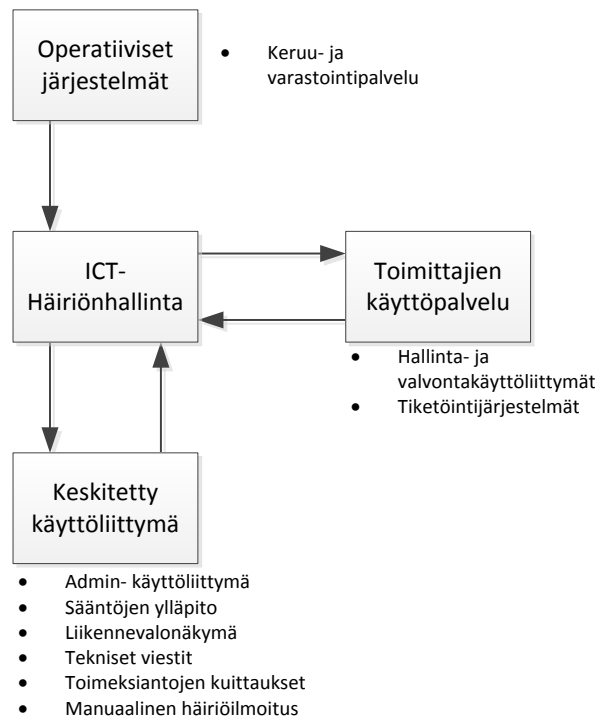
Projektin lopputuloksena syntyi palvelukeskeiseen arkkitehtuuriin perustuva valvontajärjestelmä, jossa hyödynnettiin yrityksen olemassa olevaa palveluväylää. Palveluväylää ja sen olemassa olevia palveluita hyödyntämällä, saatiin kaikki yrityksen toteuttamat integraatiot käyttämään yhteistä palveluväylää ja samalla saatiin myös vanhat integraatiot valvonnan piiriin ja voitiin luopua vanhanaikaisista point-to-point sekä hub-and-spoke –mallisista integraatioista.

### 8.4.1 Palveluarkkitehtuuri

Projektin myötä päätettiin luoda täysin uusi palveluarkkitehtuuri, jonka avulla valvontaa ja tiedotusta olisi helppo ohjata. Uusi palveluarkkitehtuuri (Kuva 13.) muodostuu useista eri komponenteista, jotka ovat: Operatiiviset järjestelmät, ICT-häiriönhallinta, yleinen valvontakäyttöliittymä sekä operatiiviset palvelut. Operatiiviset järjestelmät integroituvat toimintamalleiltaan ja teknisiltä ratkaisuiltaan yrityksen palvelumalliin sekä kokonaisarkkitehtuuriin.

ICT-häiriönhallinta koostuu kolmesta eri osasta: *häiriönhallintaosuudesta*, joka rakennettiin osaksi palveluväylää. *Häiriönhallinnan väylästä* (yhteiskäyttöinen palveluväylä), joka tarjoaa integraatoratkaisun eri palveluiden ja palvelupisteen väliselle viestiliikenteelle, esimerkiksi valvontatiedon suhteen. *Actional –valvontatuotteesta*, jolla saadaan valvontaan (tarvittaessa) kaikki SOA –pohjainen toiminnallisuus ja palvelimet. Sekä *Oracle BAM-ratkaisusta*, joka mahdollistaa Actional –tuotetta laajemman valvomisen ja hyväksynnän liiketoiminnan tasolta saakka

Yleinen valvontakäyttöliittymä tarjoaa palvelupisteelle, yrityksen edustajille ja operatiivisten palveluiden tuottajille keskitetyn valvontaratkaisun. Lisäksi Operatiivisten palveluiden omat valvonta- ja hallintatoiminnot täydentävät keskitettyä valvontaratkaisua.



Kuva 13. Palvelun ja hallinnan kokonaisarkkitehtuuri

*Operatiiviset järjestelmät.* Kokonaisuuteen liittyvät operatiiviset palvelut, kuten esimerkiksi keruu- ja varastointipalvelu, jotka välittävät ajantasaista valvontatietoa häiriönhallinnan väylään (yleinen palveluväylä). Operatiivisten järjestelmien tehtävänä on välittää valvontaa edellyttävien kohteiden ja tapahtumien poikkeamatiedot väylään reaaliaikaisesti. Kunkin poikkeamaviestin tulee sisältää riittävästi tietoa, jotta viesti voidaan kohdentaa oikein, valvontakäyttöliittymään saadaan relevantti tilannetieto ja mahdollisia tiketöinti- ja hallintatoimenpiteitä, sekä SLA-raportointia varten saadaan riittävä tietosisältö. Operatiiviset järjestelmät tuottavat tilannekohtaisesti häiriötietoa virheistä ja poikkeamista, kuten esimerkiksi keräämättä jäänyt tieto tai tietoliikenneyhteyksissä ilmennyt vika. Poikkeamaviesti sisältää vikakohteen konfiguraatietokannan sisältöä vastaavan tunnisteiden sekä aikaleimatiedon, jotta häiriötietoa voidaan hyödyntää myös automatisoidun palvelutasoraportoinnin tuottamisessa. Viestin tunnisteesta ja tilasta on pääteltävissä onko tarpeen luoda asiaan liittyen uusi tiketti. Yhdestä ja samasta häiriöstä ei generoidu päällekkäisiä toimeksiantoja, koska päällekkäisten häiriöilmoitusten ja toimenpidepyyntöjen estämiseksi lähetetään vain yksi ilmoitus palveluketjussa prioriteetissa korkeimmalla olevalle.

*Häiriönhallinnan väylä (palveluväylä).* Palveluväylä tarjoaa viestijono-teknoologiaan pohjautuvan integraatoratkaisun häiriö- ja tikettitietojen välitykseen monitoimittajaympäristössä ja eri

operatiivisten palveluiden välillä. Palveluväylä sisältää sääntökirjaston konfiguraatietietokannan sisältämien elementtien vikatilannekäsittelyyn. Sääntöjen avulla voidaan esimerkiksi ohjata tiettyä vikaa koskeva häiriötieto keskitettyyn tiketointijärjestelmään ja lähettää automaattisesti viesti kyseisen vian korjauksesta vastaavalle henkilölle. Samoin virhetilanteiden seurauksena voidaan automaattisesti generoida riittävällä informaatioisällöllä varustettu tiketti palvelupisteen ja operatiivisen palvelutuottajan tiketointijärjestelmiin. Palvelutasoraportoinnin kannalta on olennaista, että eri vika- ja tikettiviesteihin kirjattujen aikaleimojen kautta voidaan jäljittää palvelutason toteutuminen kunkin palveluketjussa mukana olevan toimijan osalta.

Kunkin operatiivisen järjestelmän virhetilanteen osalta on olemassa hienojakoisempi virheenkäsittely. Esimerkiksi lyhytkestoinen tietoliikenneongelma ei saa aiheuttaa välitöntä hälytystä, koska vika voi korjaantua itsestäänkin nopealla aikavälillä. Jokaisen yhteysvirheen jälkeen tullut hälytys kuormittaisi turhaan vastuullista toimittajaa, joka saattaisi heijastua palvelun laadun heikkenemisenä. Toisaalta, jokaisesta viasta tulee kuitenkin jäädä lokiin merkintä ja toistuvat ”väärät” hälytykset voivat indikoida todellista vikaa, jolloin todellinen hälytysviesti tulee lähettää. Lähtökohtaisesti tällaiset virheenkäsittelyn säännöt luotiin osaksi keskitettyä häiriönhallinnan kokonaisratkaisua. Keruu- ja varastointipalvelun toimittaja osallistuu säännösten luontiin yhdessä yrityksen ja muiden palvelutoimittajien kanssa.

*Keskitetty valvontakäyttöliittymä.* Kokonaisnäkyvän tarjoaminen eri järjestelmien toimintaan ja häiriötietoon on keskitetyn valvontakäyttöliittymän tärkein tehtävä. Sen kautta on saatavilla:

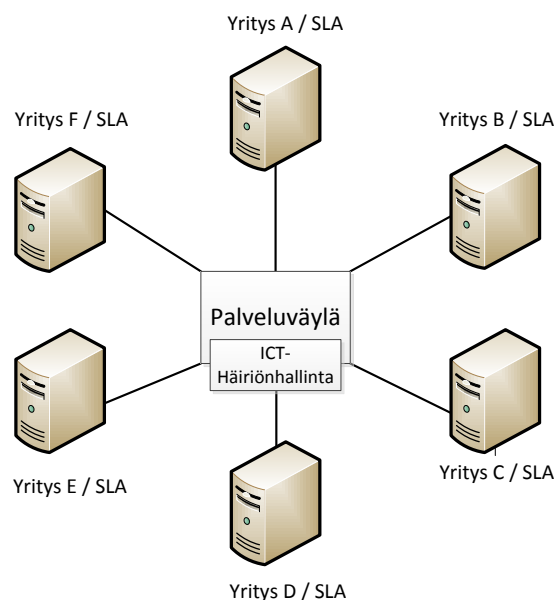
- Näkymä ja pääsy eri palveluiden tuottamaan häiriötietoon.
- ”Liikennevalonäkymä”, joka välittömästi kohdentaa vian yhteen järjestelmään tilanteessa, jossa häiriön vaikutusalue ulottuu useamman palvelun piiriin. Palveluiden välisiä rajapintoja tarkkailevien vasteaika-agenttien antamat tulokset ovat tässä keskeisessä roolissa.
- Annettujen huoltotoimeksiantojen tai vikakorjausten kuittaustoiminnot.
- Palvelukohtaisten käyttäjäoikeuksien hallinta.
- Häiriöviestien käsittelsääntöjen konfigurointi.
- Häiriöviesteihin liittyvän viestinnän konfigurointi (esimerkiksi SMS-viestien lähetyksen vastuuhenkilöille).

Palvelukohtaisen suorituskyvyn ja kokonaisvasteaikojen mittaamisessa hyödynnetään Actional - valvontatuotetta, ja osa valvontatiedosta on käytännössä saatavilla Actional -tuotteen käyttöliittymän kautta. Actionalin hyödyntäminen kytkeytyy erityisesti mittausagenttien välittämän tiedon esittämiseen.

*Operatiivisten palveluiden omat valvonta- ja hallintatoiminnot.* Keruu- ja varastointipalvelun rakentamisen yhteydessä toteutettiin sen käyttöpalvelun tuottamisen mahdollistava valvonta- ja hallintakäyttöliittymä. Keskitetty valvontakäyttöliittymä integroituu keruu- ja varastointipalvelun hallintakäyttöliittymään siten, että keskitetyn käyttöliittymän yleisestä tiedosta on tarpeen vaatiessa voitava avata keruu- ja varastointipalvelun hallintakäyttöliittymä, joka antaa tilanteesta yksityiskohtaisempaa tietoa. Myös yrityksen edustajilla on pääsy keruu- ja varastointipalvelun hallintatoimintoihin.

#### 8.4.2 Palvelutaso, muutoksenhallinta ja ennakoitua huoltotoimenpiteet

Palvelutaso määritellään kunkin toimittajan kanssa erikseen (Kuva 14.). ICT-häiriönhallinnan kautta saadaan tarkempaa tietoa kuinka toimittaja on häiriötiedon vastaanottamisesta aina korjaukseen asti toiminut. Jokaisesta toimenpiteestä jää järjestelmään aikaleima, jonka pohjalta palvelutason toimivuutta voidaan seurata toimittajakohtaisesti.



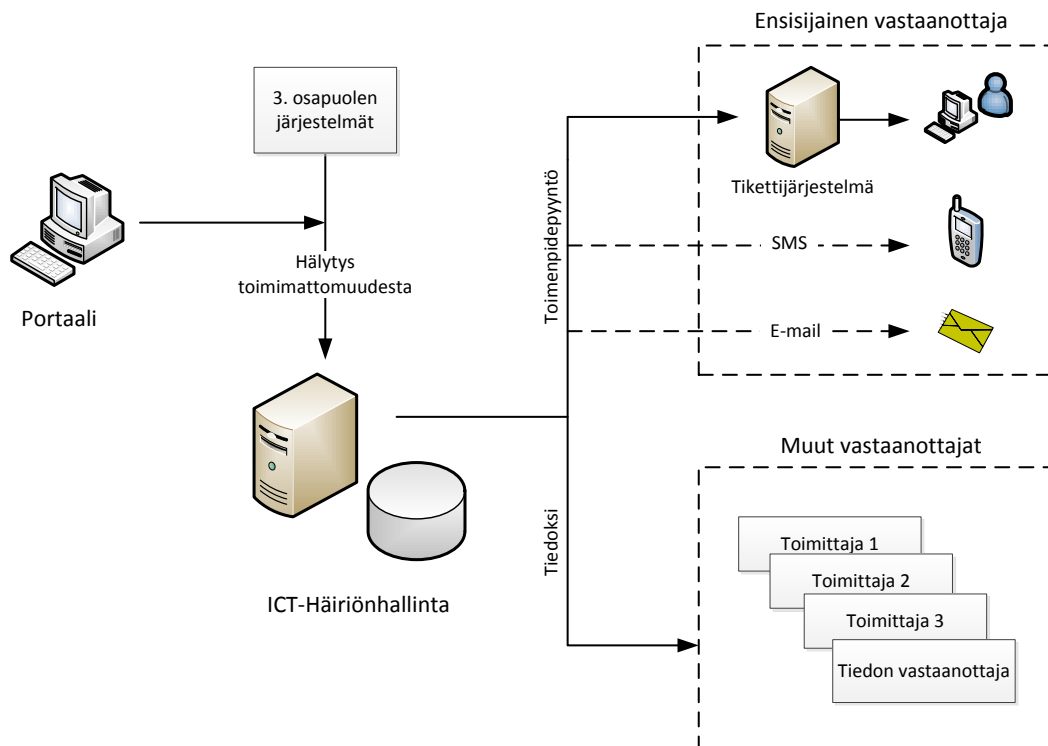
Kuva 14. Monitoimittajaympäristö ja SLA

Järjestelmien jatkokehityksen muutoksenhallinta menee kunkin yhtiön käytäntöjen mukaan. Palvelukeskeisessä arkkitehtuurissa häiriöilmoitukset toimittajille kulkevat sähköisesti esimerkiksi sähköisten rajapintojen kautta, sähköpostina ja SMS-viestinä. Jos ICT –häiriönhallintajärjestelmästä lähetetään tietoa, joka ei ole SLA:n alaista, se voidaan lähettää toimittajalle tiedoksi. Tämän jälkeen häiriöstä tuotetaan tarvittaessa kunkin toimittajan käytäntöjen mukainen selvitys- tai muutospyyntö. Selvitys- tai muutospyyntö tekee tässä tilanteessa ICT-häiriönhallintajärjestelmän omistava yritys. SLA:sta riippuen ICT –häiriönhallintajärjestelmästä voidaan lähettää joko toimenpidepyyntö tai vain ilmoitus häiriöstä.

Ennakoiduissa huoltotoimenpiteissä ICT -häiriönhallintajärjestelmää voidaan käyttää informaation jakelukanavana, jolloin voidaan varmistua että kaikkia osapuolet saavat tiedon huoltokatkoksista. ICT -häiriönhallintajärjestelmä tietää palveluketjuittain järjestelmät/palvelut sekä niiden toimittajat ja muut tarvittavat osapuolet kuten esimerkiksi järjestelmän haltijat ja tiedon vastaanottajat. Huoltokatkostiedotteessa voidaan välittää esimerkiksi ajankohta, kesto, mahdolliset vaikutukset, varautumiset ja roll back –tiedot (palautus alkuperäiseen tilanteeseen). Tietojen syöttö tapahtuu portaalikäyttöliittymän kautta tai suoraan toimittajan tikettijärjestelmästä syötetystä tiketistä.

### 8.4.3 Tiedottaminen palveluketjussa

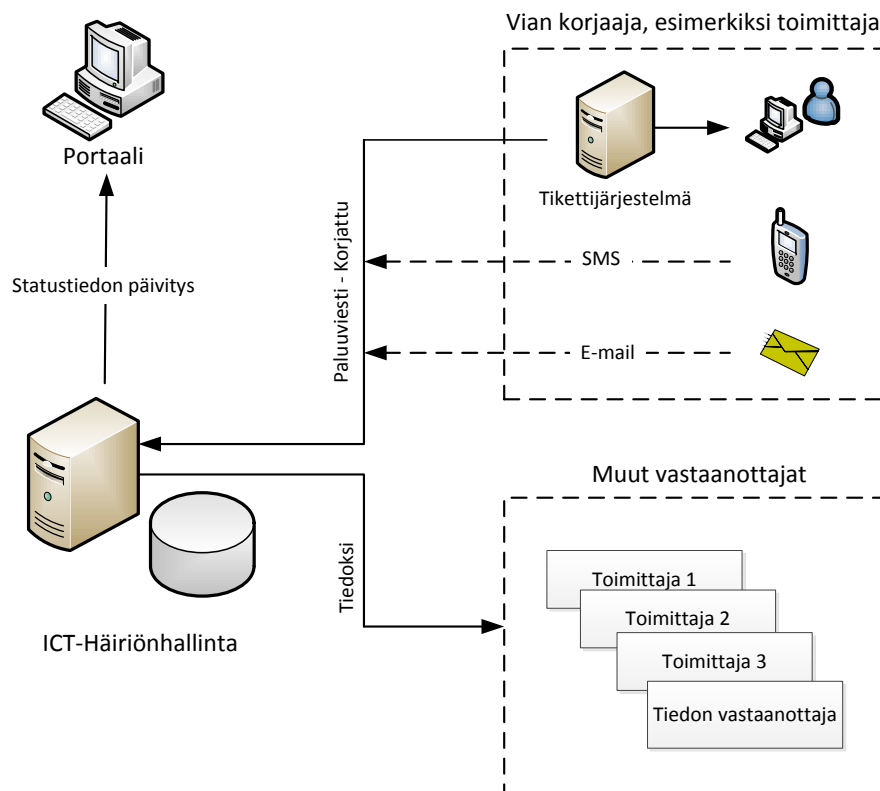
*Tiedon vastaanottaminen.* Häiriötietoa vastaanotettaessa (Kuva 15.) tutkitaan sääntöjen mukainen käsittely ja lähetetään esisijaiselle vastaanottajalle toimenpidepyyntö ja muille ketjussa toimijoille tiedonanto häiriöstä. ”Tiedoksi” vastaanottajat voivat olla esimerkiksi käyttöpalvelu, palveluketjussa olevat muut toimittajat ja mahdollisesti tiedon lopullinen vastaanottaja. Häiriötiedotus tehdään ennalta toteutettujen sääntöjen mukaisesti. Häiriötiedot voidaan lähettää eri osapuolille palveluäylyn avulla erilaisissa muodoissa kuten esimerkiksi XML, E-mail ja SMS-viestinä.



Kuva 15. Häiriötiedon vastaanotto, toimenpidepyyntö ja tiedotus

*Tiedon päivittäminen tai kuittaaminen.* Toimittajan korjattua vian, toimittaja lähettää tarvittavat tunnistetiedot häiriöstä, jotta sen statusta voidaan päivittää ja siitä voidaan informoida kaikkia tarvittavia osapuolia (Kuva 16.). Mikäli kuitenkin käy niin että esimerkiksi toimittaja on tarkistanut ”oman” osuutensa palveluketjusta ja todennut sen toimivan oikein, niin toimittaja voi lähettää siitä viestin järjestelmään, jolloin järjestelmä tutkii säännöstöä ja toimii sen mukaisesti. Seuraava sääntö voi olla esimerkiksi toimenpidepyynnön lähettäminen kaikille palveluketjussa oleville häiriön selvittämiseksi. Kun häiriön syy on löydetty ja korjattu, voi toimittaja päivittää tietoa häiriötietojärjestelmään, jolloin muita palveluketjussa toimijoita tiedotetaan häiriön syyn löytymisestä. Häiriöille voidaan asettaa myös automaattinen eskalointi tai muistutus, mikäli toimittaja ei reagoi SLA:n mukaisesti häiriöilmoitukseen.





Kuva 16. Häiriötiedon päivitys ja tiedottaminen

#### 8.4.4 Palveluiden valvonta ja hallintamalli

Projektissa toteutunut valvonta-arkkitehtuuri (Kuva 17., liitteet) mahdollistaa ongelmatilanteiden tarkemman ja nopeamman paikallistamisen, reagoinnin ja tiedottamisen palvelun tuottajien eri osapuolten kesken. ICT-häiriönhallinnalla halutaan varmistaa, että palveluketjussa toimivat toimittajat ja muut mahdolliset häiriötiedon tarvitsijat saavat tiedot reaaliaikaisina sitä mukaa kun eri osapuolet häiriötietoa päivittävät. ICT-häiriönhallinnan kautta yritys tulee pääsemään myös paremmin kiinni palvelutasosopimuksien valvontaan. Jokaisesta häiriöstä voidaan ajaa raportti, josta ilmenee vian ilmenemisaika aina korjaukseen asti. ICT-häiriönhallintajärjestelmään voidaan liittää tieto suoraan palvelutasosta, jolloin ne saadaan samalle raportille. Järjestelmän myötä yritys pystyy seuraamaan reaaliaikaisesti häiriöiden selvitystä ja niiden tilaa sekä SLA:n toteutumista.

Rakennettu valvonta- ja häiriöhallinnan kokonaisuus koostuu seuraavista osista:

- Actional sekä Oracle BAM – valvontatuotteet, joilla saadaan yrityksen palvelualustoilta tuotetut järjestelmät ja palvelut, mukaan lukien palveluväylän jonot valvontaan. Valvontaa

voidaan laajentaa tarpeen mukaan uusilla valvonta-agenteilla kaikkiin tarpeellisiin kohteisiin.

- ICT-häiriöhallintajärjestelmä muodostuu sääntömoottorista ja yhteiskäyttöisestä palveluväylästä. ICT-häiriöhallintajärjestelmään kuvataan tarpeelliset alustan, järjestelmien ja palveluiden virheilmoitukset sekä niiden käsittelysäännöt. Samoin ICT-häiriöhallinta tulee sisältämään virhekohtaisen vastaanottajan ja häiriötiedon lähetysmuodon (E-mail, SMS, XML tai jokin muu haluttu muoto). Tietoihin voidaan laittaa myös oletettu korjausaika (SLA). Häiriötietoa voidaan lähettää myös muille vastaanottajille kuten esimerkiksi helpdesk:lle, käyttöpalvelutoimittajan tivoli-järjestelmään ja muille tiedon vastaanottajille. ICT-häiriöhallintajärjestelmällä on oma portaalikäyttöliittymänsä, jonka kautta voidaan raportoida manuaalisesti virheistä sekä lisätä ja muokata kaikkia tarvittavia tietoja.
- Portaali-valvontakäyttöliittymä, jonka kautta nähdään reaaliajassa järjestelmien ja palveluiden toimivuuden kokonaistilanne. Käyttäjakohtaisesti räätälöidyllä etusivulla näytetään reaaliajassa ”liikennevaloin” ympäristön toimivuus. Käyttäjän on mahdollista nähdä suoraan värien perusteella, missä integraatiossa on ongelmia tiedon käsittelyssä. Liikennevalot toimivat linkkeinä valvontakohtaiseen näkymään, josta on mahdollista nähdä palvelukohtainen toiminta ja sitä kautta selvittää palvelusta vastaava toimittaja.

Toteutuneessa arkkitehtuurissa yhteiskäyttöisellä palveluväylällä on kaksi roolia:

- Toimia integraatiovälineenä eri sovellusten välillä ja
- Toimia osana ICT-häiriöhallintajärjestelmää

#### 8.4.5 Yhtenäinen CMDB

Configuration Management Database (CMDB) eli konfiguraationhallintatietokanta on tietokanta, jota käytetään säilyttämään konfiguraatietueet koko niiden elinkaaren ajan.

Konfiguraationhallintajärjestelmä ylläpitää yhtä tai useampaa konfiguraatietietokantaa, ja kukin tietokanta säilyttää rakennneosien attribuutit (ominaisuudet) ja suhteet toisiin rakennepsiin [Madduri *et al.*, 2010].

Configuration Item (CI) eli konfiguraation rakenneposa on mikä tahansa komponentti tai palveluomaisuus, jota täytyy hallita IT-palvelun toimittamisessa. Informaatio jokaisesta

konfiguraation rakenneosasta kirjataan konfiguraatitietueena konfiguraationhallintajärjestelmään, ja sitä ylläpidetään koko sen elinkaaren ajan konfiguraationhallintaprosessilla. CI:t ovat muutoksenhallinnan valvonnassa. CI:t ovat tyypillisesti palveluiden ja prosessien kuvauksia, IT-palveluja, reititys-sääntöjä, laitteistoja, ohjelmistoja, rakennuksia, ihmisiä ja virallisia asiakirjoja, kuten prosessidokumentaatio ja palvelutasosopimukset [Sauvé et al. 2006].

ICT-häiriönhallintajärjestelmässä CMDB:a käytetään kuvauskantana, jossa on kuvattuna kaikki palvelut ja palvelukomponentit riittävällä tarkkuudella. Komponentteihin on liitetty yksilölliset tunnisteet (identifier), mikä mahdollistaa kaikkien tapahtumien ja säännösten kytkemisen osaksi tiettyä palvelukomponenttia. Kytkemällä CMDB osaksi valvonta- ja ICT-häiriönhallintapalvelua voidaan yhdistää tapahtumat yksikäsitteisesti tiettyyn palvelukomponenttiin sekä luoda tunnisteisiin perustuva säännöstö- ja päättelylogiikka, jolla tapahtumia käsitellään ja niihin reagoidaan Jos yhtenäistä CMDB:a ei ole, joudutaan tekemään säännöstössä kuvauksia eri osapuolien käyttämien tunnisteiden välillä.

#### 8.4.6 Palvelukeskeinen arkkitehtuuri keskiössä

SOA-arkkitehtuuri koostuu yleensä eri teknologioilla toteutetuista järjestelmistä. Teknologisista ja sisällöllisistä eroista johtuen järjestelmien integrointi keskenään on hankalaa. Ratkaisukeinona tähän on toteuttaa tarvittava logiikka jokaiseen moduuliin erikseen, tai ottaa käyttöön keskitetty kommunikointiväylä, jossa pyritään ratkaisemaan nämä ongelmat [Papazoglou & van den Heuvel 2007]. Palveluväylä on yksi SOA-arkkitehtuurin keskeisimmistä komponenteista, jolla ratkaistiin vanhojen sekä ulkoisten järjestelmien integrointi projektissa. Palveluväylän päätehtävänä on arkkitehtuurin viestinvälitys järjestelmien välillä.

Projektissa pyrittiin ketteryyteen, jotta toiminnallisuudet olisivat mahdollisimman uudelleenkäytettäviä. Arkkitehtuuri suunniteltiin alun perin kerroksittain toteutettavaksi kokonaisuudeksi, jolloin voitiin luoda palvelukohtaisia kerroksia. Samalla luotiin standardit rajapinnat, jotta toiset järjestelmät integroituisivat osaksi järjestelmää. Rajapinnat huolehtivat tiedonvälityksestä sekä mahdollisista tiedon ja teknologian muunnoksista.

### 8.5 Mitä häiriönhallinnan arkkitehtuurilla saatiin aikaan?

Tapaustutkimuksen kohteena olevasta projektiin käyttöönotosta on kulunut aikaa vuosi ja tässä ajassa on saatu aikaan merkittävää palvelun laadun ja saatavuuden tehostumista. Lisäksi liiketoiminnan tasolla on saavutettu merkittäviä kustannussäästöjä sekä asiakastyytyväisyyttä.

Uuden arkkitehtuurin avulla häiriöilmoitusten selvittäminen on nopeutunut ja vastuussa olevan toimittajan tunnistaminen selkeytynyt sekä nopeutunut. Samalla uuden arkkitehtuurin myötä saatiin aiemmat vanhanaikaiset integraatiot mukaan valvonnan piiriin, jolloin niiden tukeminen on myös parantunut. Yritys on saanut solmittua uusia sopimuksia asiakkaiden kanssa ja uusia asiakkuuksia on syntynyt projektia edeltävää tilaa enemmän.

Alla on kuvaus häiriönhallintaprosessin etenemisestä uudessa valvonta-arkkitehtuurissa:

1. Häiriöilmoitus tuotetaan automaattisesti generoituna (ICT-häiriöhallintajärjestelmä) tai toimittajan tikettijärjestelmästä. Käyttäjän on myös mahdollista tehdä manuaalisesti häiriöilmoitus extranetin tai portaalin kautta.
2. Yhteiskäyttöinen palveluväylä välittää tiedot tarvittaville osapuolille halutulla tiedonsiirtoformaattilla. Jos esimerkiksi välitetään tieto XML -muodossa, on toimittajan mahdollista automatisoida oma tikettijärjestelmänsä siten, että saapuvasta viestistä perustetaan uusi tiketti tai päivitetään olemassa olevan tiketin tietoja.
3. Toimittajan reagointi (mahdolliset korjaukset, kommentoinnit tai siirto toiselle toimittajalle)
4. Toimittaja lähettää ICT-häiriönhallintajärjestelmään kuittauksen tehdyistä toimenpiteistä yhteneväisen rajapinnan kautta tai päivittää häiriön tilatiedon extranet-käyttöliittymän kautta.
5. Jos häiriö on korjattu, lähetetään tieto kaikille osapuolille, mikäli ei korjattu, edetään seuraavan säännön mukaisesti.

Asiakkaan on mahdollista nähdä jatkuvasti häiriön tilanne yhteiskäyttöisestä portaalista tai, jos asiakkaalla on käytössään oma tiketointijärjestelmänsä, hän voi tarkistaa ratkaisun etenemisen omasta järjestelmästänsä. Ratkaisun lähtökohtana on aina läpinäkyvyys asiakkaalle, jolla saavutetaan parempi asiakastyytyväisyys. Asiakas voi kommentoida ratkaisun etenemistä esimerkiksi sellaisessa tapauksessa, jos häiriö onkin ratkennut ja palvelu on jälleen saavutettavissa. Tällöin yritys ei sulje ratkaisutikettiä vaan selvittää, mistä häiriö johtui ja päivittää tiketille selvityksen tuloksen.

## 8.6 Ennakoimattomat vaikutukset ja mahdolliset parannusehdotukset

Kaikkia arkkitehtuurille asetettuja vaatimuksia ei ehditty toteuttamaan vaan osa vaatimuksista päätettiin siirtää seuraavaan julkaisuun, jolloin kokemuksia järjestelmän käytöstä on ehtinyt kertyä.

Samalla tunnistettiin muutamia ennakoimattomia vaikutuksia, jotka selvisivät vasta järjestelmän testaamisvaiheessa.

Yksi tärkeimmistä ennakoimattomista vaikutuksista oli päällekkäisten häiriöilmoitusten ja toimenpidepyyntöjen estäminen palveluketjussa. Tällainen tilanne saattoi tulla kyseeseen silloin, kun järjestelmä huomasi häiriön palvelussa ja loi häiriöilmoituksen automaattisesti. Samalla asiakas saattoi huomata häiriön palvelussaan ja luoda samankaltaisen ilmoituksen. Tätä skenaariota ei ollut otettu huomioon järjestelmän suunnittelussa, mutta se oli kirjattu ylös tarkasteltavana mahdollisena parannusehdotuksena. Vaikka kyseessä oli ehdotus, päätettiin se kuitenkin toteuttaa ja testata vielä ennen varsinaisen tuotantoversion julkaisua. Ehdotus oli tärkeä, koska se olisi vaikuttanut asiakastyytyväisyyteen.

Alla on kuvaus tehdystä korjauksesta:

Päällekkäisten häiriöilmoitusten ja toimenpidepyyntöjen estämiseksi lähetetään vain yksi ilmoitus palveluketjussa prioriteetissa korkeimmalle olevalle häiriölle. Palvelualustan häiriöprioriteetit menevät aina sovellus- ja järjestelmäprioriteetin ohi. Tämä sen takia että alustan toimivuudesta ei lähetetä turhaan toimenpidepyyntöjä ylemmälle kerrokselle. Palveluketjussa vastaavasti tiedon kulkusuunnassa lähempänä tiedon lähettävää päätä (lähdettä) on yleensä merkitsevempi kuin ketjussa myöhempana oleva häiriö. Kummassakin tilanteessa kaikille osapuolille lähetetään kuitenkin tieto häiriöistä, että käyttöpalvelutoimittaja ja järjestelmä- ja palvelutoimittajat voivat mahdollisesti varautua tekemään jotain toimenpiteitä tai tiedon loppukäyttäjää osaa varautua, että tieto ei kulje tai kulkee vain osittain häiriön takia. Raportoidusta häiriöstä pidetään tilatietoa, jolloin häiriöjärjestelmä tietää sen kenelle häiriöstä on lähtenyt toimenpidepyyntö ja kenelle tiedoksi. Mikäli joku palveluketjussa päivittää tilatietoa esimerkiksi että häiriö on korjattu, lähtee siitä välittömästi tieto kaikille muille palveluketjussa toimijoille.

Jo alkuun päätettiin, että projektin ensimmäisessä vaiheessa ei oteta huomioon laitteistopuolen häiriöitä vaan niitä tarkastellaan ja tehdään korjaustoimenpiteet. Laitteistopuolen häiriöt kuitenkin kirjataan ylös ja selvitetään, kuinka näiden häiriöiden selvittäminen voitaisiin ottaa mukaan valvonta-arkkitehtuuriin. Järjestelmää testattaessa huomattiin, että valitut valvontatyökalut tukevat myös laitteistopuolen valvontaa, joka kirjattiin parannusehdotukseksi seuraavaan versioon.

Vaikka työkalujen päällekkäisiä ominaisuuksia ja valvonta-ominaisuuksia selvitettiin etukäteen, huomattiin, että työkalujen konfiguroinnit ja asennukset on tehtävä ohjelmisto-yrityksen omilla työkaluilla. Käytössä on sekä Oraclen että Progress:n työkaluja, mutta molempien toimittajien

järjestelmävalvojan työt oli tehtävä heidän tarjoamillaan työkaluilla. Tämä kuitenkin todettiin hyvin vähäiseksi ongelmaksi, koska Oraclen ja Progress:n valvontatyökalut valvoivat eri toimintoja.

## 9. Yhteenveto

Palvelukeskeisen arkkitehtuurin ei ole tarkoitus toimia täydellisenä ratkaisuna nykypäivän monitoimittajaympäristöjen häiriöiden hallintaan, vaan se tuo mahdolliseksi hyödyntää aikaisempien arkkitehtuurien ja kehitystapojen parhaimmat osat. Palvelukeskeinen arkkitehtuuri voidaan ajatella olevan kehityksen seuraava looginen kehitysaskel, jonka hyödyt ovat nähtävissä niin liiketoiminnassa kuin järjestelmätasollakin. Nykyaikaisten yritysten verkostoitumisen haasteet on mahdollista ratkaista palvelukeskeisen arkkitehtuurin ja integroimisen myötä.

Palvelukeskeinen arkkitehtuuri sisältää myös riskejä, joilla voi olla suuret yritystä vahingoittavat vaikutukset ja siksi perusteellinen toimintasuunnitelma riskeineen tulee olla tehtynä ennen palvelukeskeiseen arkkitehtuuriin siirtymistä. Yleisesti suurin kompastuskivi palvelukeskeiseen arkkitehtuuriin siirtymisessä on koko kokonaisuuden hankkiminen kerralla ja pyrkimys suoraan mahdollisimman hyvään ratkaisuun. Tässä tutkielmassa esiintyvällä yrityksellä oli jo aiempaa kokemusta palvelukeskeisen arkkitehtuurin suunnittelusta, toteuttamisesta ja järjestelmien migraatiosta. Tästä syystä yritys ei lähtenyt suoraan muuttamaan olemassa olevaa järjestelmää vaan sen osia muutettiin migraatiovaiheessa niin, että myös vanhat järjestelmät toimivat rinnalla.

Yksi palvelukeskeisen arkkitehtuurin tärkeimmistä osista on SOA-hallinta, jonka avulla on mahdollista edistää palvelukeskeisyyttä. Tutkielman keskiössä olevassa yrityksessä IT-infrastruktuurin taso oli SOA:n kannalta teknisesti korkealla, mutta sen hallintomalli oli hajanainen. Ajan myötä yritys on saanut tekniikkaa rakennettua ja samalla riskejä minimoitua käyttökokemusten kartuttua.

Tässä tutkielmassa käytettiin tapaustutkimusta, jonka keskiössä oli eräs keskisuuri ohjelmistoyritys, jossa oli käynnissä uudenlainen integraatioprojekti. Tällä pyrittiin täyttämään yrityksen palvelusalkkua niin, että yritys saavuttaisi merkittävää kilpailuetua muihin saman alan yrityksiin nähden. Uudella arkkitehtuurilla pystyttiin vastaamaan markkinoiden haasteisiin häiriönhallinnan osalta sekä tarjoamaan laadukkaampaa palvelua vanhoille ja uusille asiakkaille.

Tutkielmassa ei ollut mahdollista laajempaan kvantitatiiviseen tutkimukseen, koska integraatioprojektin myötä luotu uusi palveluarkkitehtuuri ei ole ollut käytössä pitkään. Näin ollen laadukasta ja soveltuva aineistoa on saatavilla vasta useamman vuoden jälkeen käyttöönotosta. Jatkotutkimuksen kannalta olisi oleellista selvittää, onnistuiko integraatioprojekti täyttämään sille

asetetut tavoitteet ja haasteet. Kuluneen vuoden aikana on kuitenkin saatu jo merkittäviä kustannussäästöjä sekä huomattu asiakkaan toimesta huomattavaa palvelun laadun parantumista. Lisäksi yritys on onnistunut saamaan uusia asiakkaita sekä voittamaan tarjouskilpailuja uuden valvonta-arkkitehtuurin myötä.



## Viiteluettelo

- [Bapna et al., 2010] Bapna, R., Barua, A., Mani, D. & Mehra, A. (2010). Research commentary— Cooperation, coordination, and governance in multisourcing: An agenda for analytical and empirical research. *Information Systems Research*, 21(4), 785-795.
- [Bartolini et al., 2012]. Bartolini, C., Erbes, J., Graupner, S. & Nezhad, H. R. M. (2012). A process- and policy-aware cross enterprise collaboration framework for multisourced services. Teoksessa *SRII Global Conference (SRII), 2012 Annual* (s. 488-493). Los Alamitos, CA: IEEE Computer Society.
- [Bartolini et al., 2010] Bartolini, C., Stefanelli, C. & Tortonesi, M. (2010). Business-impact analysis and simulation of critical incidents in IT service management. Teoksessa *IFIP/IEEE International Symposium on Integrated Network Management, IM'09* (s. 9-16). Los Alamitos, CA: IEEE Computer Society.
- [Bennet, 1995] Keith Bennet, Legacy systems: Coping with success. *IEEE Software*, 12 (1995), 19–23.
- [Bussler, 2010] Christoph Bussler, *B2B Integration*. Springer, 2010.
- [Buxman and Kaiser, 2012] Buxmann, P. and Kaiser, J. (2012). Organizational design of IT supplier relationship management: a multiple case study of five client companies. *Journal of Information Technology*, 27 (2012), 57-73.
- [Cassel and Symon, 2004] Catherine Cassell and Gillian Symon, *Essential Guide to Qualitative Methods in Organisational Research*. Sage, 2004.
- [Cohen and Young, 2006] Cohen, L. and Young, A. *Multisourcing: Moving beyond outsourcing to achieve growth and agility*. Cambridge: Harvard Business Press, 2006.
- [Cumberlidge, 2007] Matt Cumberlidge. *Business Process Management with Jboss JBPM*. Packt Publishing, 2007.
- [DeLone and McLean, 1992] William DeLone and Ephraim McLean, Information systems success: The quest for the dependent variable. *Information Systems Research*. 3 (1992), 60-95.

- [DeLone and McLean, 2003] William DeLone and Ephraim McLean, The DeLone and McLean model of information systems success: A ten-year update. *Journal of Management Information Systems* **19** (2003), 9 - 30.
- [Erl, 2008] Thomas Erl, *Service-Oriented Architecture – Concepts, Technology, and Design*. Prentice Hall, 2008.
- [Geraci *et al.*, 1991] Geraci, A., Katki, F., McMonegal, L., Meyer, B., Lane, J., Wilson, P., Radatz, J., Yee, M., Porteous, H. and Springsteel, F. 1991. IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries, *The Institute of Electrical and Electronics Engineers Inc.*
- [Grönroos, 2010] Christian Grönroos, *Palvelujen johtaminen ja markkinointi*. WSOYpro, Helsinki, 2010.
- [Hailstone *et al.*, 2007] Hailstone, R., Illsley, R., Jones, T. & Kellett A. 2007. *SOA Platforms– Software Infrastructure Requirements for Successful SOA Deployments*. Butler Group, Butler Direct Ltd., 2007.
- [Hevner *et al.*, 2004] A.R. Hevner, S.T. March, J. Park and S. Ram, Design science in information systems research. *MIS Quarterly* **28** (2004), 75-105.
- [Hohpe, 2012] Gregor Hohpe, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Pearson Education, 2012
- [Hyötyläinen, 2013] Tahvo Hyötyläinen, *Path to Improved Firm Performance with Business Process Management (BPM) and BPM Systems*. Väitöskirja, Luonnontieteiden ala, Tampereen yliopisto, 2013.
- [ITIL, 2011] IT Infrastructure Library. *ITIL Service Operation*. (2011 painos). Iso-Britannia: The Stationery Office (TSO).
- [Josuttis, 2007] Nicolai M. Josuttis, *SOA in Practice, the Art of Distributed System Design*, O'Reilly Media, 2007.
- [Järvinen ja Järvinen, 1996] Pertti Järvinen ja Annikki Järvinen, *Tutkimustyön metodeista*. Opinaja, Tampere, 1996.
- [Kajastila, 2012] Joonas Kajastila, *Palvelukeskeisen arkkitehtuurin hallinta – tapaus Liikennevirasto*. Opinnäytetyö, Tekniikan ala, Tietotekniikan koulutusohjelma, Lahden ammattikorkeakoulu, 2012.

- [Madduri *et al.*, 2010] Madduri H., Shi S.S.B., Baker R., Ayachitula N., A configuration management database architecture in support of IBM Service Management *The Institute of Electrical and Electronics Engineers Inc* **46** (2010), 441-457.
- [March and Smith, 1995] March, S. Smith, F. Design and Natural Science Research on Information Technology. *Decision Support Systems* **4** (1995), 251-266.
- [Mykkänen *et al.*, 2003] Juha Mykkänen, Jari Porasmaa, Juha Rannanheimo ja Mikko Korpela, A process for specifying integration for multi-tier applications in healthcare. *International Journal of Medical Informatics* **70** (2003), 172-182.
- [Papazoglou and van den Heuvel, 2007] Papazoglou, M. and van den Heuvel, W-J. 2007. Service oriented architectures: approaches, technologies and research issues. Springer, Heidelberg. The VLDB Journal, Volume 16, Number 3, 389-415. Saatavilla: <http://www.springerlink.com/content/d2228066725332u2/fulltext.pdf> (luettu 15.4.2015).
- [Pollock, 2001] Pollock, J.T., 2001, The Big Issue: Interoperability vs. Integration International Organization for Standardization [Online]. Saatavilla: [http://isotc.iso.org/livelink/livelink/fetch/1287650/2426503/XML\\_semantic\\_interoperability-Integration\\_Pollock\\_1.pdf?nodeid=2425856&vernum=0](http://isotc.iso.org/livelink/livelink/fetch/1287650/2426503/XML_semantic_interoperability-Integration_Pollock_1.pdf?nodeid=2425856&vernum=0) (luettu 20.01.2015).
- [Sauvé *et al.*, 2006] Business-Driven Decision Support for Change Management: *Planning and Scheduling of Changes* 173-184, Springer Berlin Heidelberg, Berliini, 2006.
- [Schön, 1983] Donald A. Schön, *The reflective practitioner, how professionals think in action*. Basic Book Inc., 1983
- [Trowbridge *et al.*, 2004] David Trowbridge, Ulrich Roxburgh, Gregor Hoppe, Dragos Manolescu and E.G Nadhan, *Integration Patterns*. Microsoft, 2004.
- [Tähtinen, 2005] Sami Tähtinen, *Järjestelmäintegraatio – Tarve, vaihtoehdot, toteutus*. Talentum, Helsinki, 2005.
- [van der Aalst *et al.*, 2003] Wil van Der Aalst, Arthur H.M. Hofstede, Bartek Kiepuszewski, and Alistair P. Barros. Workflow Patterns In: *Distributed and Parallel Databases* **14** (2003), 5–51.
- [Wisniewski, 2001] Mik Wisniewski, Using SERVQUAL to assess customer satisfaction with public sector services. *Managing Service Quality* **11** (2001), 380-388.

[Yin, 1989] Robert K. Yin, *Case Study Research: Design and Methods*. SAGE Publications, Beverly Hills Ca., 1989

## Liitteet

Taulukko 1. Monitoimittajaympäristön haasteet ja niiden ratkaisut

Haaste	Ratkaisu
Ulkoistettujen toimintojen keskinäisten riippuvuuksien ymmärtäminen.	Päätöksenteon rajoitetun rationalisuuden (bounded rationality) huomioiminen [Bartolini <i>et al.</i> , 2012]  Arvoverkostomallinnus [Buxman and Kaiser, 2012]
Luottamuksen luominen sekä toimittajien ja asiakkaan että yhteistyötä tekevien toimittajien välille [Bartolini <i>et al.</i> , 2012]	Jaettavan tiedon hallinta ja suojaaminen [Bartolini <i>et al.</i> , 2012]  Toimittajien ja asiakkaan välinen yhteistyö prosessitasolla [Bartolini <i>et al.</i> , 2012]
Yritysrajoja ylittävien monimutkaisten suhteiden ja yhteistyön johtaminen [Bartolini <i>et al.</i> , 2012 ja Bapna <i>et al.</i> , 2010]	Toimittajien ja asiakkaan välinen yhteistyö prosessitasolla [Bartolini <i>et al.</i> , 2012]  Jaettavan tiedon hallinta ja suojaaminen [Bartolini <i>et al.</i> , 2012]  Ulkoistamissuhteiden keskitetty johtaminen [Buxman and Kaiser, 2012]  Arvoverkostomallinnus [Buxman and Kaiser, 2012]
Toimittajien toimintojen koordinointi ja niiden linjaus liiketoiminnan tavoitteisiin [Bapna <i>et al.</i> , 2010]	Ulkoistamissuhteiden keskitetty johtaminen [Buxman and Kaiser, 2012]  Keskinäisten riippuvuuksien parempi ymmärrys [Bapna <i>et al.</i> , 2010]

	Arvoverkostomallinnus [Bartolini <i>et al.</i> , 2012]
Oikeiden kannustimien tarjoaminen toimittajien yhteistyön motivoimiseksi [Bapna <i>et al.</i> , 2010]	Keskinäisten riippuvuuksien parempi ymmärrys [Bapna <i>et al.</i> , 2010]

## Taulukko 2. Tyypillisiä ICT-palvelujen laatumääreitä

Englanninkieliset sanat ovat ITIL:ssä käytettäviä termejä [ITIL, 2011].

<b>Termi</b>	<b>Selitys</b>
Asiakastyytyväisyys	Mittaa asiakkaan tietohallinnon (tilaaja) avainhenkilöiden tyytyväisyyttä palveluun ja palveluyhteistyöhön.
Kapasiteetti	Hyvin laaja käsite. Voi tarkoittaa tietoliikennekapasiteettia (läpäisykyky), tallennuskapasiteettia, tietokantakapasiteettia, varmistuskapasiteettia, prosessorikapasiteettia, laitetilan laitepaikkakapasiteettia, laitetilan jäähdytyskapasiteettia tms. Jopa asiantuntijaresurssien minimimäärää voidaan pitää kapasiteettia koskevana laatumääreenä.
Katkojen maksimilukumäärä	Suurin määritellyllä tarkasteluvälillä palveluaikana tapahtuvien katkojen lukumäärä (tarkasteluvälin katkojen summa).
Koulutustaso, osaamistaso	Jatkuvien palveluiden ja asiantuntijapalveluiden toteuttajien koulutukselle ja osaamiselle (sertifikaatit) voidaan asettaa minimiehtoja tai tavoitteita.
Käytettävyys, saatavuus	Availability. Käytettävyydellä tarkoitetaan kohteena olevan laitteen, palvelun päälläoloa ja kykyä tuottaa sitä palvelua, jota kohteelta edellytetään sovitun toiminnon suorittamiseksi vaadittuna aikana. Käytettävyys lasketaan vähentämällä käyttökatojen aika ideaalikäytettävyydestä palveluaikana. Erityisesti teknologia-alustan palveluissa käytettävä laatumääre.
Käyttäjätyytyväisyys	Mittaa loppukäyttäjien tyytyväisyyttä palveluun tai palvelukokonaisuuteen.

Maksimikatko	Pisin yksittäinen yhtämittainen palvelukatko, joka sallitaan palvelun palvelutasotavoitteiden puitteissa palveluaikana sovitulla tarkasteluvälillä. Esimerkiksi 2 tuntia yhden kalenterikuukauden aikana.
Palveluaika	Service hours. Sovittu aikaväli, jolloin asiakkaalle tai palvelun kohteelle tuotetaan palvelukuvauksen mukaista palvelua. Esimerkiksi arkisin kello 08 – 16:00. Hyvin yleinen laatumääre lähes kaikissa palveluissa.
Ratkaisuaika (palvelupyynnöt, häiriöt)	Ratkaisuaika kuvaa, miten nopeasti häiriötä tai vikatilannetta koskeva tapahtuma tai palveluntuottajan itse havaitsema häiriö tulee saada korjattua ja tilanne normalisoitua.
Ratkaisukyky	Kuvaa, miten tehokkaasti palvelupyynnön vastaanottava taho (Esimerkiksi service desk) pystyy itsenäisesti ratkaisemaan sille tulevia palvelupyynnön (siirtämättä/ohjaamatta palvelupyynnön eteenpäin muille tukitasoille/palvelujonoille).
Reagointiaika (palvelupyynnöihin, häiriöihin)	Response Time. (Maksimi) aika, jonka kuluessa tapahtuman tai häiriön havaitsemisesta tulee häiriön korjaaminen tai tapahtuman käsittely aloittaa. Reagointiaika riippuu yleensä häiriön kriisyysluokasta.
Suorituskyky	Hyvin laaja käsite. Voi tarkoittaa laitteiden tai jopa niiden osien (esimerkiksi prosessorin laskentateho) suorituskykyä tai kokonaisten järjestelmien tai niiden komponenttien suorituskykyä (kuinka monta sanomaa integraatoratkaisu välittää minuutissa tai miten nopeasti talousraportti muodostetaan tietystä aineistosta).



Tavoitettavuus	Tavoitettavuudella tarkoitetaan palveluntuottajan palvelupisteen (Service desk, Help desk) kykyä vastata sovituksessa ajassa sinne tuleviin palvelupyyntöihin. Tyypillisesti tavoitettavuus koskee puhelinpalvelua ja määritetään keskimääräisenä tavoitettavuutena.
Toimitusaika	Aika, jonka kuluessa palvelupyynnön jättämisestä tai tilauksesta tilattu palvelu/tehtävä tulee toimittaa/toteuttaa. Vertaa ratkaisuaika, jonka kuluessa häiriö tai ongelma tulee poistaa. Erittäin yleinen laatumääre.

### Taulukko 3. Palvelukeskeisyyden periaatteet

Englanninkieliset sanat ovat lähteen käyttämiä termejä periaatteille [Erl, 2008].

<b>Periaate</b>	<b>Määritelmä</b>
Autonomia, itsehallinto (Autonomy)	Palveluiden sisältämä (encapsulate) logiikka on niiden omassa hallinnassa.
Palvelusopimus (Service contract)	Palvelut noudattavat kommunikaatiosopimusta, jonka määrittävät yksi tai useampi palvelukuvaus ja näihin liittyvät dokumentit.
Löyhät kytkennät (Loose coupling)	Palveluilla on olemassa suhde, jossa on mahdollisimman vähäinen määrä riippuvuuksia. Tämä suhde edellyttää vain, että palvelut ovat vain tietoisina toisistaan.
Abstrahointi (Abstraction)	Logiikka on palveluiden ulkopuolisilta piilottamaa, jota ei ole kuvattu palvelukuvauksessa.
Tilattomuus (Statelessness)	Palvelut pyrkivät säilyttämään mahdollisimman vähän aktiviteetteihin liittyvää informaatiota.
Yhdisteltävyys (Composability)	Palveluita voidaan yhdistää sekä koordinoida komposiittipalveluiksi (Composite services).
Löydettävyys (Discoverability)	Palvelut tulee suunnitella ulospäin kuvaaviksi, jotta niiden löytäminen ja arviointi saatavilla olevien hakumekanismien avulla on mahdollista.

## Taulukko 4. ESB:n tärkeimmät ominaisuudet

Ominaisuudet on poimittu lähdekirjasta [Papazoglou and van den Heuvel, 2007].

<b>Periaate</b>	<b>Määritelmä</b>
Dynaminen liitettävyys	Palveluita on mahdollista liittää toisiinsa dynaamisesti toteutusprotokollasta riippumatta.
Olemassa olevien toteutusten hyödyntäminen	Pyrittävä hyödyntämään jo olemassa olevia perinnejärjestelmiä mahdollisimman paljon.
Palveluiden välinen kommunikointi	Protokollariippumaton kommunikointi.
Tiedon muuntaminen	Kyky muuntaa viestejä erilaisiin formaatteihin, liitettäessä useita palveluita ESB:iin.
Aihe- ja sisältöpohjainen reitittäminen	Viestejä voidaan reitittää sekä aihe- että sisältöpohjaisesti.
Palvelun löydettävyys ja palvelutaso	Eri liiketoimintasisältöisiä palveluita on mahdollista paikallistamaan ja kytkeään toisiinsa. Eri palvelutasot mahdollistavat parhaan palveluinstanssin käytön.
Tietoturva	Tietoturvan on oltava toimiva palveluiden kesken ja käyttäjille on tarjottava useita vaihtoehtoja tietoturvan varmistamiseen.
Luotettava viestinvälitys	Viestien toimittaminen vastaanottajalle varmasti.
Skaalautuvuus	Skaalautuvuus auttaa muunnoskyvykkyyden toteuttamisessa, koska muunnosten käsittely tarvitsee paljon resursseja.
Hallinta ja monitorointi	Palveluiden hallinta ja valvonta on kyettävä toteuttamaan organisaatorajojen yli.

Kuva 17. Projektissa toteutunut valvonta-arkkitehtuuri

