

Juha Mykkänen, Timo Itälä, Saara Savolainen, Hannu Virkanen

Yhteentoimivuus, standardit ja palveluarkkitehtuuri

SOLEA-hanke
Itä-Suomen yliopisto
Aalto-yliopisto



Juha Mykkänen, Timo Itälä, Saara Savolainen, Hannu Virkanen

Yhteentoimivuus, standardit ja palveluarkkitehtuuri

Itä-Suomen yliopisto ja Aalto-yliopisto
Kuopio
2012



©Itä-Suomen yliopisto ja Aalto-yliopisto 2012
SOLEA-hanke
<http://www.uef.fi/solea>

ISBN 978-952-61-0694-6 (PDF)

Juha Mykkänen, Timo Itälä, Saara Savolainen, Hannu Virkanen. Yhteentoimivuus, standardit ja palveluarkkitehtuuri. Itä-Suomen yliopisto ja Aalto-yliopisto. 2012. 69 s.
ISBN 978-952-61-0694-6 (PDF)

Tiivistelmä

Tietojärjestelmien yhteentoimivuus on yhä enemmän elinehto verkottuvien organisaatioiden toiminnassa. Järjestelmien integrointitoimenpiteet ja yhteentoimivuusvaatimukset ovat siirtymässä organisaatioiden sisältä yhä enemmän yksiköiden ja organisaatioiden välisten prosessien ja yhteistoimintaverkostojen tietoteknisen tuen mahdollistamiseen. Nopea mukautuvuus toimintaympäristön ja asiakkaiden tarpeiden muutokseen edellyttää ketteryyttä, joka pohjimmiltaan on toimintojen ja tietojen mukauttamista uusiin tilanteisiin. Tämä puolestaan edellyttää tietojärjestelmäkokonaisuuksilta mukautuvuutta.

Palvelukeskeisyys (service orientation) tarjoaa sekä organisaatioiden että tietojärjestelmien kehittämisessä mallin, jossa toiminta ja tietojärjestelmät jäsennetään ulkoisesti palvelujen käyttäjilleen täyttämien tarpeiden ja tulosten ja sisäisesti palvelujen tuottamiseen käytettyjen ratkaisujen kautta. Palveluarkkitehtuuri (SOA) korostaa organisaatioiden tietojärjestelmäkehityksessä uudelleenkäytettävien sovellus- tai tietojärjestelmäpalvelujen hyödyntämistä ja suunnittelua siten, että palvelut ovat itsenäisiä ja hyödynnettävissä useissa eri sovelluksissa tai prosesseissa. Rajapinnat ja yhteentoimivuus ovat keskeisiä palveluarkkitehtuurin kulmakiviä.

Tietojärjestelmien yhteentoimivuutta tukemaan on tuotettu suuri joukko teknisiä ja sisällöllisiä standardeja. Standardien avulla voidaan ratkaista monia tietojärjestelmien ja palvelujen yhteensovittamisen haasteita, mutta on tärkeää myös tunnistaa mihin yhteentoimivuuksiin kukin standardi ottaa kantaa ja mitkä on ratkaistava tai sovittava tapauskohtaisesti. Monet yhteentoimivuusstandardit pohjautuvat muihin kuin palvelupohjaiseen ajattelutapaan, ja standardien soveltaminen tai tuottaminen palveluarkkitehtuurin yhteydessä edellyttää niiden suhteuttamista palvelukeskeisen kehittämistavan periaatteisiin.

Tässä dokumentissa on koottu yhteen SOLEA-projektissa vuosina 2008-2012 hyödynnettyjä malleja palveluarkkitehtuurin (SOA) ja yhteentoimivuuden kannalta. Dokumentissa esitettävät mallit perustuvat kirjallisuuslähteisiin, projektin työpajoissa sekä siihen liittyvissä tapahtumissa käsiteltyihin työpapereihin ja esityksiin sekä yhteistyöhön projektin osapuolten sekä kumppaniprojektien sekä yhteistyöorganisaatioiden kanssa. Dokumentti pyrkii kokoamaan joukon hyödyllisiä malleja ja esimerkkejä palvelukeskeisyyden soveltamiseen integrointiratkaisujen, standardoinnin sekä standardien hyödyntämisen eri vaiheissa. Mukana on tarpeiden määrittelyyn ja kuvaamiseen, ratkaisujen etsimiseen ja määrittelyyn sekä rajapintojen ja infrastruktuurin suunnittelupäätöksiin liittyvää sisältöä sekä viittauksia tarkempaan materiaaliin. Esimerkit painottuvat terveydenhuoltoon. Lisäksi dokumentissa tunnistetaan yhteentoimivuuden ja palvelupohjaisen integraation jatkotutkimuskohteita.

Luokitus (UDK): tietotekniikka 004, internet 654 & 004, tietojenkäsittely 004

Asiasanat (YSA): tiedonhallintajärjestelmät, verkkopalvelut, järjestelmäarkkitehtuuri, ohjelmistokehitys, standardit, ohjelmistotuotanto, palvelujärjestelmät, prosessit, palveluliiketoiminta, tiedonsiirto -- organisaatiot, hajautetut järjestelmät, hankinta, toiminnanohjaus, systeemityö, protokollat, standardointi, tietovarannot

Sisällys

1	Johdanto	7
1.1	Dokumentin tavoitteet.....	7
1.2	Yhteentoimivuuden ja standardien merkitys.....	8
1.3	Yhteentoimivuus kokonaisarkkitehtuurissa	8
2	Yhteentoimivuuden viitemalleja	12
2.1	European Interoperability Framework, versio 2	12
2.2	Integrointitapojen luokittelu.....	13
2.3	Seitsentasoinen yhteentoimivuuden viitemalli.....	14
3	Integraatioarkkitehtuurit ja SOA	18
3.1	Integraation topologia	18
3.2	Adapterit.....	20
3.3	Palveluväylä ja sen vaikutukset	21
4	SOA ja standardit	24
4.1	Palvelukeskeinen yhteentoimivuuden standardointi.....	24
4.2	Palveluarkkitehtuuriin liittyvien standardien sijoittuminen integrointitasoille.....	25
4.3	Toimialastandardien ja palveluarkkitehtuurin yhteensovittamisen haasteet.....	27
4.4	Standardien arviointi ja valinta	31
5	Esimerkkejä palvelukeskeisyydestä integraatiossa ja standardeissa	33
5.1	Palvelupohjaisuus SOLEA palvelutapahtumat -työkohteessa	33
5.2	Tapahtumakeskeinen arkkitehtuuri Satakunnan sairaanhoitopiirissä	37
5.3	KanTa-liityntäpisteen toteuttaminen.....	41
5.4	HL7 Service-Aware Interoperability Framework (SAIF).....	46
5.5	Healthcare Services Specification Project (HSSP)	49
5.6	IHE-profiilien hyödyntäminen palveluarkkitehtuurissa.....	51
6	Yhteenveto	55
	Lähteet	58
	Liite 1: Sanasto SOLEA-hankkeen keskeisistä käsitteistä	61

1 Johdanto

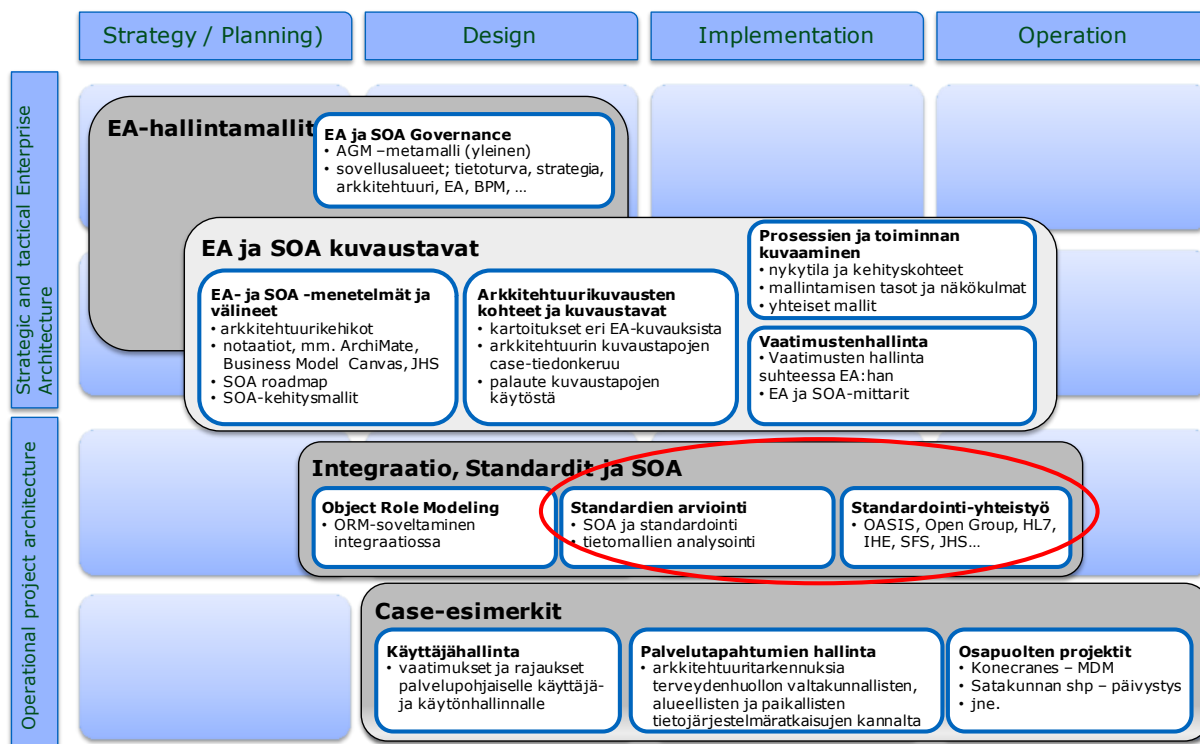
1.1 Dokumentin tavoitteet

Tämä dokumentti kokoaa SOLEA-hankkeessa tehtyä työtä palvelukeskeiseen arkkitehtuuriin ja erityisesti sen yhteentoimivusratkaisuihin liittyen. Sisältö perustuu kirjallisuuslähteiden lisäksi useisiin eri työpapereihin ja esityksiin sekä työpajoihin ja yhteistyöhön projektin osapuolten sekä kumppaniprojektien sekä yhteistyöorganisaatioiden kanssa.

Dokumentti on tarkoitettu erityisesti tietohallinnon, kokonaisarkkitehtuurin, hankintojen ja vaatimusmäärittelyjen parissa toimiville asiantuntijoille, jotka toimivat arkkitehtuurin ja integraatioiden kehittämisessä. Kokonaisarkkitehtuurissa yhteentoimivuuden kuvaukset ovat mm. tietohallintolain myötä nousemassa keskeisiksi kuvauksiksi. Palveluarkkitehtuurissa (SOA) yhteentoimivuuks korostuu, koska arkkitehtuurin periaatteena on uudelleenkäytettävien ohjelmistopalvelujen yhdistely.

Dokumentti pyrkii kokoamaan joukon hyödyllisiä malleja ja esimerkkejä siihen, kuinka palvelukeskeisyyttä (service orientation) sovelletaan integrointiratkaisujen ja standardoinnin sekä standardien hyödyntämisen toteuttamisen eri vaiheissa. Mukana on tarpeiden määrittelyyn ja kuvaamiseen, ratkaisujen etsimiseen ja määrittelyyn sekä rajapintojen ja infrastruktuurin suunnittelupäätöksiin liittyvää sisältöä sekä viittauksia tarkempaan materiaaliin. Lisäksi dokumentissa tunnistetaan yhteentoimivuuden ja palvelupohjaisen integraation jatkotutkimuskohteita.

Dokumentissa ei käsitellä yksittäisiä standardeja syvällisesti. Dokumentti ei myöskään syvenny kokonaisarkkitehtuurin yleisiin malleihin ja menetelmiin vaan keskittyy palvelukeskeisen arkkitehtuurin ja yhteentoimivusratkaisujen läpikäyntiin integraationäkökulmasta.



Kuva 1. Dokumentin kohdealue.

Tekijät kiittävät projektin osallistujia, erityisesti tässä dokumentissa käsiteltyjen palvelutapahtumat -työkohteen työhön osallistuneita lukuisia henkilöitä eri organisaatioista sekä muiden case-esimerkkien osallistujia Satakunnan sairaanhoitopiiristä sekä Intersystems Oy:stä.

1.2 Yhteentoimivuuden ja standardien merkitys

Tietojärjestelmien yhteentoimivuus on yhä enemmän elinehto verkottuvien organisaatioiden toiminnassa. Järjestelmäintegraation painopiste on siirtynyt viime vuosina vähitellen organisaatioiden sisältä yhä enemmän yksiköiden ja organisaatioiden välisten prosessien ja yhteistoimintaverkostojen tietoteknisen tuen mahdollistamiseen. Nopea mukautuvuus toimintaympäristön ja asiakkaiden tarpeiden muutoksiin edellyttää ketteryyttä, joka pohjimmiltaan on toimintojen ja tietojen mukautamista uusiin tilanteisiin. Tämä puolestaan edellyttää tietojärjestelmäkokonaisuuksilta mukautuvuutta.

Tiedonhallinnassa käytettävien välineiden ja järjestelmien hankintavaihtoehdot ovat monipuolistuneet jatkuvasti 70-luvulta lähtien. Aiempien "osta tai rakenna" vaihtoehtojen lisäksi suunnittelun ulkoistaminen, komponentti- ja palvelupohjaiset järjestelmien kokoamistavat, ASP sekä pilvipalvelut ovat nykyisin realistisia vaihtoehtoja organisaatioiden ja käyttäjien tietoteknisten tarpeiden täyttämiseen. Aiemmin tietohallinnossa päävaihtoehtoina voitiin pitää kokonaisvaltaisen toiminnanohjausjärjestelmän hankintaa ja tarvittaessa mukauttamista tai räätälöidyn ratkaisun hankkimista tai kehittämistä. Nykyisin näiden vaihtoehtojen välissä on joukko erilaisia tietotekniikka- ja sovelluskehitysmalleja, joille yhteistä on kuitenkin se, että eri lähteistä tulevia, eri toimintoihin tarkoitettuja ja eri-ikäisiä kokonaisjärjestelmän osia ja niiden tuottamia tietoja täytyy pystyä yhdistelemään, jotta toiminnan tarpeet voidaan täyttää. Tämä edellyttää rajapintoja ja liittymiä järjestelmien välille. Liittymissä on sovittava monista teknisistä ja sisällöllisistä seikoista sekä tietoja tuottavien että niitä hyödyntävien järjestelmien välillä.

Kahdenväliset liittymät ja rajapinnat järjestelmien välillä ovat kuitenkin kalliita suunnitella ja vaikeita ylläpitää. Koska monissa organisaatioissa on runsaasti samantyyppisiä tarpeita, on järkevää vakioita ja standardoida usein ja monien tahojen tarvitsemia ratkaisujen osia.

Sovellusintegraation perinteinen lähestymistapa on erikseen toteutettujen sovellusten tai tietojärjestelmäpalvelujen integrointi tiettyjä tarkasti määriteltyjä tilanteita varten. Tämän tyyppisessä alhaalta-ylös integraatiossa ohjelmistojen valmiit rajapinnat ja ohjelmistojen sisäisten ominaisuuksien ja tietomallien välisten vastaavuuksien rakentaminen ovat keskeisiä lähtökohtia. Prosessilähtöisen ratkaisujen määrittelyn ja myös kokonaisarkkitehtuurin lähtökohtana on ylhäältä alas ohjautuva, strategiasta sekä toiminnan ja prosessien tavoitteista lähtevä ratkaisujen, rajapintojen ja tietojärjestelmäpalvelujen määrittely. Käytännössä on yhdisteltävä molempia lähestymistapoja, kun rakennetaan palvelupohjaisia tietojärjestelmäratkaisuja sekä koosteisia palveluja, jotka integroivat useita alemman tason vanhoja ja uusia osasia prosessien ja koostettujen sovellusten toteutuksiksi.

1.3 Yhteentoimivuus kokonaisarkkitehtuurissa

TOGAF-arkkitehtuurikehikossa (Open Group 2009) yhteentoimivuus määritellään:

1. Kyvyksi jakaa tietoa ja palveluja,
2. Kahden tai useamman järjestelmän tai komponentin kyvyksi vaihtaa ja käyttää tietoja,
3. Järjestelmien kyvyksi tarjota ja käyttää muiden järjestelmien palveluja siten että saadaan aikaan haluttuja vaikutuksia.

Näistä määritelmistä järjestyksessä toinen on laajasti käytetty IEEE:n määritelmä. TOGAF-arkkitehtuurikehikon menetelmäsuositusten yhteentoimivuuskuvausten määrittelyssä (Open Group 2009) suositellaan, että arkkitehtuurimenetelmän visiovaiheessa tunnistetaan, kuinka tiukkaa ja hienojakoista integraatiolähestymistapaa käytetään. Integroitivaatimusten tarkentamiseen toiminta-arkkitehtuurin yhteydessä suositellaan matriisia, jossa määritellään minkä tasoista yhteentoimivuutta eri toimijoiden välillä tarvitaan. Edelleen tietojärjestelmäarkkitehtuurin tarkennuksissa kokonaisarkkitehtuurissa voidaan kuvata eri tietojärjestelmien tai tietojärjestelmäkokonaisuuksien välillä tavoiteltava yhteentoimivuuden taso. Mallin heikkoutena voidaan kuitenkin pitää sitä, että samojen toimijoiden välillä on usein erityyppisiä tarpeita, eikä toiminta-arkkitehtuurissa pelkästään toimijoita tarkastelemalla vielä päästä riittävän tarkalle tasolle jotta voitaisiin päättää "oikeasta" yhteentoimivuuden tasosta.

Standardisalkku on yksi keskeisistä yhteentoimivuuteen liittyvistä kokonaisarkkitehtuurin kuvauksista. TOGAF-mallissa *Technology standards catalog* sijoittuu teknologia-arkkitehtuuriin. Sikäli kuin standardisalkku sisältää myös muita kuin teknisiä kuvauksia (esimerkiksi menetelmä-, mallinus- tai tietoturva-vaatimusstandardeja), se voidaan sijoittaa myös periaatteet ja ohjaus-tasolle. Myös monet muut kokonaisarkkitehtuurin kuvaukset liittyvät integraatioon. TOGAF version 9 metatallissa määritellyistä kuvauksista (Open Group 2009) yhteentoimivuuteen liittyy tai siihen suoraan syötteitä antaa ainakin 18 kuvaustyyppiä¹.

TOGAF:in tekninen viitemalli (TRM) sisältää yhteentoimivuuteen liittyviä palveluja, jotka pohjautuvat pääosin olioväyläpohjaiseen (ORB) ajatteluun ja aiempaan TAFIM (Technical Architecture Framework for Information Management) viitemalliin. Perusideana on kahden keskeisen "teknisen" rajapintatason tarjoaminen sovellusten tuottamiseen: viestintäinfrastruktuuri (communications infrastructure) tarjoaa tiedonvälityspalveluja ja sovellusalusta (application platform) sovelluksissa usein tarvittavia "peruspalveluja". Palveluja on tunnistettu 92 kappaletta 12 eri luokassa, mukaan lukien mm. tiedonhallintapalvelut, sijainti- ja hakemistopalvelut, käyttöjärjestelmäpalvelut, järjestelmänhallintapalvelut ja sovelluskehityksessä käytettävät palvelut. Liiketoiminnan ja infrastruktuurin sovellukset rakennetaan näiden pohjapalvelujen päälle. Malli on varsin komponentti-orientoitunut eikä huomioi täysin esimerkiksi nykyaikaisia SOA-periaatteita tai palvelualustojen käyttöä. Malli tekee suuren joukon oletuksia sovellusalustassa löytyvistä palveluista, ja yksittäisiä palveluja ei ole kuvattu nimen lisäksi yleensä lainkaan.

Suomessa vuonna 2011 voimaan tullut Laki julkisen hallinnon tietohallinnon ohjauksesta (634/2011) (nk. tietohallintolaki) pyrkii edistämään tietojärjestelmien yhteentoimivuutta julkishallinnossa. Lain tarkoitus on *"tehostaa julkisen hallinnon toimintaa sekä parantaa julkisia palveluja ja niiden saatavuutta säätämällä julkisen hallinnon tietohallinnon ohjauksesta ja tietojärjestelmien yhteentoimivuuden edistämisestä ja varmistamisesta."*

Laissa esitetty tietojärjestelmien yhteentoimivuuden määritelmä on rekursiivinen: yhteentoimivuudella tarkoitetaan *"teknistä ja tietosisällöllistä yhteentoimivuutta muiden julkisen hallinnon viranomaisten tietojärjestelmien kanssa silloin, kun järjestelmät käyttävät samoja tietoja."* Määritelmä ei ota kantaa siihen, tarkoitetaanko samojen tietojen käytöllä semanttisesti samojen tietokokonaisuuksien, samojen yksityiskohtaisten tietoelementtien tai esimerkiksi jaettujen tietovarantojen (yhdessä pisteessä fyysisesti sijaitsevan alkuperäisen tiedon) käyttöä.

¹ organization / actor catalog, service / function catalog, process / event / control / product catalog, business interaction matrix, functional decomposition diagram, process flow diagram, event diagram, data entity / data component catalog, data entity / business function matrix, system / data matrix, data dissemination diagram, requirements catalog, application portfolio catalog, interface catalog, system / function matrix, application interaction matrix, application communication diagram, technology standards catalog, platform decomposition diagram.

Laissa säädetään, että Valtiovarainministeriö huolehtii julkisen hallinnon tietohallinnon yhteisen kokonaisarkkitehtuurin edellyttämien yhteentoimivuuden kuvausten ja määritysten laatimisesta ja ylläpidosta. Kunkin ministeriön on huolehdittava, että sen toimialalle laaditaan ja ylläpidetään sen toimialan tietojärjestelmien yhteentoimivuuden kuvaukset ja määritykset, joista voidaan säätää asetuksella. Jos ministeriöiden toimialojen välillä on yhteentoimivuutta estäviä päällekkäisyyksiä, on ministeriöiden sovittava niiden poistamisesta.

Lakia voidaan täydentää asetuksilla mm. yhteentoimivuuden kuvausten sisältöön ja tietoarkkitehtuurin osalta, ja valtiovarainministeriö voi päättää, että julkisen hallinnon suosituksista (myös ja erityisesti yhteentoimivuuteen liittyen) voidaan tehdä julkisen hallinnon standardeja.

Julkisen hallinnon viranomaisten on suunniteltava ja kuvattava kokonaisarkkitehtuurinsa sekä noudatettava sitä ja sen *"edellyttämiä yhteentoimivuuden kuvauksia ja määrittämiä sekä toimialakohtaisia tietojärjestelmien yhteentoimivuuden kuvauksia ja määrittämiä."* Uusia tietojärjestelmiä hankittaessa, vanhoja olennaisesti muutettaessa tai palvelusopimusten päättyessä on tietojärjestelmät saatettava vastaamaan asetusten mukaisia yhteentoimivuusvaatimuksia. Asetuksella voidaan myös säätää määräaika yhteentoimivuuskuvausten mukaisuudelle tietojärjestelmissä. Viranomaisen on pyrittävä järjestämään toimintansa niin, että se käyttää väestötietojärjestelmän, yhdistysrekisterin, kaupparekisterin, säätiörekisterin, kiinteistötietojärjestelmän, yritys- ja yhteisötietojärjestelmän sekä maastotietoja koskevien tietojärjestelmien tietoja mikäli se tarvitsee niitä toiminnassaan.

Laissa ei ole tarkempia määritelmiä "yhteentoimivuuden kuvaus" -käsitteelle. Sen sijaan julkisen hallinnon kokonaisarkkitehtuurissa määritellään, että julkisen hallinnon yhteisen arkkitehtuurin tavoitetilan kuvausten tavoitteena on muodostaa oleellinen informaatio tietohallintolain tarkoittamien yhteentoimivuuden kuvausten ja määritysten toteutusta varten (VM 2011a). Yhteentoimivuuden kuvauksissa ja määrittämissä pyritään määrittelemään muun muassa tietojärjestelmien keskinäiset riippuvuudet, tietojärjestelmien vaihtamat tiedot ja tietojärjestelmien suhteet muihin kokonaisarkkitehtuurin elementteihin kuten tietovarantoihin ja toimintaprosesseihin. Yhteentoimivuuden kuvausten osat määritellään seuraavasti (VM 2011a):

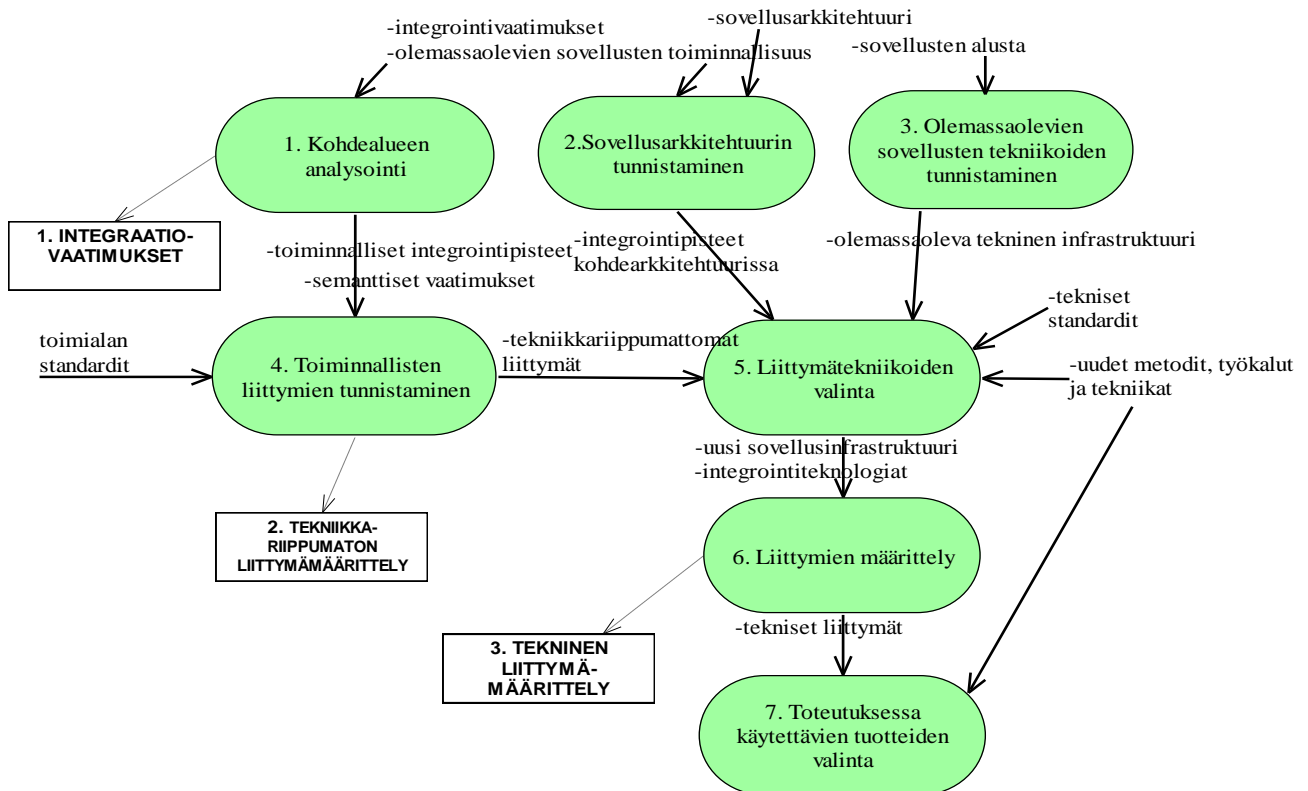
- käyttötarkoitus tai asiayhteys
- tietosisältö: siirrettävät tiedot ja niiden rakenne
- toiminnallisuus pääasiallisella tasolla tiedonsiirto- tai rajapintapalvelussa
- sanomarakenne tietojen siirrossa
- toteutusteknologia tiedonsiirto- tai rajapintapalvelulle.

Tavoitetilan viite- tai kohdearkkitehtuurikuvauksissa määritellään prosessien ja toimintojen välinen yhteentoimivuus ja palvelujen ja tietojärjestelmien välillä siirrettävät tiedot. Tarkemmissa kohdearkkitehtuurikuvauksissa määritellään tiedonsiirto- ja rajapintapalvelujen toteutusteknologiat ja tekniseen toiminnallisuuteen liittyvät kuvaukset. Yhteentoimivuuden todetaan edellyttävän käytännössä välitettävän tiedon standardointia termien ja käsitteiden välisistä suhteista kulloinkin käytettävässä kontekstissa. Eräänä julkisen hallinnon kokonaisarkkitehtuurin fyysisen tason kuvauksena on määritelty "Julkisen hallinnon tietojärjestelmien rajapinnat" -kuvaus, jossa kuvataan yhteisten palveluiden käyttämät integraatoratkaisut ja rajapinnat yleiskuvatasolla. Kuvaus sisältää keskeiset tiedot järjestelmien, palvelujen ja tietovarantojen välisten tiedonsiirtojen toteuttamista varten. Kustakin rajapinnasta kuvataan toimintaprosessi, jonka tarpeeseen rajapinta on tarkoitettu, rajapinnan toteuttamat käyttötapaukset, rajapinnan avulla välitettävät tiedot, virheen käsittely, standardi jolle rajapinta perustuu sekä rajapinnan tekninen kuvaus.

Julkisen hallinnon kokonaisarkkitehtuurisuosituksissa ei käsitellä tarkemmin esimerkiksi olemassa olevien standardien käyttöä, kansainvälistä yhteensopivuutta esimerkiksi tietyn toimialan vakiintu-

neiden standardien kautta tai mallipohjaisen lähestymistavan mahdollistamaa käsitteellistä ja sisällöllistä yhteentoimivuutta, joka voidaan toteuttaa eri tekniikoiden ja niiden välisten adapterien avulla. Pääasiallinen integraatiotapa (ks. luku 2.2) suosituksissa on tietopohjainen eikä prosessi- tai esim. käyttäjä- tai portaalilähtöistä integraatiota käsitellä rajapintojen yhteydessä. JHS-suosituksena julkaistuun Julkisen hallinnon standardisalkkuun (JHS 2011) on pyritty kokoamaan joukko yhteentoimivuutta edistäviä standardeja, joita voitaisiin hyödyntää koko julkishallinnossa. Standardisalkun standardien sitovuustasoja ei ole toistaiseksi määritelty tarkasti, eikä standardisalkkuun valituista standardeista ole yhdenmukaisesti kuvattu esimerkiksi kunkin standardin pääasiallista kohdealuetta (scope). Saatavilla on myös rakenteisia standardisalkkuja tarkemmille kohdealueille tai toimialoille (esim. Viinikainen ym. 2011).

Rajapintamäärittelyjen ja yhteentoimivuuden kuvausten tuottamisessa on mahdollista hyödyntää iteratiivista integraatoratkaisujen määrittelyprosessia ja -menetelmää, jossa määrittelyjen integraatiovaatimusten perusteella tuotetaan toiminnalliset ja tekniset kuvaukset sekä toteutuskohtaiset tarkennukset eri integraatiotilanteita varten. Alla esitetään yleiskuva integraatoratkaisujen ja yhteentoimivuuden kuvausten määrittelyn yhdestä iteraatiosta menetelmää käyttäen (Mykkänen ym. 2004, Mykkänen 2007).



Kuva 2. Integraatoratkaisun määrittelyprosessi.

2 Yhteentoimivuuden viitemalleja

2.1 European Interoperability Framework, versio 2

European Interoperability Framework (EIF) on joukko eurooppalaisia suosituksia hallintoviranomaisten, yritysten ja kansalaisten palvelujen yhteentoimivuuden ja viestinnän kehittämiseksi EU:n sisällä ja eri EU-maiden välillä. Se itse määrittelee yhteentoimivuuskehikon "sovituksi yhteentoimivuuden lähestymistavaksi organisaatioille, jotka toimivat yhdessä julkisten palvelujen tarjoamiseksi" ja korostaa monien erityyppisten yhteisten yhteentoimivuuden elementtien määrittelyä (EC 2010).

EIF-kehikon ensimmäinen² versio kehitettiin IDABC-ohjelman² osana ja julkistettiin vuonna 2004. Monet EU-maat ovat käyttäneet dokumenttia kansallisten yhteentoimivuuskehikoiden pohjana ja ohjeena projektien hallinnassa ja hankinnoissa. EIF-kehikon versio 2 (EC 2010) hyväksyttiin Euroopan komissiossa joulukuussa 2010. Version 2 ympärillä käytiin poliittista debattia teknisten ja kaupallisten seikkojen ja toimittajakohtaisten ohjelmistojen lobbauksen tiimoilta (mm. Friedrich 2009).

EIF-kehikon luvussa 2 esitellään 12 peruseriaatetta (underlying principles) Euroopan julkisten palvelujen kehittämiselle³ (EC 2010). Luvussa 3 esitetään käsitteellinen malli julkisille palveluille. Malli pohjautuu palvelukeskeiseen ajattelutapaan, jossa yhteentoimivuuden toteuttajat, perusrekisterit sekä ulkoiset palvelut ovat peruspalveluja, joista voidaan turvallisen tiedonvälityksen kautta muodostaa ja orkestroida koostettuja palveluja käyttäjille.

EIF:n luku 4 esittelee yhteentoimivuustasot (ks. kuva 3), joita "suositellaan huomioon otavaksi kun kehitetään uusia julkisia palveluja". Luvussa pyritään muodostamaan yhteistä sanastoa yhteentoimivuushaasteiden ja -ratkaisujen kuvaamiselle.

EIF version 2 yhteentoimivuustasot ovat lainsäädäntötaso, organisaatiotaso, semanttinen taso ja tekninen taso. Lainsäädäntötasolla pyritään nostamaan esiin eri maiden lainsäädännöstä johtuvien yhteentoimivuuden esteiden poistamista ja säädösten kehittämistä tukemaan tietojen vaihtoa eri maiden välillä huomioiden kunkin osallistujan säädökset esimerkiksi tietoturvallisuuden osalta. *Organisaatiotasolla* käsiteltäviä asioita ovat mm. prosessien yhteensovittaminen, organisaatioiden väliset yhteydet ja muutostenhallinta. Näiden seikkojen hallinta viittaa siihen, että käytössä tulisi olla esimerkiksi kokonaisarkkitehtuurikehikko, mutta suosituksessa ei suoraan kehoiteta tähän. *Semanttisen tason* yhteentoimivuuden kuvauksessa korostetaan kykyä hyödyntää ulkoisista lähteistä tulevia tietoja ja tiedon merkityksen tarkkaa säilyttämistä ja ymmärtämistä. Suosituksessa korostetaan sektorikohtaisten tietorakenteiden ja -elementtien määrittelyä, ja semanttinen yhteentoimivuus edellyttää myös syntaktista yhteentoimivuutta. Semanttisen yhteentoimivuuden edistämiseen suositellaan Euroopan laajuisten ja kansallisten kehittämisohjelmien käynnistämistä ja niihin osallistu-

² IDABC (Interoperable Delivery of Pan-European eGovernment services to Administrations, Business and Citizens) was an European Union programme with as objective to "identify, support and promote the development and establishment of pan-European eGovernment services and the underlying interoperable telematic networks supporting the Member States and the Community in the implementation, within their respective area of competence, of Community policies and activities, achieving substantial benefits for public administrations, business and citizens".

³ 1. hajauttaminen ja toimivallan rajoittaminen, 2. käyttäjakeskeisyys, 3. palvelujen saatavuus, 4. turvallisuus ja yksityisyys, 5. monikielisyys, 6. hallinnon yksinkertaistaminen, 7. läpinäkyvyys, 8. tiedon säilyttäminen, 9. avoimuus, 10. uudelleenkäytettävyys, 11. tekniikkariippumattomuus ja mukautuvuus, 12. tehokkuus ja haluttujen vaikutusten tuottaminen.

mista. *Teknisen yhteentoimivuuden* tasolle mallissa sijoitetaan rajapintamäärittelyt, liityntäpalvelut, tietointegraatiopalvelut, tiedonvaihdon ja esittämisen muodot jne. Teknisen yhteentoimivuuden toteuttamiseksi suositellaan joko tunnustettujen standardiorganisaatioiden standardeja tai teollisuuskonsortioiden tuottamia teknisiä määrittelyjä.

Tasot sijoitetaan "poliittisen kontekstin" sisään. Tämän "tason" asioiksi dokumentissa tunnistetaan poliittisella tasolla tehtävä priorisointi, poliittinen tuki ja poliittiset visiot sekä eri jäsenvaltioiden poliittinen, taloudellinen ja aikataulullinen sitoutuminen yhteentoimivuuden edistämiseen.



Kuva 3. EIF version 2 yhteentoimivuustasot (EC 2010).

EIF version 2 luvussa 5 käsitellään yhteentoimivuussopimuksia ja suositellaan, että yhteentoimivuudessa sovittavien asioiden tulisi pohjautua valmiisiin formalisoiuihin määrittelyihin tai työskennellä olemassa olevien yhteisöjen kanssa niiden tuottamiseksi. Lisäksi suositellaan rakenteisen, läpinäkyvän ja objektiivisen lähestymistavan käyttöä hyödynnettävien määritysten valinnan tueksi. Määritysten avoimuuteen liittyen esitetään kolme näkökulmaa: avoin osallistuminen, vapaa saatavuus ja jakaminen sekä toteutusmahdollisuudet sekä avoimen että suljetun koodin kehitystavoilla. Viranomaisten osallistumista standardointiprosesseihin kannustetaan. Luvussa 6 suositellaan yli eri hallintotasojen käytettävän yhteentoimivuuden hallintamallin kehittämistä.

EIF versio 2 -mallia voidaan pitää osittaisena kokonaisarkkitehtuurikehikkona yhteentoimivuuteen liittyen. Se tarjoaa yleisen yhteentoimivuuden viitekehyksen etenkin poliittisen, eri maiden välisen ja organisaatioiden välisen yhteentoimivuuden osalta. Toisaalta kehikossa on menetelmällisiä suosituksia esimerkiksi järjestelmällisten arviointi- ja valintamallin (ks. luku 3.4) käyttämiseksi. Kehikon suositusten epämääräisyyteen on kuitenkin kohdistunut arvostelua. Lisäksi on kiistelty esimerkiksi standardien käyttöön liittyvästä terminologiasta (termi "open specification" korvannut pääosin termin "open standard"). On todennäköistä, että standardien arvioinnissa korostuva määrittelyn riittävän laaja käyttö markkinoilla ja virallisessa de jure -standardoinnissa ilmenneet puutteet ratkaisujen hyödynnettävyydessä ovat vaikuttaneet esitettyihin suosituksiin.

2.2 Integrointitapojen luokittelu

Integrointiratkaisujen määrittelyissä on järkevää jo vaatimusten kartoituksen alkuvaiheessa tunnistaa vaatimusten ja perusratkaisujen ensisijainen luonne. Tähän voidaan käyttää esim. seuraavaa luokittelua (pohjautuen Linthicum 2003):

- *Tietopohjainen integraatio*, joka nojautuu tiedonsiirtoon tai kopiointiin osapuolten välillä. Yleisesti käytettyjä tietopohjaisia lähestymistapoja ova tietokantaliitännät, viesti- tai sanomaliikenne tai yhtenäiset asiakirjojen (business documents) määrittelyt. Integrointitapa on deklaratiiivinen eli perustuu siihen että sovitaan osapuolten välillä siirrettävistä tiedoista ja niiden merkityksestä. Tietopohjaisia integraatioita on käytetty runsaasti eri standardeissa ja integrointiratkaisuissa. Tietopohjaiset ratkaisut voivat nojautua yksinkertaisiin tietomäärittelyihin tai hyvinkin monimutkaisiin semanttisiin malleihin ja ontologioihin. Tietopohjaista integraatiota voidaan tukea erityyppisten integrointialustojen tai palveluväylän avulla.
- *Palvelupohjainen integraatio* perustuu jaettuihin ja yhtenäisesti määriteltyihin toimintoihin tai operaatioihin. Yhteisten tietojärjestelmäpalvelujen (common services) käyttö on tyypillinen tapa palvelupohjaisen integraation määrittelyyn. Historiallisesti palvelupohjaisessa integraatiossa on nojaututtu esimerkiksi etäohjelmakutsuihin ja niitä tukevien väliohjelmistojen käyttöön. Web services- ja WSDL-tekniikoiden tarjoamien operaatioiden käyttö sovel-luspalvelujen rajapintojen kuvaamiseen on tyypillinen palvelupohjaisen integraation muoto. Integrointitapa on imperatiivinen eli perustuu siihen, että palvelun tarjoaja suorittaa tehtäviä palvelun tarvitsijalle. Palvelupohjaisessa integraatiossa korostuvat uudelleenkäyttö sekä päällekkäisyyksien vähentäminen tietosisällöissä, toiminnallisuuksissa, toteutustyössä ja ylläpidossa. Palvelupohjaista ja tietopohjaista integraatiota on yhdistetty yksinkertaisten CRUD (create, read, update, delete) -operaatioita sisältävien rajapintojen tai REST-tyyppisten ohjelmistopalvelujen kautta.
- *Käyttäjälähtöinen integraatio* perustuu siihen, että muodostetaan yhdenmukainen näkymä moniin järjestelmiin tai vähennetään erillisten järjestelmien käytön haasteita käyttäjän näkökulmasta. Käyttäjälähtöisessä integraatiossa käytettyjä toteutustapoja ovat esimerkiksi eri tietolähteitä tai taustajärjestelmiä integroivat organisaatio- tai asiakasportaalit, yhtä aikaa käytettävien sovellusten synkronointi esimerkiksi samaan asiakkaaseen ja kertakirjautumisratkaisut. Ratkaisuihin korostuvat usein käytettävyys-, personointi- tai monikanavaisuusvaatimukset.
- *Prosessipohjainen integraatio* perustuu joko määriteltyjen ja keskitetysti ylläpidettyjen prosessien kerrokseen (orkestraatio) tai prosessin eri osapuolten noudattamiin yhdenmukaisiin toimintaohjeisiin noudattamiseen (koreografia). Orkestraatio-mallissa arkkitehtuurissa on prosessin koordinaattori joka ohjaa prosessin vaiheiden etenemistä ja kutsuu eri vaiheita toteuttavia osapuolia tai palveluita. , prosessien hajauttaminen. työnkulkujen ymmärtämisestä määrittelyyn ja ohjaukseen.

Kaikkiin integrointitapoihin on olemassa joukko erilaisia käytäntöjä ja standardeja. Niitä voidaan myös käyttää organisaatio-, kohdealue- tai projektikohtaisiin tarpeisiin mahdollisten integrointivaihtoehtojen hahmottamisessa ja ratkaisujen määrittelyssä.

2.3 Seitsentasoinen yhteentoimivuuden viitemalli

Ohjelmistoja liitettäessä järjestelmiä on sovitettava yhteen monilla eri tasoilla. Eri tasojen tunnistaminen on hyödyllistä yhteensovittamista suunniteltaessa ja tekniikoita ja standardeja valitessa. Ohjelmistojen liitettävyyden viitemallina käyttökelpoinen on seitsentasoinen yhteentoimivuusmalli (Herzum, Sims 2000) (ks. kuva 4). Yhteentoimivuus toteutuu viime kädessä aina kahden toisiinsa yhteydessä olevan järjestelmän välillä. Mallin kaikilla tasoilla jokainen protokolla, jota ei käytetä molemmissa osallistuvissa järjestelmissä, vaatii tulkkauksen (mapping). Jollei tällaista tulkkauksia ole saatavilla, ei yhteensovittaminen ole mahdollisia ilman järjestelmiin tehtäviä muutoksia.

Lait, ohjeet, toimintatavat
Prosessit
Kehitysprosessin liittymät
Toiminnallinen viitemalli
Semantiikka
Toiminnalliset liittymät
Sovellusinfrastruktuuri
Tekninen infrastruktuuri
Tekniset liittymät
Verkot
Laitteet

Kuva 4. Seitsentasoinen yhteentoimivuuden viitemalli täydennettynä laitteisto- ja prosessi / kontekstitasoilla.

Varsinaisen seitsentasoisen mallin tasot alimmasta korkeimpaan ovat (muk. Herzum, Sims 2000):

1. *Tekniset liittymät*: ohjelmistomekanismeja, joilla kohdejärjestelmään ollaan yhteydessä. Tämän tason sopiminen tarkoittaa käytettävien tiedonsiirto-, ja rajapintakuvaustekniikoiden sopimista, esim. http:n, CORBA/IIOP-tiedonsiirtoprotokollan, RPC:n, DCE:n tai suoran SQL-kielen käyttöä jaettuun tietoon pääsemiseksi. Teknisen liittymän protokolla voi olla myös usean tekniikan yhdistelmä, esim. XML skeeman tai REST-mekanismien käyttö http-protokollan kanssa. Tekninen liittymä voi olla myös tietokantaperusteinen, tiedostopohjainen yhdyskäytävä tai ohjelmointirajapinta (API) tai näiden yhdistelmä.
2. *Tekninen infrastruktuuri*. Mahdollisuus kutsua teknisesti toisen järjestelmän liittymää ei poista kahden järjestelmän välisen vuorovaikutuksen teknisiä haasteita. Esimerkiksi kohdejärjestelmän on joko oltava käynnissä jotta sitä voi kutsua, tai kohdeympäristön on tiedettävä, kuinka käynnistää kohdejärjestelmä kutsun tullessa. Esimerkiksi tämäntyyppinen aktivointi kuuluu teknisen infrastruktuurin vastuulle. Tekninen infrastruktuuri kattaa kahden järjestelmän välisen kommunikoinnin teknisen tuen, ja sisältää mm. virheidenkäsittelyn, nimeämisen, transaktiot jne. Se sisältää myös työnkulun tekniset näkökohdat tai mahdolliset skriptaukset, joilla voidaan helpottaa ja koordinoita vuorovaikutusta. Teknisen infrastruktuurin ja teknisen liittymän erottaminen on olennaista, koska infrastruktuurin on toisaalta tuettava valittuja integrointitekniikoita, toisaalta toimittava nimenomaisissa suoritusympäristöissä. Teknistä infrastruktuuria on standardoitu vähemmän kuin rajapintoja tai teknisiä liittymiä.
3. *Sovellusinfrastruktuuri*. Toiminnallisen yhteensopivuuden saavuttamiseksi tarvitaan sovellusinfrastruktuuria, joka sisältää arkkitehtuuriset päätökset, liittymäkäytännöt ja suunnittelumallit. Kukin teknisen infrastruktuurin protokolla on sovitettava sovellusinfrastruktuuriin, eli kuvattava mitkä osat osallistujien sovellusarkkitehtuurissa toteuttavat yhteentoimivuusratkaisun. Tällä alueella ei ole juuri standardeja, pois lukien jäsentävät viitearkkitehtuurit sovellusarkkitehtuurin osalta (ks. Itälä ym. 2012, luku "5.2 S3: A Service-Oriented Reference Architecture"). Järjestelmien integraatiopisteiden sopiminen niiden sovellusarkkitehtuurin kerroksissa on olennainen osa sovellusinfrastruktuurin integrointipäätöksiä. Integraation keskeiset arkkitehtuuripäätökset

kuten miten ratkaisuun käytetään tietoverkkoa tai tapahtuuko yhdistäminen samassa osoiteavaruudessa, ovat keskeinen osa tätä sopimista. Sovellusinfrastruktuuriin kuuluvat myös esimerkiksi virheidenkäsittelyprotokollat sekä sopiminen tavasta, jolla tunnistetaan osallistuvat järjestelmät, tai toimintasäännöistä, joilla toisille järjestelmille sallitaan operaatioita ja toisilta ne kielletään.

4. *Toiminnalliset liittymät.* Mikäli integraation toteutuksessa on päätetty, että liittymien määrittelyyn käytetään jotain tiettyä tekniikkaa, tiedetään, millainen tekninen infrastruktuuri tarvitaan ja miten liittymää kutsutaan teknisesti. Tällöin ei kuitenkaan tiedetä mitään toiminnallisesta liittymästä. Esimerkiksi se, onko operaatio *hae_potilaan_tiedot* vai *lisaa_tutkimus*, ei ole osa teknisiä päätöksiä. Toiminnallisessa liittymässä määritellään rajapintojen ja niiden operaatioiden toiminnalliset nimet ja parametrit tai tietoelementit. Teknisten ja toiminnallisten liittymien muodot ovat suorassa yhteydessä toisiinsa: toiminnallisten liittymien määrittelyssä käytetään teknisten liittymien ja teknisen infrastruktuurin tarjoamia käytäntöjä. Nykyisillä tekniikoilla on myös mahdollista määrittellä toiminnalliset liittymät ilman sitoutumista tiettyihin teknisiin liittymiin. Mallipohjainen MDA (Model Driven Architecture) perustuu siihen, että ratkaisu voidaan määrittellä toteutustekniikka- ja alustariippumattomasti (platform). Toiminnallinen liittymä voi perustua imperatiiviseen "prosessointikäskyyn" tai deklaratiiiviseen "dokumentti/tietopaketti" ajatteluun. Ohjelmointirajapintoja käyttävissä liittymissä nämä päätökset näkyvät operaatioiden nimien ja parametrien määrittelyissä sekä operaatioiden välisissä yhteyksissä. Tietopaketti- tai dokumenttipohjaisissa ratkaisuisissa dokumentti- tai sanomatyyppit sekä elementtien nimet ovat keskeisiä.
5. *Semantiikka.* Kun toiminnallisista liittymistä on sovittu, on tarpeen sopia tietojen ja interaktioiden tarkasta merkityksestä. Pelkästään liittymiä tarkastelemalla ei rajapinnan merkityksestä saa kuvaa. Toiminnan kannalta liittymän operaatioiden ja tietoelementtien merkitykset ovat kuitenkin äärimmäisen tärkeitä. Sama liittymä voi periaatteessa johtaa eri tapauksissa hyvin erilaiseen lopputilaan järjestelmässä, ja kutsujalle palautuvan tiedon merkitys voi poiketa paljonkin eri tapauksissa. Näin ollen alemmilla tasoilla tarkasti sovitulla liittymällä voi olla hyvin erilaisia merkityksiä: esimerkiksi kuvaako operaation paluarvo "1" kutsutun operaation onnistumista, muutettujen tietokantarivien lukumäärää. Semantiikan määrittelyissä voidaan nojautua kehittäjien tulkitsemaan luonnolliseen kieleen, ylemmän tason viitemallien hyödyntämiseen tai yhteisesti sovittujen koodistojen, luokitusten ja terminologioiden käyttöön. Kun tietyn järjestelmän semantiikan määrittely on riittävän korkealla tasolla, voidaan toteuttaa semantiikan toteuttavia yhdyskäytäviä, semanttisia siltoja tai semanttisesti "älykkäitä" kyselyjä. Laajamittainen semanttinen yhteensopivuus tällä tasolla vaatii kuitenkin kattavaa sopimista ja standardointia. Yhteiset viitetietomallit, sanastot, luokitukset ja koodistot sekä eritasoisia ontologioita hyödyntävien välittäjien ja rekisterien käyttö ovat semanttisen yhteentoimivuuden automatisointiin tärkeitä menettelytapoja.
6. *Toiminnallinen viitemalli.* Järjestelmien toiminnallinen viitemalli sisältää oletukset ja käytännöt, joita järjestelmän toteuttamisessa on käytetty. Viitemallin protokollat voivat olla järjestelmien sisäisiä tai välisiä. Esimerkkinä sisäisen viitemallin vaikutuksista integraatioon voidaan pitää esimerkiksi henkilön tunnisteiden pituutta tietokannassa: jos kutsuvassa järjestelmässä henkilön tunniste on 10 merkkiä pitkä ja kutsun vastaanottavassa järjestelmästä tunniste on 20-merkkinen, ei kutsuva järjestelmä kykene käyttämään saamaansa tietoa oikein. Toiminnalliset viitemallit voivat keskittyä useisiin alempiin tasoihin, kuten semantiikka, toiminnalliset liittymät tai yllä kuvattu sovellusinfrastruktuuri. Esimerkki tietomallin (toiminnalliset liittymät) ja semantiikan tasolla toimivasta standardoidusta toiminnallisesta viitemallista on esimerkiksi tietosisältöjä ja merkityksiä standardoivan HL7 RIM -viitetietomallin (ks. Itälä ym. 2012, luku "5.3 RIM HL7 V3 reference information model") käyttö järjestelmien suunnittelun pohjana.

7. *Kehitysprosessin liittymät.* On tilanteita, joissa järjestelmässä tiedetään jo aikaisessa kehitysvaiheessa, millaista yhteentoimivuutta tarvitaan. Tällöin esimerkiksi mukana olevien järjestelmien rajoitukset ja mahdollisuudet tai hyödynnettävät yhteiset standardit voidaan ottaa huomioon jo varhain kehitysprosessissa. Tyypillisissä erikseen kehitettyjen ohjelmistojen integraatiotilanteissa ja vanhoissa perinnejärjestelmissä voi kuitenkin olla hyvin vaikeaa toteuttaa perustavanlaatuisia muutoksia, jolloin ei välttämättä saavuteta halutun tason yhteentoimivuutta.

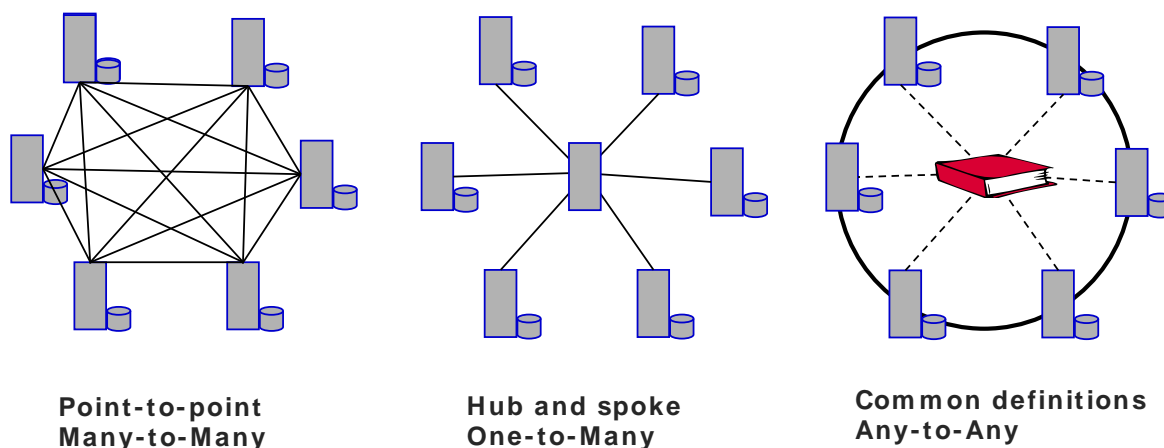
Yllä kuvatuista tasoista tasot 1-6 on ratkaistava aina, kun järjestelmiä integroidaan. Tämän lisäksi yhteentoimivuutta tukevat säädökset ja esimerkiksi yhteiset prosessit voivat ohjata huomattavasti integraatiota ja eri tasoilla tehtäviä ratkaisuja. Lisäksi käytännön integraatoratkaisut on toteutettava teknisesti erilaisia laite- ja verkkoratkaisuja hyödyntäen. Monissa standardeissa ja esimerkiksi yhteentoimivuuskuvausten määrittelyissä (ks. luku 1.3) keskitytään vain joihinkin yllä kuvatuista tasoista. Standardeissa ja määrittelyissä on pahimmillaan piilotettuja oletuksia ja vaikutuksia myös muille yhteentoimivuuden tasoille kuin niihin joihin määrittely pääosin kohdistuu. Integrointiprojekteissa ja palvelupohjaisessa järjestelmien kehittämisessä voidaan törmätä ikäviin yllätyksiin, ellei kaikkia tarvittavia tasoja oteta huomioon.

Järjestelmäkehityksessä teknisen arkkitehtuurin on otettava huomioon teknisten liittymien ja teknisen infrastruktuurin yhteentoimivuuskäytännöt. Sovellusarkkitehtuurissa on otettava huomioon sovellusinfrastruktuuri ja mahdollisesti teknisen ja toiminnallisen arkkitehtuurin sekä toiminnallisen viitemallin vaikutukset arkkitehtuuriin. Toiminnallisen arkkitehtuurin määrittelyssä on otettava huomioon toiminnalliset liittymät, semantiikka ja toiminnallinen viitemalli. Järjestelmän kehitysprosessissa ja arkkitehtuurilinjauksissa tulee huomioida eri yhteentoimivuustasojen vakiointien ja standardien lisäksi erityisesti myös kehitysvälineet ja kehitysprosessin liittymät.

3 Integraatioarkkitehtuurit ja SOA

3.1 Integraation topologiat

Integroitavien tietojärjestelmien lukumäärän kasvaessa tulee ratkaistavaksi kysymys integraation topologiasta eli verkon arkkitehtuurista, joka on hajautetuissa järjestelmissä erityisen keskeinen sovellusinfrastruktuurin (ks. luku 2.3) arkkitehtuurilinjaus. Oheinen kuva esittää topologian erilaisia vaihtoehtoja (muk. Linthicum 2003). Kaikkia topologioita on mahdollista käyttää "SOA-tyyppisesti".



Kuva 5: Integraation topologiat

Organisaatoiden sisäisten ja niiden välisten tietojärjestelmien integraatio lähti aikoinaan liikkeelle käyttämällä kiinteitä yhteyksiä. Tarpeen vaatiessa yhdistettiin aina kaksi osapuolta keskenään jolloin päädyttiin kahdenväliseen (point-to-point) topologiaan. Tämän topologian haittapuolena on tarvittavien yhteyksien lukumäärän voimakas kasvu integroitavien osapuolten määrän lisääntyessä. Kaikkien mahdollisten yhteyksien lukumäärä voidaan ilmaista kaavalla $n*(n-1)/2$, jossa n on integroitavien järjestelmien lukumäärä. Yhteyksien määrä kasvaa siis toisessa potenssissa.

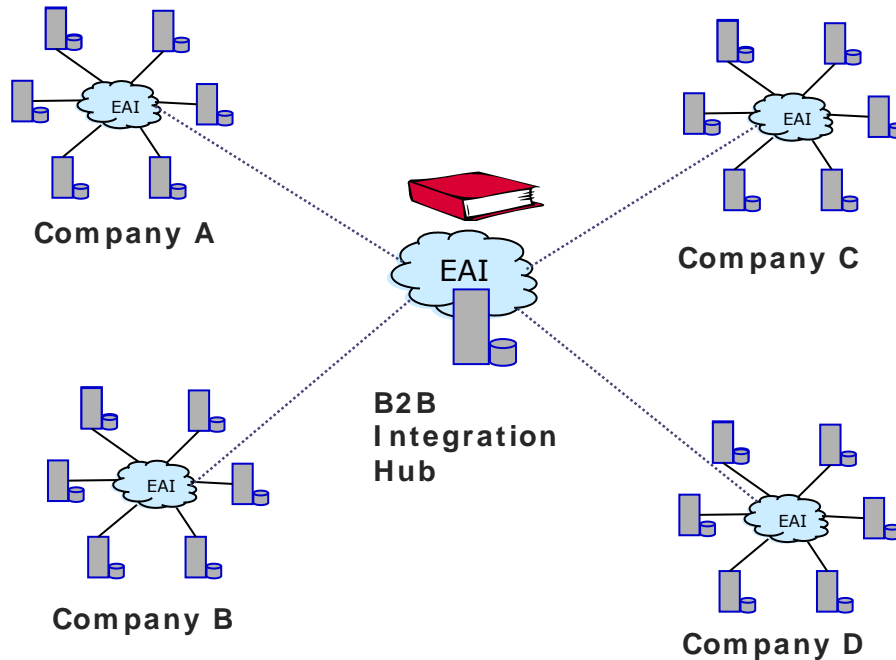
Point-to-point -tyyppisen topologian vaihtoehdoksi alkoi nousta keskitettyyn integraatioalustaan perustuva topologia. Siinä on keskellä solmu - hub - joka toimii tietoliikenteen välityksen järjestelmänä ja johon kaikki muut järjestelmät liittyvät. Siitä juontaa englanninkielinen nimitys hub-and-spoke -topologia, jota myös tähtimalliksi kutsutaan. Tässä topologiassa tarvittavien yhteyksien lukumäärä on n eli sama kuin integroitavien järjestelmien lukumäärä. Kiinteitä yhteyksiä käytettäessä hankittiin yhteydet kustakin integroitavasta järjestelmästä keskitettyyn yhteyskeskukseen.

Internetin yleistyessä ja ulottuessa kaikkialle on yleistynyt myös dynamisempi topologia, jossa yhteydet avataan vasta tarpeen vaatiessa yhteyttä tarvitsevien osapuolten välille. Tämä perustuu yhteisten standardien käyttämiseen, esimerkiksi tcp/ip tietoliikenteessä ja http/html sovellusten välisessä tiedonsiirrossa ja tiedon esityksessä. Yhteisesti sovittujen standardien käyttö antaa mahdollisuuden kahden minkä tahansa eri osapuolen väliselle yhteydenpidolle, siitä nimitys any-to-any.

Joissakin tilanteissa on tarvetta tietoturvalle ja suojatulle yhteydelle. Siinä osapuolet tunnistavat ja todentavat toisensa luotettavasti ja yhteys on suojattu ja salattu ulkopuolisten hyökkäyksiltä. Yksi ratkaisumalli näihin tilanteisiin on VPN (Virtual Private Network), jossa avoimen any-to-any ver-

kon päälle muodostetaan kahden osapuolen välinen suojattu point-to-point yhteys vain tarvittavaksi ajaksi.

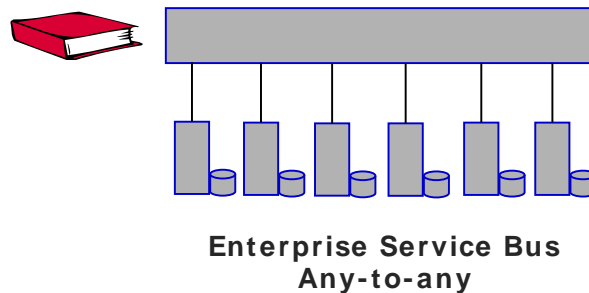
Yritysten ja organisaatioiden välisessä integraatiossa on osoittautunut tarkoituksenmukaiseksi jäsentää topologia tähtimallin mukaisesti sekä yrityksen sisällä että yritysten välillä oheisen kuvan mukaisesti:



Kuva 6: Organisaatioiden sisäinen ja organisaatioiden välinen integraatio

Tämä topologia helpottaa erityisesti organisaatioiden välisen integraatioalustan käyttöönottoa ja hallintaa, kun kukin yritys näkyy vain yhtenä osapuolena, jonka takana ovat kaikki organisaation moninaiset tietojärjestelmät.

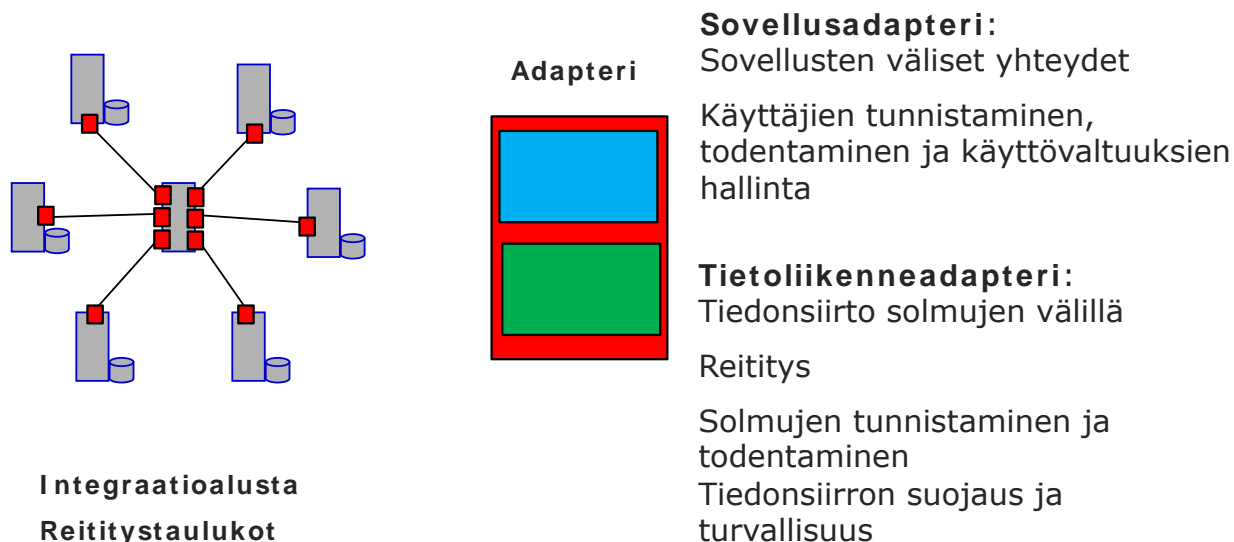
Topologiana on yleistynyt myös oheisen kuvan mukainen palveluväylä, Enterprise Service Bus. Ajatuksena on liittää kaikki osapuolet yhteiseen integraatioalustaan, "väylään", jonka välityksellä eri osapuolet voivat olla toisiinsa yhteydessä. Palveluväylän vaikutuksia käsitellään tarkemmin luvussa 3.3.



Kuva 7: Palveluväylä

3.2 Adapterit

Käytännössä erilaisten, eri-ikäisten ja eri teknologiaa edustavien tietojärjestelmien integraatiossa on käyttökelpoiseksi ratkaisuksi osoittautunut adaptereiden käyttö. Adapterin rooli on toimia välittäjänä. Käytännöksi on muodostunut jakaa adapterin toiminnallisuus kahteen kerrokseen. Alemmalla kerroksella sovitetaan yhteen tiedonsiirtotason protokollat ja sitä hyödyntävälle kerroksella sovitetaan yhteen sovellustason protokollat. Kummallakin tasolla sovitaan tarpeelliset kuittaukset ja virheenselvitys- ja toipumiskäytännöt. Adaptereista voidaan myös pyrkiä tekemään yleiskäyttöisiä siten, että ne ovat helposti liitettävissä erilaisiin lähdejärjestelmiin.



Kuva 8: Integraation kerrostaminen

Yllä olevassa kuvassa sovellusadapteri luvun 2.3 mallin mukaisesti hoitaisi sovellusinfrastruktuurin, toiminnalliset liittymät, sekä tarvittaessa esimerkiksi semanttisen sovittamisen. Tietoliikenneadapteri hoitaa teknisen infrastruktuurin sekä tekniset liittymät (tietoliikenne- ja rajapintatekniikat) ja kommunikoi verkkoa käyttäen. Sovellus- ja tiedonsiirtotasolla tarvittavat kuittaukset kuuluvat kooluvat sovellus- tai teknisen infrastruktuurin vastuulle.

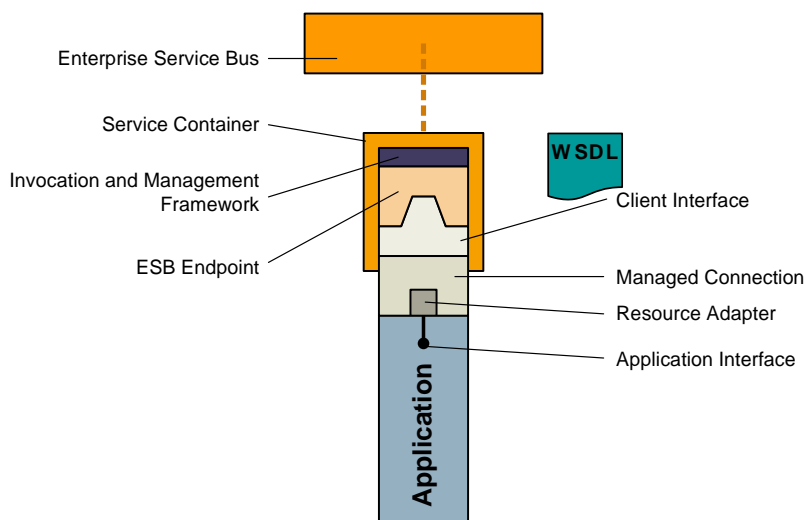
Käytännön integraatioprojekteissa on sovittava, tehdäänkö yhteisiin standardeihin sovittamien osallistuvassa järjestelmässä vai integraatioalustan avulla. Integraatioissa on usein helpompaa tehdä tekninen sovitus integraatioalustan kautta. Näin on laita mm. tietovarastointihankkeissa, joissa ETL (Extract-Transform-Load) -työvälineiden avulla erilaisten lähdejärjestelmien tiedonsiirtomahdollisuudet ja sisällölliset eroavuudet ovat yhdenmukaistettavissa, jotta tieto saadaan ladattua yhteiskäyttöiseen tietovarastoon, mikäli semanttiset vastaavuudet pystytään määrittelemään.

Integroitavien järjestelmien lukumäärän kasvaessa ja varsinkin jos osapuolia on useista eri organisaatioista on tarkoituksenmukaisempaa vaatia lähdejärjestelmiä yhdenmukaistamaan tiedonsiirtokäytäntönsä ja tietosisältönsä yhteisesti sovittujen standardien mukaisiksi. Tarvittaessa osallistuviin järjestelmiin voidaan rakentaa adaptereita tai yhteisiä liittymispalveluja. Näin on lähdetty toteuttamaan valtakunnallisia integraatioita kuten terveydenhuollon KanTa-järjestelmää. Semanttisella tasolla perusteita luovat asiakirjojen yhdenmukaistetut sisältö- ja rakennemäärittelyt sekä yhteisten koodistojen käyttö, jota varten on perustettu kansallinen koodistopalvelu.

3.3 Palveluväylä ja sen vaikutukset

Enterprise Service Bus (ESB) eli palveluväylä on yleistynyt yhdeksi integraatiota tukevaksi toteutus- ja suunnittelumalliksi 2000-luvun alkupuolella. Palveluväyliä on kehitetty myös organisaatioiden sovellusintegraatioissa (EAI) käytetyistä integrointialustoista tukemaan palvelukeskeistä kehittämismallia. Palveluväylälle tunnusomaista ovat sanomavälitykseen perustuva integraatioalusta, Web Services-tekniikoiden tukeminen, sisällön perusteella tapahtuva reititys, ja XML:n käyttö sisällön esittämisessä ja sen avulla tapahtuvissa muunnoksissa. Palveluväylät tukevat sekä synkronisia että ei-synkronisia viestinvälitysmalleja ja yleensä myös tapahtumapohjaista publish / subscribe-integrointimallia (ks. luku 4.2). Palveluväylät tukevat myös perinteisten integrointialustojen toiminnallisuuksia kuten sanomien reititystä ja muunnoksia eri sanoma- tieto- tai dokumenttirakenteiden välillä sekä kokonaisten sanomien säilyttämisen ja suodattamisen ratkaisuja. Sanomien sisältöjä voidaan jakaa, yhdistää, rikastaa eri lähteistä ja suodattaa. Palveluväylässä on usein integraatioon liittyvien metatietojen kuten osoitteistojen, rajapinta- ja skeemakuvausten sekä policy-määrittelyjen hallintaan ja tuottamiseen tarkoitettuja välineitä ja tietovarastoja. Samoin palveluväylät voivat tukea suoritusympäristössä tapahtuvaa tapahtumien seuranta (esimerkiksi BAM, Business Activity Monitoring), teknisen toimivuuden monitorointia sekä teknisen palvelutason (SLA) seuranta ja integrointiratkaisuissa tarvittavien turvallisuusominaisuuksien toteuttamista. Palveluväylät sisältävät usein myös prosessi- tai työnkulkumoottorin, jota voidaan käyttää prosessipohjaisen integraation orkestraatioissa. Palveluväylän eri integraatiokomponenttien hallitsemiseksi väylässä voidaan usein välittää myös kontrollitietoja. Palveluväylä-nimitystä on käytetty myös keskitetyistä integrointialustoista, mutta joidenkin näkemysten mukaan aidossa palveluväylässä myös integrointikomponentit on hajautettu palveluiksi, joita on oltava mahdollista käyttää ja yhdistellä toisistaan riippumatta SOA-periaatteiden mukaisesti.

Oheinen piirrosnotaatio⁴ kuvaa palveluväylän keskeisiä osia ja palveluväylään liittymistä (Chappell 2004).



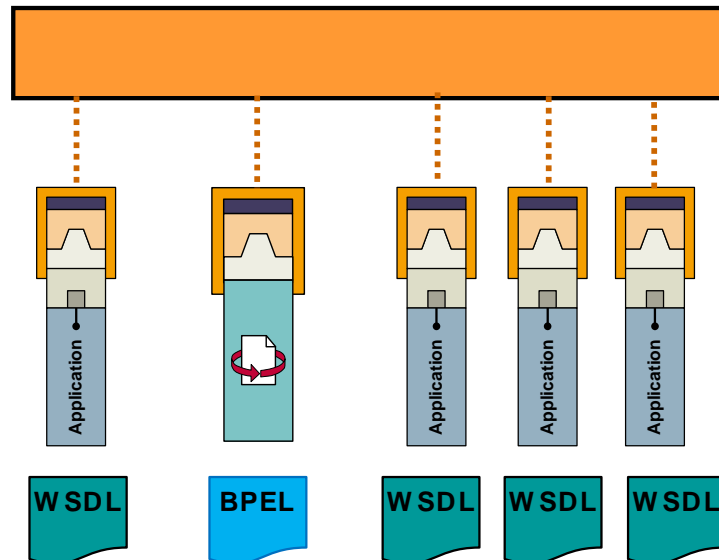
Kuva 9: Palveluväylä ja siihen liittyminen

Web services-pohjainen palveluväylä näkee siihen liittyvät osapuolet Web Service -palveluina, jotka kuvataan WSDL-dokumentilla. Palveluväylä voi tarjota myös tietoliikenne- ja sovellusadapte-

⁴ Myös notaation mukaiset kuvakkeet Visio- ja OmniGraffle-välineisiin ovat ladattavissa verkosta.

reiden toiminnallisuuden, joiden avulla erilaiset sovellukset on mahdollista saada näkyviin Web Service -palveluina.

Palveluväylä tarjoaa myös erilaisia palveluita. Oheisessa kuvassa on prosessia suorittava BPEL-moottori omana palvelunaan, mutta jotkin integrointialustat tarjoavat itsekkin prosessin suoritukseen tarvittavaa palveluiden orkestrointia.



Kuva 10: Esimerkki palveluiden orkestroinnista

Palveluväylä voi muodostua myös tiedonsiirtotasolla keskitetystä loogisesta integraatioalustasta, joka puolestaan tarjoaa kullekin järjestelmälle sopivan sovellusadapterin yllä olevan kuvan mukaisesti.

Palveluväylän avulla pyritään helpottamaan sovellusten tai palvelujen välisen integraation toteuttamista mm. seuraavin tavoin:

- Palveluväylä voi tukea teknistä yhteensovittamista ja muunnoksia eri formaattien ja protokollien välillä. Palveluväylä tarjoaa välineitä sovittaa rajapintojen syntakseja ja rajapintatekniikoita sekä eri kommunikaatioprotokollia. Näin voidaan vähentää sovellusten teknisiin adaptereihin kohdistuvia vaatimuksia. Myös eri viesti- ja rajapintaprotokollien välisten vastaavuuksien määrittely tai niiden välisten muunnosten automatisointi on usein mahdollista.
- Palveluväylällä pyritään parantamaan integraatioympäristön hallittavuutta mm. tarjoamalla ainakin loogisella tasolla keskitetty yhteys- ja valvontapiste. Keskitetty hallinta voidaan toteuttaa, vaikka integrointipalvelut olisikin hajautettu palveluväylässä. Hajautuksen avulla voidaan myös pyrkiä siihen, että yhdessä pisteessä tapahtuva virhe ei halvauta koko integraatioinfrastruktuuria.

Palveluväylän avulla pyritään siis lisäämään joustavuutta ja erilaisia integraation toteutusmahdollisuuksia. Palveluväylä kuitenkin lisää myös uuden kerroksen järjestelmään. Erilaisten integraation toteuttamistapojen vakiointi myös palveluväylää käytettäessä on järkevää. On otettava kantaa myös siihen, mitkä seikat hoidetaan palveluväylän avulla ja missä määrin etenkin sovellusinfrastruktuurin (ks. luku 2.3) vaatimuksia on toteutettava liitettäviin sovelluksiin tai tietojärjestelmäpalveluihin. Tällaisia seikkoja ovat mm. seuraavat:

- Mitkä seikat otetaan mukaan toiminnallisiin rajapintamäärittelyihin ja mitkä hoidetaan palveluväylän tarjoamien välineiden avulla. Palveluväylä voi tukea esimerkiksi yleiskäyttöisiä rajapintamäärittelyjä erityyppisten resurssien etsimiseen ja hakemiseen, jolloin voidaan käyttää hyvin yleiskäyttöistä rajapintaa, joka vasta kutsujasta, kutsun tietystä parametrilla tai paikallisesta toimintapolitiikka (policy) määräyksestä riippuen saa lopullisen toiminnallisen merkityksensä ja esitysmuotonsa palveluväylässä (esimerkiksi "hae potilaskertomusasiakirja" tai "hae päätöksentuen palvelun tietosisältövaatimusten kuvaus").
- "Tiedon tai pyynnön vastaanottaja" voidaan yhdessä tilanteessa määritellä osana tarjottavaa rajapintaa "kenelle viesti lähetetään", selvittää toisessa tilanteessa rekisteristä kutsujan toimesta "mistä tämä palvelu tai tieto löytyy", kolmannessa tilanteessa hyödyntää väylän reititysominaisuuksia "mitkä osallistajat tarvitsevat tämän tiedon".
- On päätettävä pyritäänkö kaikki palvelukutsut tai sanomaliikenne ohjaamaan palvelualueen kautta. Eri projektien ja organisaatioiden arviot siitä, kuinka laajoja kokonaisuuksia voidaan toteuttaa kahdenvälisillä yhteyksillä. On esitetty sekä arvioita, että noin 20 sovelluspalvelusta koostuvan sovelluksen rakentaminen kahdenvälisillä yhteyksillä on vielä hoidettavissa, mutta myös n. 200 sovelluspalvelun kokonaisuuksia on hallittu pääosin kahdenvälisten yhteyksien avulla. Tämä vaatii kuitenkin selkeää ohjeistusta ja palveluhakemistojen tarjoamista kehittäjille ja integraattoreille. Hakemistot voivat perustua kehittäjille suunnattuun dokumentaatioon tai sisältää myös koneellisesti tulkittavaa tietoa. Hakemistopalvelujen tarkentamista vaatimuksista riippuen mm. ebXML-rekisterit, UDDI-hakemistopalvelut sekä RDF-pohjaiset verkkoresurssien kuvaamistavat voivat olla hakemistojen teknisten ratkaisujen pohjana.

Palvelualueiden käyttö ei kuitenkaan vaikuta kaikkiin integrointitasoihin ja ratkaisuihin. Epäyhteensopivuudet erikseen rakennettujen sovellusten tai palvelujen semanttisissa ja toiminnallisissa perusratkaisuissa kuten tietoelementtien ja kokonaisuuksien yksityiskohtaisissa merkityksissä on hankalaa sovittaa yhteen väylän teknisten palvelujen avulla. Väylä voi tosin helpottaa tietojen yhdistelyjä ja jakoja sekä tarjota terminologia- ja vastaavuuksien määrittelypalveluja helpottamaan semanttista yhteensovittamista, tai täydentää ulkoisista lähteistä tietystä järjestelmästä puuttuvia tietoja. Lisäksi luvussa 2.2 kuvattujen integrointitapojen yhteensovittaminen on hankalaa, mikäli osassa osallistuvia sovelluksia tai palveluja on samojen tarpeiden ratkaisuun käytetty täysin erilaisia integrointitapoja. Esimerkiksi mikäli on hyödynnetty käyttäjäintegraatiota saman tiedon välitykseen yhdessä sovelluksessa ja sanomaintegraatiota toisessa, voi yhteensovittamisessa syntyä haasteita. Tällöin ei voida myöskään olettaa, että alustan kautta keskitetty seurantapiste voisi päästä käsiksi "sanomiin", joita toteutetaan käyttäjäliittymätasolla eikä alustan palvelujen kautta. Myös toiminnallisen viitemallin ja sen tietorajoitteiden (ks. luku 2.3) yhteensovittaminen palveluväylän avulla on haasteellista.

4 SOA ja standardit

4.1 Palvelukeskeinen yhteentoimivuuden standardointi

Toiminnan ja tietojärjestelmien yhteensovittaminen on ollut keskeinen arkkitehtuuri- ja yhteentoimivuustyön tavoite. Sekä kokonaisarkkitehtuuria että palveluarkkitehtuuria on pyritty hyödyntämään tämän tavoitteen saavuttamiseen. SOA:n useat tavoitteet kuten kyky vastata ketterästi uusiin vaatimuksiin ja liittää olemassa olevia sovelluksia ja palveluja osaksi uusia prosesseja ja koostettuja ratkaisuja tarvitsevat tuekseen yhteentoimivuusratkaisuja. Yhteentoimivuusstandardien kehittämisellä ja hyödyntämisellä tietojärjestelmäratkaisuisissa on pitkät perinteet. Standardit tukevat SOA-pohjaisissa ratkaisuisissa sekä uudelleenkäytön että yhteentoimivuuden tavoitteita.

Yhteentoimivuuteen liittyviä standardeja on kuitenkin hyvin monenlaisia lukuisiin eri tarkoituksiin. Perinteisiä yhteentoimivuusstandardeja ovat tietosisältöihin liittyvät järjestelmien, asiakirjojen tai rajapintojen määrittelyt, joissa on tyypillisesti käytetty tietopohjaista integraatiota. Runsaasti on myös standardoitu tiedon siirto- ja esitysmuotoja kuten sanomien tai asiakirjamuotojen rakenteita, tietotyyppejä ja muita yhteentoimivuuden perusrakenteita. 2000-luvulla myös palvelurajapintojen standardointi on yleistynyt, erityisesti keskitettyjen ja monissa prosesseissa tarvittavien ydinpalvelujen (common services) osalta.

Myös järjestelmien toiminnallisten ominaisuuksien (toimintojen, prosessien tai työnkulun osien) standardoitu määrittely on lisääntynyt viime vuosina. Järjestelmien arkkitehtuuriin (osien ja niiden välisten suhteiden sekä kehittämisperiaatteiden) osalta standardeja ei hyödynnetä vastaavassa laajuudessa kuin rajapintojen osalta. Tälläkin alueella on tosin standardoituja suosituksia ja viitemalleja.

SOA-periaatteita tehokkaasti tukevien standardien tulisi olla modulaarisia. Standardit muodostavat päällekkäisiä tasoja ja "pinoja". Standardit kuitenkin sisältävät usein myös oletuksia siitä, kuinka on ratkaistu jokin toinen integrointitaso kuin se mihin standardi varsinaisesti keskittyy. Sisällölliset (toiminnallisten rajapintojen ja semantiikan) standardit voivat nojautua siihen, että alemmalla tasolla esimerkiksi tiedonsiirto perustuu tietyn standardin käyttöön. Sisällölliset rajapinnat kuvataan standardoituja kuvaustapoja ja rajapintatekniikoita hyödyntäen. Myös turvallisuusratkaisuihin on saatavilla suuri joukko standardeja, lähtien tietoturvakäytännöistä ja päättyen yksityiskohtaisiin rajapinta-, salaus-, allekirjoitus- ja varmennestandardeihin.

Palvelukeskeisissä ratkaisuisissa korostuu toisaalta palvelujen toiminnallisten rajapintojen yhteensopivuus, toisaalta oletukset siitä miten esimerkiksi tietoturvaseikat hoidetaan. Erona aiempiin komponenttipohjaisen ohjelmistokehityksen malleihin palvelupohjaisessa kehittämistyössä ei voida nojautua siihen, että suoritusympäristössä olisi tietty vakioitu komponenttimalli, joka hoitaisi yhdenmukaisesti esimerkiksi turvallisuuteen, tietojen säilyttämiseen tai transaktionhallintaan liittyviä vaatimuksia. Tästä syystä näihin seikkoihin liittyvien oletusten, vaatimusten ja reunaehtojen dokumentointi on erityisen olennaista tietojärjestelmäpalveluja kehitettäessä ja yhdisteltäessä.

Moniin seitsentasoisen mallin yhteentoimivuustasoihin on saatavilla valmiita määrittelyjä ja standardeja. Kussakin ympäristössä ja projektissa on kuitenkin valittava, mitkä ovat olennaisimmat standardien avulla ratkaistavat seikat ja missä taas voidaan noudattaa paikallisia tai projektikohtaisia käytäntöjä. Uudelleenkäytettäviä ratkaisuja voidaan kaikilla tasoilla löytää erilaisilta "paikalli-

suustasolta": globaaleista tai eurooppalaisista standardeista, kotimaisista määräyksistä tai soveltamisoppaista, useissa ympäristöissä yhdenmukaisista tuoteratkaisuista tai yhden käyttäjäorganisaation vakioituista ohjeista. Viime kädessä kaikki tasot on kuitenkin ratkaistava viimeistään projekti- tai kehittämiskohdekohtaisen käytännön avulla.

Tässä raportissa standardilla tarkoitetaan tunnustetun osapuolen hyväksymää dokumenttia, jossa on määritelty yleistä ja toistuvaa käyttöä varten sääntöjä, ohjeita tai piirteitä tuotteille, prosesseille tai palveluille (ks. sanasto). Tunnustettu osapuoli voi olla virallinen standardointijärjestö, avointa standardointimallia noudattava teollisuuskonsortio tai muulla tavoin hyväksytty taho.

4.2 Palveluarkkitehtuuriin liittyvien standardien sijoittuminen integrointitasoille

Edellä kuvattuja uudelleenkäytettäviä standardeja ja määrittelyjä on runsaasti myös palveluarkkitehtuurin avulla tapahtuvaa kehitystä varten. Kuvassa 11 on joukko palveluarkkitehtuuria hyödyntävissä projekteissa ja organisaatioissa käytettyjä standardeja, jotka on sijoitettu standardin pääasiallisen vaikutusalueen mukaisesti luvussa 2.3 esitettyyn viitemalliin.

Teknisten liittymien tasolla SOA-arkkitehtuurimalleja voidaan toteuttaa käytännössä kaikilla hajautettujen sovellusten toteuttamiseen tarkoitetuilla rajapinta-, tietoliikenne- ja tietojen rakenteistamisen ja esittämisen tekniikoilla. 2000-luvun alkupuolella laajasti käytetyt Web services-tekniikat (kuten WSDL ja SOAP) ovat saaneet rinnalleen teknisesti yksinkertaisia REST-pohjaisia teknisiä ratkaisuja. Erityisesti organisaatioiden välisten ja laajalti myös organisaatioiden sisäisten palvelu- ja sovellusrajapintojen määrittelyissä nojaututaan yleisesti edelleen vahvasti XML-pohjaisiin ja XML Schema-määrittystä hyödyntäviin sanomien ja dokumenttien määrittelyihin. Integraation sisältöä kuvaavan tekniikan valintaan vaikuttaa olennaisesti se, kohdistuuko ratkaisuihin asiakirjapohjaisuuden (business documents) vaatimuksia kuten luettavuus myös käyttäjien kannalta, asiakirjan kokonaisuuden säilyttämisen tai allekirjoittamisen tarve. Perustason web-sovelluspalvelutekniikoiden yhdenmukaista soveltamista ovat edesauttaneet mm. WS-I:n integrointiprofiilit, joilla on vähennetty eri toteutusten ja välineiden epäyhteensopivuuksia.

APQC Process Classification Framework	Prosessit
OASIS SOA Reference Model, OMG SoaML, Group SOA ontology, BPMN, OSIMM	Open Kehitysprosessin liittymät
ebXML RIM, HL7 RIM, HL7 EHR-S FM, HL7 PHR-S FM, HL7 D-MIM	Toiminnallinen viitemalli
OASIS UBL, BizTalk specifications, HSSP, RosettaNet, ebXML CPP/CPA, (OMG Domain Technical Committees), HL7 Scheduling, HL7 Finland	Semantiikka
minimikontekstinhallinta	Toiminnalliset liittymät
IEEE P1723 (S3), OASIS RA4 SOA Foundation, Open Group SOA Reference Architecture	Sovellusinfrastruktuuri
WS-RM, WS-Security, SAML, XACML, WS-Trust, WS-SecureConversation, BPEL, SCA, WS-Policy, UDDI v3	Tekninen infrastruktuuri
WSDL, SOAP, WS-I, JMS, CORBA, HTTP, REST, FTP, XML-RPC	Tekniset liittymät

Kuva 11. Palveluarkkitehtuuriin yhteydessä hyödynnettyjen standardien sijoittumista integrointitasoille.

Teknisen infrastruktuurin tasolla on yritysympäristöissä käytössä jonkun verran viestinvälityksen luotettavuuteen, turvallisuusseikkojen hallintaan tai vaikkapa hakemistopalveluihin tarkoitettuja määrittelyjä, joita löytyy erikseen mm. web-sovelluspalvelujen laajennuksiin (WS-*) sekä esimerkiksi ebXML-sanomavälitykseen. Vanhemmissa olioväliohjelmistoissa olioväylän palveluilla ja komponenttisäiliöillä (mm. JEE- ja CORBA-ympäristöissä) on hoidettu useita näistä seikoista.

Sovellusinfrastruktuurin ja -arkkitehtuurin tasolla standardeja löytyy lähinnä palvelupohjaisen arkkitehtuurin viitemalleihin. IEEE:n P1723-standardia on määritelty palvelupohjaisen S3-viitearkkitehtuun (ks. Itälä ym. 2011) pohjalta. Kyseisessä mallissa esitetään kehittämis- ja arkkitehtuurityön ohjauksessa käyttökelpoinen SOA-viitearkkitehtuuri kerroksineen (Arsanjani ym. 2007). Lisäksi OASIS ja Open Group ovat kehittäneet SOA-viitearkkitehtuureja ja käsitelmalleja, joissa pyritään erityisesti määrittelemään kattavasti palvelukeskeisen kehittämisen käsitteet ja niiden väliset suhteet. Näitä malleja on pyritty sovittamaan yhteen myös eri mallinnusstandardien kuten SOAML sekä kypsyystasomallien kanssa (Kreger, Estefan 2009).

Yhteentoimivuuden käytännön toteuttamisen kannalta keskeisimpiä SOA-standardeja ovat palvelujen toiminnalliset rajapinnat sekä dokumenttipohjaisten tietojärjestelmäpalvelujen dokumenttimäärittelyt. Sekä toiminnalliset rajapinnat että niihin liittyvien tietojen ja toimintojen semantiikka ovat keskeisimpiä yhteentoimivuuden sisällöllisen toimivuuden kannalta sovittavia asioita. Näiden tason standardit keskittyvät toiminnallisiin ja sisällöllisiin seikkoihin ja liittyvät usein tietyn kohdealueen tai toimialan toimintaan tai dokumentteihin. Rajapinnat kuvataan yleensä teknisesti käyttäen tekniset liittymät-tason standardoituja mekanismeja. Tällä alueella eti standardointijärjestöt ovat tuottaneet pohjaa yhteentoimivuudelle jo vuosikymmenten ajan. SOA-soveltamisen haasteena ovat kuitenkin standardien tuottamisessa tehdyt oletukset ja viittaukset yksittäisiin väliohjelmistoihin, teknisiin protokolleihin tai oletukset käytetyistä integrointitavoista (esim. dokumenttipohjaisuus, sanomapohjaisuus, rpc-malli).

Toiminnallisten viitemallien alueella erityisesti tietoarkkitehtuurissa yleiskäyttöisiä viitetietomalleja on saatavilla toimialakohtaisesti (esimerkiksi HL7 RIM ja ebXML RIM). Yhtenäisiä kattavia tietomalleja on kehitetty myös kansallisesti toimialakohtaisella tasolla (esimerkiksi Sosiaalialan tietoteknologiahankkeen tietomalli) (Sosmeta 2011). Myös yhteisiä toiminnallisuuden viitemalleja (lueteloita ohjelmistojen sisältämistä toiminnoista) on standardoitu, esimerkiksi sähköisten potilaskertomusten toimintojen (HL7 EHR System Functional Model) osalta.

Kehitysprosessin liittymien tasolla on samantyyppinen tilanne kuin sovellusinfrastruktuurissa: eri konsortioiden viitemalleja ja standardiehdotuksia ja suosituksia on saatavilla tukemaan SOA-pohjaisen kehittämisen yleistä lähestymistapaa ja käsitteiden hallintaa (Kreger, Estefan 2009). Myös ehdotuksia SOA-pohjaisen kehittämisen prosessien kypsyystasojen standardoinniksi on tehty (OSIMM 2011). Kehitysprosessien kannalta olennaisia ovat myös mallintamisessa ja kehitystyön hallinnassa käytettävät standardit, jotka helpottava eri kehittäjien välistä viestintää.

Myös seitsentasomallin sovelluskeskeisten integrointitasojen ylä- ja alapuolella on SOA-kehittämisessä käyttökelpoisia standardeja. Esimerkiksi organisaatioiden prosessien tunnistamisen ja määrittelyn pohjana on mahdollista hyödyntää APQC-konsortion Process Classification Framework -määrittelyä, jossa on tunnistettu ja jäsennetty yleisesti yritysten ja organisaatioiden prosesseja. Malli jaottelee johdon ja operatiivisen tason prosessit 12 kategoriaan ja sisältää yli 1000 tunnistettua tyypillistä prosessia (APQC 2008).

Samalla integrointitasolla voi olla useita eri seikkoihin vastaavia standardeja, ja kaikkia integraatioissa sovittavia seikkoja ei yleensä ratkaista avoimien määrittelyiden tai standardien pohjalta. Useimmat tässä luvussa kuvatut standardit ovat teollisuusstandardeja ja pohjautuvat eri tyyppisten yritysten ja käyttäjäorganisaatiokonsortioiden määrittelyihin. Virallisessa globaalissa ISO-standardoinnissa on ollut käynnissä mm. DAPS-työryhmä (Distributed Application Platforms and Services, ISO/IEC JTC 1/SC 38), joka on toistaiseksi keskittynyt pääosin pilvipalveluiden alueelle kehitettäviin standardeihin sekä OSIMM-kypsyysmallin määrittelyihin. SOA-kehittämisen osalta on tuotettu mm. raportteja yleisistä teknisistä SOA-periaatteista. Suomessa SFS:n seurantaryhmä 310 (Verkkosovellukset) seuraa aihealueen kansainvälistä ISO/IEC-standardisointia.

4.3 Toimialastandardien ja palveluarkkitehtuurin yhteensovittamisen haasteet

Standardeja on yleensä kehitetty tukemaan useita erityyppisiä käyttötapauksia ja käyttökohteita. Standardointiprosessit nimenomaisesti pyrkivät edistämään monenvälistä ja laajaa hyödyntämistä. Tämä asettaa kuitenkin haasteita erityisesti toimialakohtaisille ja ihmisten toimintaa ja työnkulkuja lähellä oleville standardeille. Käytettäessä standardeja paikallisten tarpeiden täyttämässä ja prosessien tukemisessa nousee esiin runsaasti paikallisia tarpeita. Samoin nousee esiin poikkeuksia yleissääntöihin, joita pitäisi pystyä tukemaan, mikäli paikalliset tarpeet halutaan täyttää. Tästä nousee esiin tarve standardien paikalliseen soveltamiseen. Paikallinen sovittaminen kuitenkin voi vaikeuttaa ratkaisujen uudelleenkäyttöä ja yhteentoimivuuden saavuttamista.

SOA:n ja toimialakohtaisten standardien yhteensovittamisen haasteita on käsitelty tutkimalla kahta toimialakohtaista standardia: RosettaNet Order Management sekä HL7 versio 3 ajanvaraus -standardeja (Kotinurmi, Mykkänen, Itälä 2008), (Mykkänen, Tuomainen 2012). Standardit suhteutettiin SOA:n peruskäsitteiden löytymiseen sekä SOA-perusperiaatteisiin. Tutkimuksessa arvioitiin, missä määrin standardit tukevat tai ovat ristiriidassa suhteessa SOA-peruskäsitteisiin, perusperiaatteisiin ja käytäntöihin.

Palveluarkkitehtuuria (SOA) käsitellään tässä joukkona periaatteita ja käytäntöjä, joiden avulla pyritään palvelupohjaiseen kehittämiseen. SOA-kehittämistyössä pyritään siihen, että tietojärjestelmä-ratkaisut ovat koostettavissa uudelleenkäytettävistä sovelluspalveluista siten, että kyky reagoida muutoksiin tietojärjestelmäkokonaisuuden kehittämisessä paranee. SOA-kehittämisen keskeisiä suunnitteluperiaatteita ovat (Erl 2007):

- Sopimukset: palvelulla on hyvin määritelty rajapinta jossa määritellään formaalisti (sopimus) palvelun toiminta, sen tarvitsemat syötteet ja odotetut tulokset. Syötteet ja tulokset määritellään standardoitujen rakenteiden avulla.
- Löyhä kytkentä: palvelut ovat riippumattomia muiden palvelujen ja niitä kutsuvien tahojen tilasta ja kontekstista. Löyhän kytkennän kautta on helpompaa vaihtaa palvelujen toteutuksia ilman että tarvitaan muutoksia muihin palveluihin.
- Abstraktio: palvelut määritellään irrallaan yksittäisistä ohjelmistoresursseista. Tiedot, jotka eivät ole välttämättömiä abstraktoidaan rajapinnoista siten, että palvelua kutsuville tahoille riittää palvelun rajapinnan tunteminen. Niiden ei tarvitse tuntea palvelun toteutustapaa.
- Uudelleenkäyttö: palvelun tarjoamaa sovelluslogiikkaa voidaan käyttää uudelleen uusissa prosesseissa ja palveluissa, myös sellaisissa yhteyksissä tai sellaisten kutsujien toimesta joita ei tunnettu palvelun suunnittelun aikana.
- Autonomia: palvelujen suoritusympäristö voidaan ratkaista palvelukohtaisesti ja ympäristö asettaa mahdollisimman vähän riippuvuuksia palvelujen välillä.
- Tilattomuus: palvelun hallinnoima tilatieto ja hallinnoinnin kesto minimoidaan siten, että tilaa käsitellään vain kutsukohtaisesti, sovelluspalvelut ovat tilattomia.
- Löydettävyyys: palvelut tarjoavat metatietoja, joilla tuetaan niiden löytämistä ja tarjottujen palvelujen ymmärtämistä ja tulkintaa.

- Koostettavuus: palveluja voidaan yhdistellä muiden palvelujen kanssa uusien palvelujen muodostamiseksi.

Näiden peruseräperiaatteiden lisäksi useissa SOA-lähestymistavoissa korostetaan *liiketoimintatason palvelujen* määrittelyä. Tällaiset palvelut tarjoavat rajapinnan ja vuorovaikutusmallin tasolla, joka voidaan tunnistaa ja liittää sisällöllisesti merkityksellisiin liiketoimintaprosesseihin. Teknisten liittymien tasolla SOA-kehittämisessä korostuvat rajapinta- ja viestinvälitystekniikat, ja web-sovelluspalvelujen tekniikat kuten WSDL ja SOAP laajennuksineen ovat yleisesti käytettyjä. Palveluväylää (ks. luvut 3.1 ja 3.3) käytetään usein tarjoamaan teknisen infrastruktuurin palveluja SOA-kehittämiseen ja suoritussympäristöön. SOA-hankkeissa ovat korostuneet myös eri arkkitehtuurinäkökulmien ja periaatteiden erottaminen toisistaan, kuten toiminnallisuuden erottaminen tiedosta ja sen merkityksestä sekä palvelujen tarjoamisen erottaminen niiden kutsumisesta. Lisäksi SOA-kehittämisessä olennaista on tuki palveluista koostuvien prosessien koostamiselle ja määrittelylle.

Luvussa 3.2 käsitellyistä standardeista liiketoiminnan kannalta keskeisissä kehittämiskohteissa korostuvat toiminnallisten rajapintojen ja semantiikan standardit sekä tekniset protokollat, joilla niitä tuetaan ja toteutetaan. Sekä virallisten standardiorganisaatioiden että teollisuuskonsortioiden avoimet yhteentoimivuuden määrittelyt voidaan luokitella yleisesti toimialariippumattomiin (horisontaalisiin) ja toimialakohtaisiin (vertikaalisiin) standardeihin (Nelson 2003). Horisontaalisia standardeja ovat esimerkiksi yleiset web-sovelluspalvelujen standardit ja arkkitehtuurin viitemallistandardit. Vertikaalisia standardeja ovat tyypillisesti toiminnallisten rajapintojen ja semantiikan standardit sekä tietoihin liittyvien viitemallien standardit. Vertikaalistanstandardeissa SOA-kehittämisen tai standardien yhteensovittamisen haasteita syntyy, mikäli vertikaaliset standardit sisältävät oletuksia tai viittauksia integrointi- ja vuorovaikutusmalleista tai käytetystä infrastruktuurista (joiden standardit ovat pääosin horisontaalisia).

Vertikaalisten standardien käyttäminen palvelupohjaisten ratkaisujen pohjana on järkevää, koska standardoinnissa on usein runsaasti valmista pohjaa paikallisten vaatimusten täyttämiseen, ja standardointityön aikana on standardeihin kertynyt myös seikkoja, jotka paikallisessa projektissa voitaisiin havaita vasta projektin myöhäisessä vaiheessa. Vertikaalistanstandardien käyttäminen saattaa kuitenkin edellyttää kompromisseja SOA-periaatteiden tiukasta noudattamisesta tai oletuksista siitä, mitä standardeja eri kehittämiskohteissa kullakin yhteentoimivuustasolla käytetään tarpeiden täyttämiseen.

SOA-periaatteiden ja SOA-pohjaisten horisontaalistanstandardien sekä vertikaalistanstandardien välillä voidaan tunnistaa usean tyyppisiä konfliktityyppejä:

- ristiriidat horisontaalistanstandardin ja vertikaalistanstandardin välillä (esim. SOA-teknologian suhteen) (HV-konflikti)
- ristiriidat SOA-periaatteiden ja vertikaalistanstandardin välillä (esim. karkeajakoisuus) (SV-konflikti)
- ristiriidat vertikaalistanstandardien ja paikallisten vaatimusten välillä (RV-konflikti).

Taulukossa 1 käsitellään RosettaNet:in ja HL7 versio 3:n valittujen osa-alueiden arviointia suhteessa palveluarkkitehtuurin peruskäsitteiden ja -tekniikoiden sekä eri näkökulmien erottamisen kannalta (Kotinurmi ym. 2008).

Taulukossa 2 käsitellään tarkemmin palveluarkkitehtuurin peruseräperiaatteiden toteutumista tarkisteluissa standardeissa ja -standardiperheissä.

SOLEA

Taulukoissa on arvioitu yleisellä tasolla sitä, kuinka hyvin standardista löytyy tai kuinka hyvin se tukee kulloinkin tarkasteltavaa palveluarkkitehtuurin ominaisuutta, käyttäen seuraavaa asteikkoa:

- ++ tukee suoraan tai helposti SOA-periaatteita ja piirteitä
- + SOA-käsitettä, periaatetta tai piirrettä vastaavat käsitteet voidaan tunnistaa tai määrittellä
- on haasteellista löytää tai määrittellä SOA-käsitettä, periaatetta tai piirrettä vastaavat käsitteet
- löytyy ristiriitoja suhteessa SOA-käsitteisiin, periaatteisiin tai piirteisiin.

Molemmissa arvioituissa standardeissa painopiste on tietosisällöissä, toiminnallisissa rajapinnoissa ja riittävän yhteisen semantiikan määrittelyissä. Sekä pakollisuuksien että niiden integrointitasojen osalta joihin standardit eivät ota kantaa, on ratkaisuja täydennettävä muilla standardeilla tai projektiokohtaisilla käytännöillä. SOA-periaatteiden noudattamiselle on joitakin esteitä kuten oletukset käytetyistä teknisistä protokollista ja se että toimintoihin liittyviä vastuita ei ole määritelty palvelupohjaisesti. Tosin määriykset eivät estä monien SOA-käytäntöjen noudattamista kehittämissuhteissa.

HL7-sanomastandardeissa havaituista palvelupohjaisen kehittämisen haasteista on noussut tarpeita kehittää standardeja siten, että niissä voidaan entistä paremmin tukea palvelukeskeistä integraatiota ja eri integrointitapoja. Tätä varten HL7-organisaatiossa on mm. luotu SAIF-kehittämismalli, jossa huomioidaan nykyisiä sanomamenetelmiä paremmin palvelupohjaisuuden tarpeet ja vaikutukset (ks. luku 5.4).

Taulukko 1. Palveluarkkitehtuurin peruskäsitteiden ja yhteentoimivuuden keskeisten osa-alueiden suhde tutkittuihin standardeihin.

SOA peruspiirre	RosettaNet PIP, klusteri 3 "Order Management"	HL7 versio 3, Ajanvaraus-sovellusalue (Scheduling)
Palvelu (liiketoimintataso)	Roolit ja myös laatuvaatimukset määritelty (++)	Sovellusroolit määritelty (+)
Rajapinta (joukko toisiinsa liittyviä toimintoja)	Single action PIP ja Double action PIP -mallit saatavilla, PIP-mallit muodostavat yhdessä toisiinsa liittyviä prosesseja ja laajempia toimintoja (++)	Sovellusroolit, sanoman vastaanottajien velvollisuudet, ei tue täysin rajapina-ajattelua (-) (SV)
Operaatio (toiminnon tarjoava ominaisuus)	Lähettäminen ja vastaanotto määritelty korkealla tasolla (+), kukin PIP vastaa jotta-kuinkin yhtä toimintoa	Lähimmäs osuvia käsitteitä toisiinsa liittyvien sanomien sekvenssit sekä sovellusrooleista ja viestityypeistä yhdistely (-)
Viesti (sanoman tietosisällön semantiikan määrittelyt)	Semanttiset määrittelyt mukana. Sekä validoitava skeema että tekstuaaliset kuvaukset määritelty. (+/++)	Tarkat sanomamäärittelyt, tarkennetut sanomatietomallit (R-MIM) ja tekstuaaliset kuvaukset määrittely (+/++)
Rajapintatekniikat	Liittyvät viestitekniikan määrittelyihin (++)	XML toteutusteknologiamäärittely määrittelee HL7-mallien serialisoinnin XML-skeemaan (++) , HL7 Web Services transport profiilissa epästandardi vastaavuus WSDL-määrittelyyn (--)
Sanomanvälitystekniikat	Tuki RNIF, ebMS, AS2 tai Web services-sanomavälitykselle (++)	HL7 Web services siirtoprofiili käyttää SOAP, WS-I-määrittelyä (+), HL7 v3 transmission wrapper määrittelyssä seikkoja joita voisi toteuttaa myös WS-* tekniikoilla, (--), myös ebMS and MLLP siirtoprotokollat määritelty
Vaadittu tai oletettu tekninen infrastruktuuri / palveluväylä	Määrittelyä tukevia tuotteita ja alustoja saatavilla (+)	Sanomavälitysalustoissa ja tuotteissa tukea määrittelyille(+)
Semanttiset määrittelyt	Semanttiset määrittelyt perustuvat vapaaseen tekstiin. (+)	Yhteinen viitetietomalli RIM, rajoitetut domain- ja viestitason mallit, semanttiset määrittelyt koodistojen ja tekstin kautta (+)
Toiminnallisen ja teknisen / infrastruktuuritason määrittelyjen erottaminen	Sanomamäärittelyissä sekä teknisiä että toiminnallisia sisältöjä. (+). Sanomien käsittelyä hankaloittaa sekä siirtoon että sisältöön liittyvien osien mukanaolo.	Koostetut HL7-sanomat sisältävät toiminnallisen sisällön, työnkulun rakenteet (control act) ja sanomavälitysrakenteet (transmission) . Kerroksilla ei selkeää toiminta- ja vuorovoikutusosien erottamista. (-)
Toimintojen ja tietojen / semantiikan erottaminen	PIP-tasolla tiukat semanttiset määrittelyt, sanomatasolla yleisemmät määrittelyt.	Toiminnalliset vastuut usein implisiittisiä tai käyttävät sanomainstansseissa vaihtoehtorakenteita (--)

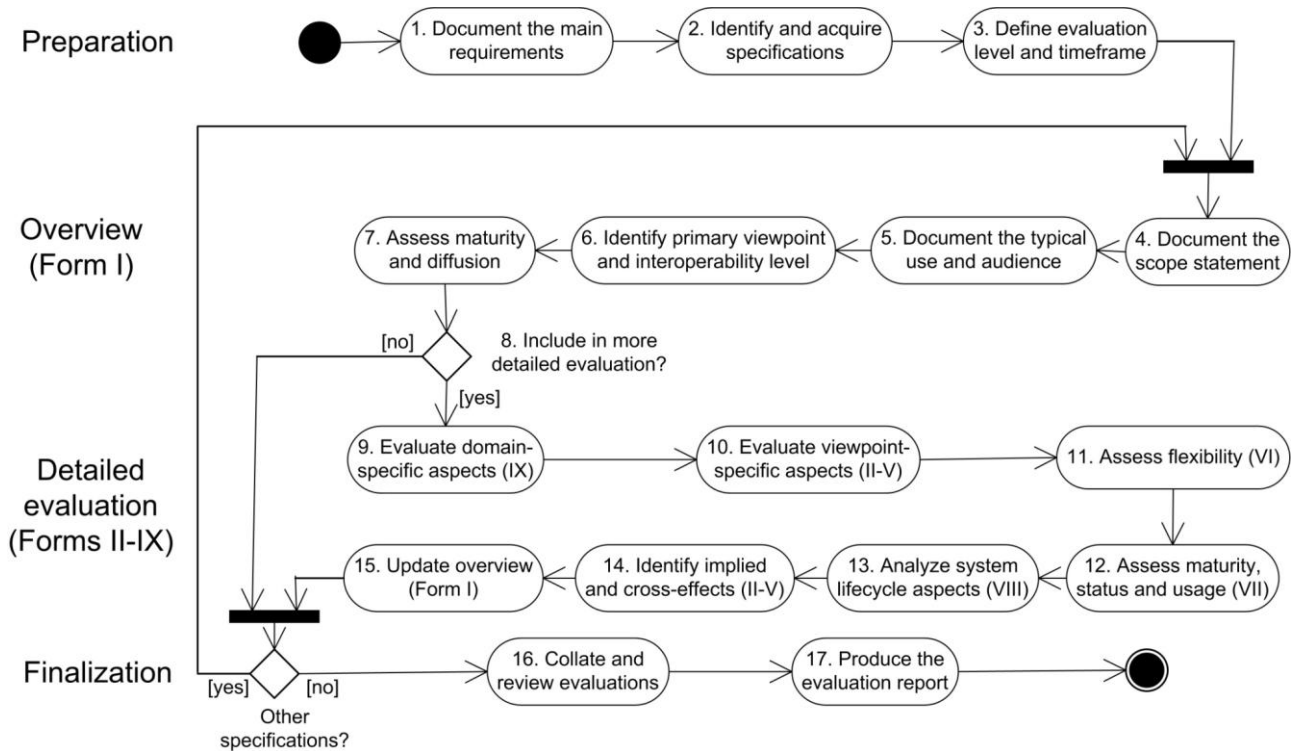
Taulukko 2. Palveluarkkitehtuurin periaatteiden suhde tutkittuihin standardeihin.

SOA periaate	RosettaNet PIP, klusteri 3 "Order Management"	HL7 versio 3, Ajanvaraus-sovellusalue (Scheduling)
Sopimus (operaatio, syötteet, tulokset)	Tiedot PIP-määrittelyissä ihmisen luettavassa muodossa, vapaaehtoisuutta (-).	Tiedot tarkasti sanomatasolla operaatiotason sijaan, ihmisen luettavassa muodossa ja koodistoina, paljon vapaaehtoisuutta (-)
Löyhä kytkentä (riippumattomuus muista palveluista)	Riippuu kustakin toteutuksesta PIP-määrittelyt karkeajakoisia tietokomponentteja (+)	Itselliset sanomat ja löyhä kytkentä (sanomapohjainen malli) (++)
Abstraktio (ohjelmistoresursseista, turhien tietojen poistaminen)	Hyvin abstraktoitu yksittäisistä ohjelmistoista (++) , paljon yksityiskohtaisia vapaaehtoisia tietoja sanomissa (-)	Hyvin abstraktoitu yksittäisistä ohjelmistoista (++) , paljon yksityiskohtaisia vapaaehtoisia tietoja sanomissa (-)
Uudelleenkäyttö (uusissa prosesseissa ja palveluissa)	Riippuu kustakin toteutuksesta. XML schema PIP-määrittelyt hyvin uudelleenkäytettäviä. DTD PIPs vähemmän uudelleenkäytettäviä. (+)	Palvelulogiikkaa ei ole usein eksplisiitisti määritelty, jossain määrin sovellusroolien kautta (-), mutta sanomia ja tietomalleja voidaan käyttää uudelleen (+)
Autonomia (minimoidut riippuvuudet muista palveluista)	Yksittäinen PIP on autonominen, mutta yksittäisissä toteutuksissa ja suoritusympäristöissä voi olla eroja. (++)	Riippuvuudet pääosin tietomallin tasolla, vain vähän oletuksia suoritusympäristöstä (+)
Tilattomuus	Sanomissa kaikki tarvittavat tiedot, ei sanomien osalta tarvetta sessionhallinnalle tai tilallisille palveluille (++)	Sanomissa tyypillisesti kaikki tarvittavat tiedot, toisiinsa liittyvät sanomat varmistetaan yksilöinnin avulla, ei tarvetta sessionhallinnalle tai tilallisille palveluille (++)
Löydettävyys (metatiedot löytämiseen ja tulkintaan)	Ei juurikaan palveluihin liittyviä metatietoja. Nojautuminen yhdenmukaisiin tai paikallisiin dokumentointi- ja toteutussuosituksiin (-)	Ei juurikaan palveluihin liittyviä metatietoja. Nojautuminen menetelmässä kuvattuihin dokumentointitapoihin ja paikallisiin soveltamisoppaisiin (-)
Koostettavuus (uusien palvelujen ja prosessien kokoaminen)	Ei kuvattu osana standardia (-) mutta yhteiset tietomallit saman sukupolven PIP-mallien välillä (+)	Ei kuvattu osana standardia (-) mutta yhteiset viite-, domain- ja sanomatietomallit (+)

4.4 Standardien arviointi ja valinta

Osana integrointiratkaisujen määrittelyä tai standardisalkun kokoamista on olennaista tehdä valintoja siitä, mitä standardeja tulisi hyödyntää. Koska kaikkia potentiaalisia standardeja ei voida kokeilla käytännössä, on löydettävä malleja siihen, kuinka voidaan valita kehityskohteen tarpeita tukevia standardeja. Edellä kuvattua viitemalleja hyödyntäen on kehitetty arviointi- ja valintamenetelmä siihen, kuinka standardeja voidaan arvioida ja kuvata niiden ominaisuuksia ja käyttökelpoisuutta tilannekohtaisia vaatimuksia vasten (Mykkänen, Tuomainen 2008). Arviointi- ja valintamallissa on joukko standardien ominaisuuksia ja eri arkkitehtuurinäkökulmia seuraavia tarkastelukysymyksiä (lomakkeita) sekä prosessi, jonka avulla arviointia suoritetaan. Arvioinnissa on olennaista muodos-

taa yleiskuva standardin kohdealueesta sekä sen kypsyydestä ja käytön laajuudesta. Lisäksi voidaan tarkastella tarkemmin sen vaikutuksia tietoarkkitehtuuriin ja semantiikkaan, toimintoihin ja vuoro-vaikutukseen, sovellusarkkitehtuuriin, teknisiin liittyisiin, laajennettavuus- ja tarkkuusominaisuuksia, markkina- ja viranomaisstatusta sekä vaikutusta järjestelmän elinkaareen. Standardien arviointiprosessin yleiskuva esitetään alla olevassa kuvassa. Nopeassa arvioinnissa suoritetaan vain ensimmäiset vaiheet kullekin arvioitavalle standardille, kun taas perusteellisemmassa arvioinnissa tarkastellaan yksityiskohtaisesti eri näkökulmiin liittyviä standardin ominaisuuksia.



Kuva 12. Yhteentoimivuusstandardien arviointi- ja valintaprosessi (Mykkänen, Tuomainen 2008).

5 Esimerkkejä palvelukeskeisyydestä integraatiossa ja standardeissa

Tähän lukuun on koottu kotimaisia ja kansainvälisiä esimerkkejä palveluarkkitehtuuriin liittyvästä integraatiosta ja standardoinnista. Esimerkit on erityisesti koottu SOLEA-hankkeen standardointiin ja terveydenhuoltoon liittyvistä työkohteista ja kumppanuushankkeista:

- Esimerkkinä toimintopohjaisesta palvelujen tunnistamisesta pohjana integraatioon käsitellään SOLEA-hankkeen palvelutapahtumat-työkohtetta (luku 5.1)
- Esimerkkinä palvelujen kerrostamisesta, palvelupohjaisesta prosessimäärittelystä ja palveluväylän hyödyntämisestä käsitellään Satakunnan sairaanhoitopiirin tapahtumakeskeistä päivystys-esimerkkiä (luku 5.2)
- Esimerkkinä konkreettisesta integraatiotopologiasta ja palvelualustatuotteen käytöstä integraatiossa käsitellään KanTa-liittymäpisteen toteutusta Intersystems Ensemble-alustan avulla (luku 5.3)
- Esimerkkinä standardointiorganisaation kehittämästä palvelukeskeisestä mallista standardien tuottamiseen käsitellään HL7 SAIF-mallia (luku 5.4)
- Esimerkkinä toimialakohtaisten SOA-standardien kehittämisestä käsitellään HSSP-yhteistyöprojektia ja sen tuottamia palvelumäärittäyksiä (luku 5.5)
- Esimerkkinä vakiintuneiden standardien, rajattujen integraatioprofiilien ja SOA:n yhteensovittamisesta käsitellään IHE-integraatioprofiilien sovittamista SOA-arkkitehtuuriin (luku 5.6).

5.1 Palvelupohjaisuus SOLEA palvelutapahtumat -työkohteessa

Palvelutapahtuma on keskeinen käsite terveydenhuollon kansallisen arkkitehtuurin asiakirjojen muodostamisessa ja niiden käytön hallinnassa. Palvelutapahtumien arkkitehtuuritarkennukset nostettiin SOLEA-hankkeen yhdeksi case-työkohteeksi syksyllä 2009. Työn tavoitteena oli tarkentaa arkkitehtuuripelisääntöjä, joiden avulla palvelutapahtuma-käsitettä voidaan käsitellä yhdenmukaisesti eri tietojärjestelmissä ja organisaatioissa, mukaan lukien potilashallinnon ydinprosesseihin liittyvät tutkimuksen tai hoidossa käytetyt erillisjärjestelmät (Mykkänen ym. 2012a). Tarkennustarpeita liittyi mm. käsitteisiin ja niiden tulkintaan, palvelutapahtumien ja hoitoprosessien suhteisiin sekä palvelutapahtuman elinkaareen. Tavoitteena oli luoda pohjaa tarkemmille tietosisältöjen ja rajapintojen määrittelyille, tunnistaa keskeisimmät tarvittavat uudet rajapinnat sekä mahdolliset päivitykset olemassa oleviin rajapintamäärittelyihin sekä pyrkiä löytämään malleja siihen, kuinka erityyppiset järjestelmät ja niiden tuottamat potilasasiakirjamerkinnot saadaan sovitettua palvelutapahtumien kautta tapahtuvaan dokumentaation hallintaan. Työssä tuotettiin ja koottiin myös käsitelmalleja ja -määrittelyjä, tietomalleja, vaatimuksia, esimerkkiskenaarioita sekä tietojärjestelmäarkkitehtuurin suunnittelupäätöksiä.

Palvelupohjaisen lähestymistavan pohjana työkohteessa toimi palvelutapahtumien ja asiakirjojen (tiedon)käsitteilyyn liittyvien käyttäjien toimenpiteiden ja järjestelmien toimintojen analyysi (Myk-

känen ym. 2011). Palvelutapahtumatietojen käsittelyyn liittyvät tietojenkäsittelytehtävät liittyvät eri tilanteisiin kuten lähetteen saapuminen, ajanvaraukseen liittyvät toimenpiteet, lähetteen käsittely, sisäänkirjaukset jne. Toiminnan kannalta merkitykselliset tehtävät (toiminnot) analysoitiin osallistuvien henkilöiden ja tietojärjestelmätyyppien kannalta (ks. taulukko 3). Kaikki toiminnot kuvattiin tarkemmin, jotta pystyttiin tunnistamaan palvelutapahtumien hallintaan liittyvät hienojakoisemmat toimenpiteet ja tietojenkäsittelytehtävät. Nämä hienojakoiset tietojenkäsittelytehtävät sijoitettiin hienojakoisiin tietojärjestelmäpalveluihin (ks. taulukko 4), joita voidaan sijoitella eri tietojärjestelmiin (ks. taulukko 5). Vaihtoehtoisten toteutusarkkitehtuurimallien arvioinnissa keskeistä oli näiden palvelujen sijoittelu eri tietojärjestelmiin tai yhteiskäyttöisiin palveluihin ja eri toteutustasoille. Toteutustasoina käsiteltiin palvelujen toteutusmahdollisuuksia paikallisesti (yhden palvelunantajan toiminnassa), alueellisesti (eri palvelun antajien välillä) ja valtakunnallisesti. Tunnistettujen palvelujen hyödyntämistä asiakirjojen muodostamisessa ja arkistoinnissa kuvattiin kaavioiden ja sanallisten kuvausten avulla (ks. kuva 13). Työssä ei tehty yksityiskohtaisia rajapintojen tai palvelujen määrittelyjä.

Taulukko 3. Palvelutapahtumien hallinnan roolit / toiminnot matriisi (Mykkänen ym. 2012a).

X: aina mukana - x: mahdollinen:

	Asiakas	Ammattihenkilö	Palvelunantajan tietojärjestelmät	Asiakkaiden tietojärjestelmät	Arkisto
Lähetteen tekeminen		X	X		X
Lähetteen vastaanottaminen		X	X		x
Potilaan laittaminen jonoon		X	X		x
Ajanvarauksen tekeminen lähetteen pohjalta	x	X	X	x	x
Ajanvarauksen tekeminen ilman lähetettä	x	X	X	x	x?
Ajanvarauksen peruminen	x	X	X	x	x?
Ilmoittautuminen ja sisäänkirjaus ennakoiden		X	X		x
Potilasta koskevien tietojen tarpeen määrittely	x	X	x	x	x
Luovutuksen perusteen määrittely	x	X	X		x
Saatavilla olevien asiakirjojen selvittäminen		X	X		X
Tarpeellisten asiakirjojen noutaminen		X	X		X
Haettujen asiakirjojen hyödyntäminen		X	x		
Hoidollisen merkinnän kirjaaminen / tekeminen		X	X		X
Palvelutapahtumaan kohdistuvan tahdonilmauksen antaminen tai muuttaminen	X	x	x	x	X
Potilaan siirto osastolle / osastojen välillä		X	X		x?
Potilaan kotiutus / käynnin päätyminen		X	X		X
Tutkimus- tai konsultaatiopyynnön tekeminen		X	X		x
Tutkimus- tai konsultaatiopyynnön vastaanottaminen ja toteutus		X	X		x

Taulukko 4. Palvelutapahtumien hallinnan palvelut-roolit / tehtävät matriisi (Mykkänen ym. 2012a).

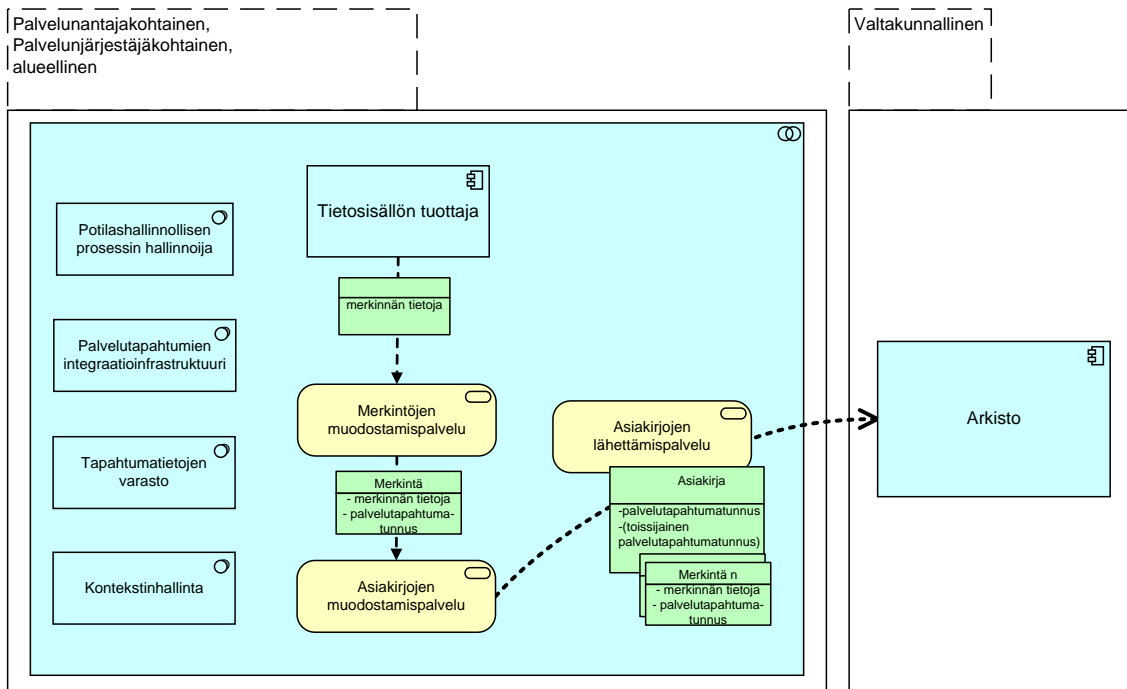
Potentiaaliset roolit palveluittain on merkitty seuraavasti: (k=kutsuja tai käynnistäjä, P=sovelluspalvelun tarjoaja, l=tietolähde), paikallisesti usein toteutetaan vain osa merkityistä rooleista (suositellaan suoritettavaksi karsintaa vähintään paikallisella toteutustasolla!).

	Käyttäjä (ammatihenkilö, asiakas)	Arkistopalvelu	Asiakirjojen muodostamispalvelu	Asiakirjojen lähettämispalvelu	Merkintöjen muodostamispalvelu	Tapahtumatietojen varasto	Asiakirjojen hakupalvelu	Luettelon hakupalvelu	Tietosisällön tuottaja	Tietosisällön hyödyntäjä	Potilashallinnollisen prosessin hallinnoija	Palvelutapahtumien integraatioinfrastruktuuri	Kontekstinhallinta
Aktiivisista palvelutapahtumista yhden valinta	k,l		k		k, P	l, P			k, P		P	l	l
Aktiivisten palvelutapahtumien selvittäminen		l?			k, P	l, P	l?		k, P		k,P		l
Asiakirjan arkistointi	k	P	l	P							k	k?	
Asiakirjan haku arkistosta	k	P				l?	P	l		k,l	l		
Asiakirjojen kuvailutietojen haku arkistosta	k	l,P				l?		P		k,l	l		
Asiakirjan liittäminen palvelutapahtumaan	l		P		l	l			l		k,l	P,l	l
Asiakirjan muodostaminen merkinnöistä			P		l				l, k		k		
Tietojen näyttämisen viivästyttäminen	l,k		P?		P?				P, k				
Asiakkaan tapahtumien listaus		l			k	P,l			k		k		
Asiakkaan valinta vastaanoton tai osaston potilasluettelosta	k								P,l		P,l		
Merkinnän liittäminen palvelutapahtumaan	l				P	l			k,l		k,l	l,P,k	l
Merkinnän tai asiakirjan siirtäminen toiseen palvelutapahtumaan	k		P?		P?				k,l		k,l		
Palvelutapahtumaan kohdistuvan tahdonilmauksen tarkistaminen	l	P					k,l?			k,l	k,l		
Merkinnän liittäminen prosessitapahtumaan	l				P	l			k,l, P		l,P,k	l,P,k	l
Palvelutapahtuman muodostaminen	k		P?		P?	P			k		P,k	k?	
Palvelutapahtuman päättäminen	k								P, k		P,k		
Palvelutapahtuman tietojen haku		l	k		k	P,l			k		k	k?	
Palvelutapahtuman tietojen muuttaminen	l	P	P?		P?						P,k	k?	
Palvelutapahtuman tietojen tallennus arkistoon	l	P	P?	P?		P?			l,k		l,k	k?	
Palvelutapahtumatunnuksen luonti			P?, k		P?, k	P?			P?, k		P?, k	k?	
Palvelutapahtumatunnuksen selvittäminen prosessitapahtuman perusteella						P?			k, l?		P?, k,l	P?,k	l?
Palvelutapahtumatunnuksen välittäminen			k?		k?	l?			k		k,l	k?	P

Taulukko 5. Palvelutapahtumien hallinnan järjestelmät / palvelut matriisi.

p=palvelun tarjoaja (varmasti jossakin tiedossa olevassa ympäristössä), m=mahdollinen palvelun tarjoaja.

	Arkistopalvelu	Asiakirjojen muodostamis- palvelu	Asiakirjojen lähettämispal- velu	Merkintöjen muodostamis- palvelu	Tapahtumatietojen varasto	Asiakirjojen hakupalvelu	Luettelon hakupalvelu	Tietosisällön tuottaja	Tietosisällön hyödyntäjä	Potilashallinnollisen proses- sin hallinnoija	Palvelutapahtumien integraa- tioinfra	Kontekstinhallinta
Valtakunnallinen palvelu	P	?		?	m	P	P					
Alueellinen tai paikallinen palvelu		m	m	m	m	m	m		m	m	P	P
(Potilashallinnon) ydinjärjes- telmä		P	P	P	P	P	P	P	P	P		m
Yksikkökohtainen erillisjär- jestelmä		P	m	m		P	m	P	m			
Organisaatioiden yhteinen erillisjärjestelmä		P	m	m		P	m	P	m			



Kuva 13. Asiakirjan arkistointi hyödyntäen palvelutapahtumien hallinnan tietojärjestelmäpalveluja ja -rooleja (Mykkänen ym. 2012a).

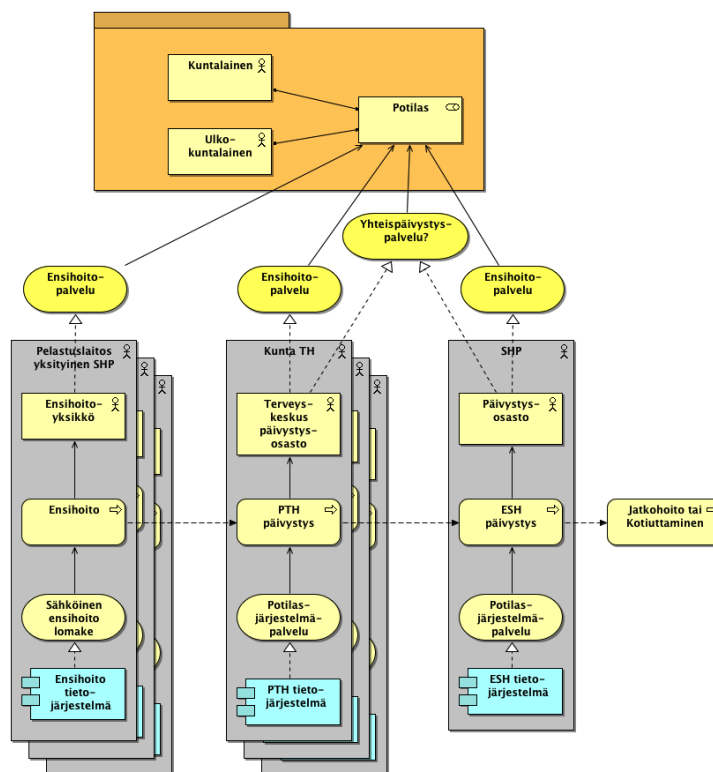
Palvelutapahtumat-työkohteen tulokset julkistettiin SOLEA-johtoryhmässä 9.12.2009 ja 8.6.2010. Julkistettuja tuloksia hyödynnettiin eri toimijoiden toimesta Kelan ja HL7 Finland ry:n projekteissa, joissa tarkennettiin erillisjärjestelmien KanTa-integraatioita sekä palvelutapahtumien hallintaan tarvittavia päivityksiä järjestelmien integraatioissa käytettyihin standardien soveltamisoppaisiin.

Tulokset toimivat pohjana myös Viila-työryhmän tarkennuksille, jotka kohdistuivat valtakunnallisiin KanTa-määrittelyihin sekä sosiaalialan tietoteknologiahankkeessa tehdyille asianhallinnan tarkennuksille. Palvelutapahtumista julkistettuun raporttiin päivitettiin lisäksi työhön osallistujille suoritun kuvaustapakyselyn tulokset (Mykkänen ym. 2012a).

5.2 Tapahtumakeskeinen arkkitehtuuri Satakunnan sairaanhoitopiirisä

Esimerkkinä palvelukeskeisestä integraatioiden määrittelyistä esitellään Satakunnan sairaanhoitopiirin kanssa SOLEA-hankkeessa tehtyä mallinnusta päivystyksen toiminnan ja tietojärjestelmien kehittämisestä.

Satakunnan sairaanhoitopiirin kuntayhtymä käsittää Porin ja Rauman ympäristöstä kaikkiaan 20 jäsenkuntaa, joissa asuu kaikkiaan 226 000 asukasta. Sairaanhoitopiiri tarjoaa jäsenkunnilleen erikoissairaanhoidon palvelut, erityishuoltopiirin palvelut sekä yhteispäivystyksen palvelut lähikuntien terveyskeskusten kanssa. Alueen ensihoidon palveluiden järjestämistä esittää oheinen Archimate-kuva.



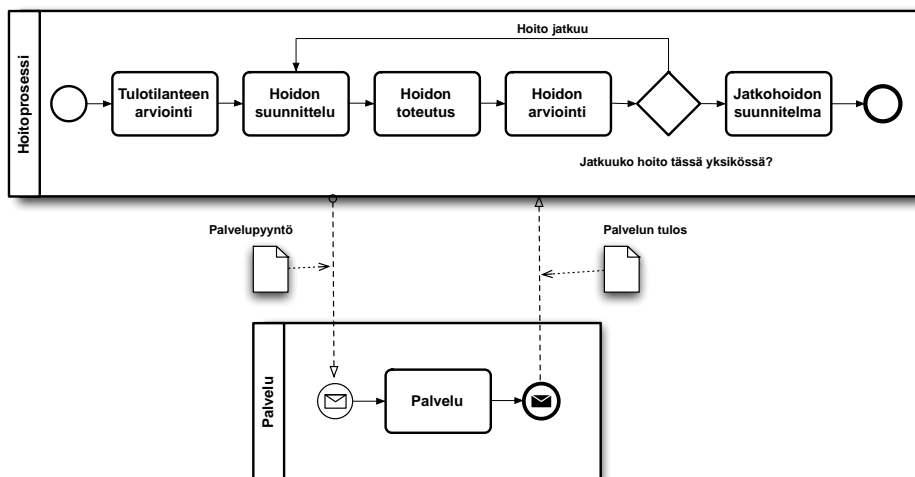
Kuva 14: Satakunnan alueen päivystystoiminta ja ensihoito

Kiireellisen hoidon tarpeessa kunnallinen pelastuslaitos tai yksityinen ambulanssi tulevat paikalle ja toteuttavat ensihoidon sekä tarvittaessa kuljettavat potilaan terveyskeskuksen tai keskussairaalan erikoissairaanhoidon päivystysvastaanottoon. Esimerkkimme rajoittuu keskussairaalaan ja siellä rajattuun kehittämiskohteeseen.

Keskussairaalan päivystysosastolla yhtenä ongelmana on ollut se, että potilaat toisinaan joutuvat odottamaan hyvinkin pitkään hoidon eri vaiheissa ennen kuin heidät kotiutetaan tai siirretään osastolle jatkohoitoon. Ongelman syytä selvitetessä osoittautui, että yksi odotusta aiheuttava tekijä on puuttuva ilmoitus siitä, että jokin lääkärin tilaama tutkimus tai konsultaatio on valmistunut ja että

lääkäri voisi jatkaa potilaan diagnosointia ja hoidon suunnittelua. Hoitajat joutuvat vähän väliä katsomaan potilasjärjestelmästä, joko palvelupyynnön tulos on valmistunut. Tilannetta ja mahdollista ratkaisua lähdettiin mallintamaan palveluarkkitehtuurin periaattein.

Lähtökohdaksi otettiin potilaan hoitoprosessi, joka jo aikaisemmin on mallinnettu BPMN-notaatiolla oheisen kuvan mukaisesti. Ydinprosessi on hyvin yleinen ja sisältää potilaan kulkuun liittyvät tehtävät. Kaikki potilaan hoitoon liittyvät tehtävät voidaan nähdä palveluina, joita pyydetään palvelupyynnöllä ja vastataan palvelun tuloksella.

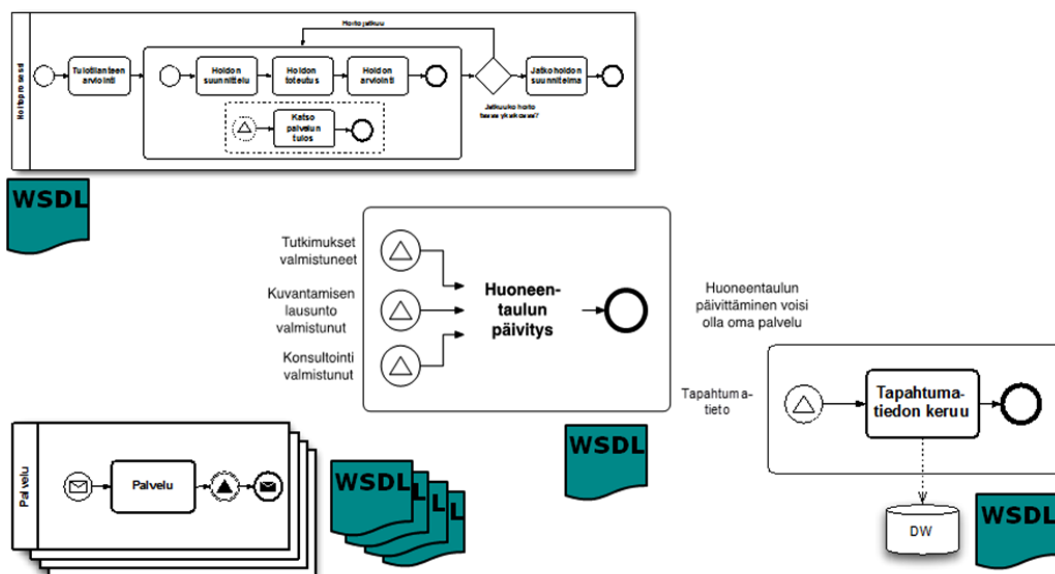


Kuva 15: BPMN-kaavio potilaan hoitoprosessista ja palveluista

Tyypillisesti palvelupyynnöt ja kohdistuu mm. laboratoriotutkimuksiin, kuvantamisen tutkimuksiin, erikoislääkärin konsultaatioihin jne. Oheisessa mallissa kiinnostava seikka on prosessin ja palvelun välinen sanomien välitys, joka on piirretty hoitoprosessi-uimaradan reunaan kiinni. Merkintä tarkoittaa, että palvelupyyntö on mahdollista tehdä mistä tahansa hoitoprosessin tehtävästä. Tällainen menettely onkin tyypillistä terveydenhuollossa, jossa lisätiedon ja palvelun tarvetta saattaa tulla missä hoidon vaiheessa hyvänsä. Toisaalta malli on niin yleinen, että siitä ei selviä, mitä tapahtuu hoitoprosessissa, kun palvelun tulos on valmis vastaanotettavaksi ja käsiteltäväksi.

Yhtenä ratkaisumallina tiedon saamiseksi palvelupyynnön valmistumisesta on ajateltu "huoneentaulu" -ratkaisua. Ajatuksena olisi seurata potilaskohtaisesti tilannetta, mitä palveluita potilaalle on tilattu ja ovatko ne valmistuneet. BPMN tarjoaa mallintajalle tapahtuma (event) -käsitteen ja siihen pohjautuvat notaatiot, joiden avulla voitiin tarkemmin mallintaa palvelupyynnöiden valmistumisesta saatavien ilmoitusten vastaanotto.

Itse hoitoprosessia täydennettiin siten, että laadittiin osaprosessi, joka sisältää hoidon suunnittelun, hoidon toteutuksen ja hoidon arvioinnin ja näille rinnakkaisena osaprosessina "katso palvelun tulos" -tehtävä. Osaprosessi on ympäröity katkoviivoilla, joka merkitsee, että osaprosessin käynnistyminen ei keskeytä pääprosessin suoritusta. Osaprosessin käynnistyminen katkoviivoitetusta ympyrästä, jonka sisällä on avoin kolmio tarkoittaa, että kyseisen tapahtuman vastaanotto käynnistää osaprosessin. Näin on saatu hoitoprosessiin sisällytettyä palvelun tulosten katselu heti sitä koskevan ilmoituksen saapuessa.



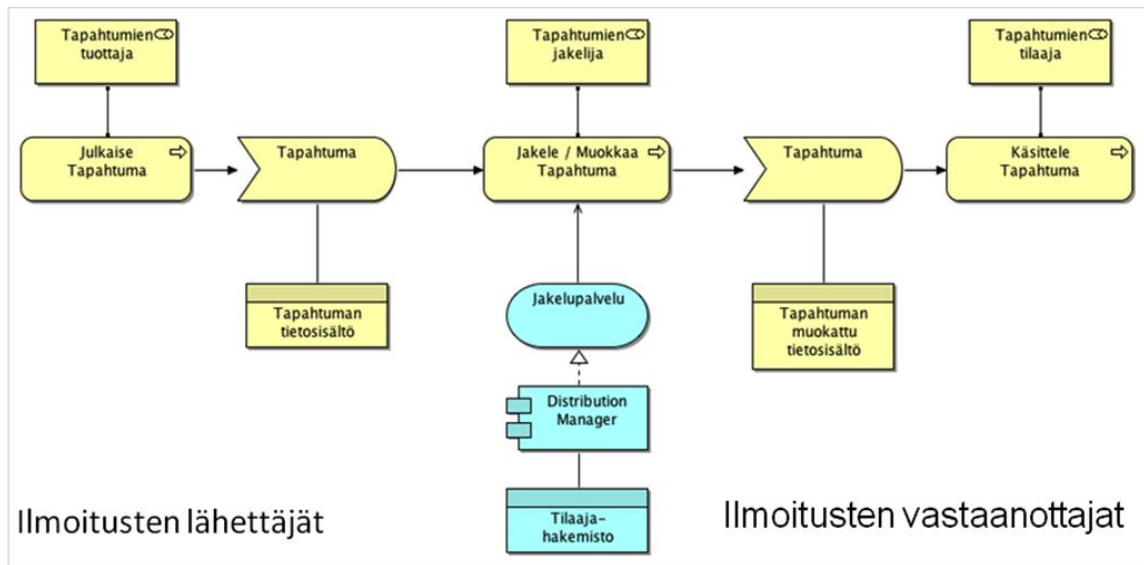
Kuva 16: Palvelun valmistumisilmoitusten vastaanotto

Valmistumista koskevan ilmoituksen lähteminen on tarpeen sijoittaa palveluprosessiin. Tämä saatiin kuvattua lisäämällä palveluprosessiin Palvelu-tehtävän jälkeen katkoviivoilla ympäröity suljettu kolmio. BPMN-notaatiossa tämä tarkoittaa "throwing event" tilannetta, jolloin prosessin suoritus aikaansaa event-ilmoituksen ja jatkuu seuraavaan tehtävään.

Potilaalle tilattujen palveluiden valmistumista seuraava huoneentaulu mallinnettiin omana osaprosessinaan, erillään hoitoprosessista tai palveluprosessista. Huoneentauluprosessilla on kolme vaihtoehtoista käynnistävää tapahtumaa, Tutkimukset valmistuneet, Kuvantamisen lausunto valmistunut tai Konsultointi valmistunut. Prosessin tehtävä on Huoneentaulun päivitys, jonka jälkeen kyseisen prosessin suoritus päättyy.

Analyysi- ja mallinnustyön aikana huomattiin, että palvelun valmistumisilmoitus voisi olla hyödyllinen myös tapahtumätiedon keruussa tietovarastoinnin ja raportoinnin tarpeisiin. Keruuprosessi mallinnettiin vastaavalla tavalla käynnistymään tapahtuman sattuessa.

Samoin havaittiin, että yleistettynä kyseessä on tapahtumapohjainen arkkitehtuuri, Event Driven Architecture, jota on mahdollista hyödyntää myös palvelupohjaisen arkkitehtuurin yhteydessä. Vuorovaikutus perustuu Publish - Subscribe malliin, erotuksena palveluarkkitehtuurin tyypillisestä Request - Reply mallista. Ajatuksena on, että tapahtumia tuottavat osapuolet (Publisher) julkaisevat tapahtumansa tietämättä lainkaan, ketkä osapuolet (Subscriber) ovat kiinnostuneita niitä vastaanotamaan ja tilanneet niitä. Väliin tarvitaan kolmas osapuoli, tapahtumien jakelija, joka vastaanottaa tapahtumat julkaisijoilta ja välittää ne tilaajille. Jakelijalla on oltava tilaajaluettelo, joka sisältää tilaajakohtaisen tiedon siitä, mistä tapahtumista kyseinen tilaaja on kiinnostunut. Publish-Subscribe palveluarkkitehtuuri antaa mahdollisuudet uusien tilaajien lisäämiseen ilman, että tapahtumia julkaisevien osapuolten on tarpeen tehdä mitään. Oheinen Archimate-kuva esittää Publish-Subscribe -mallia.

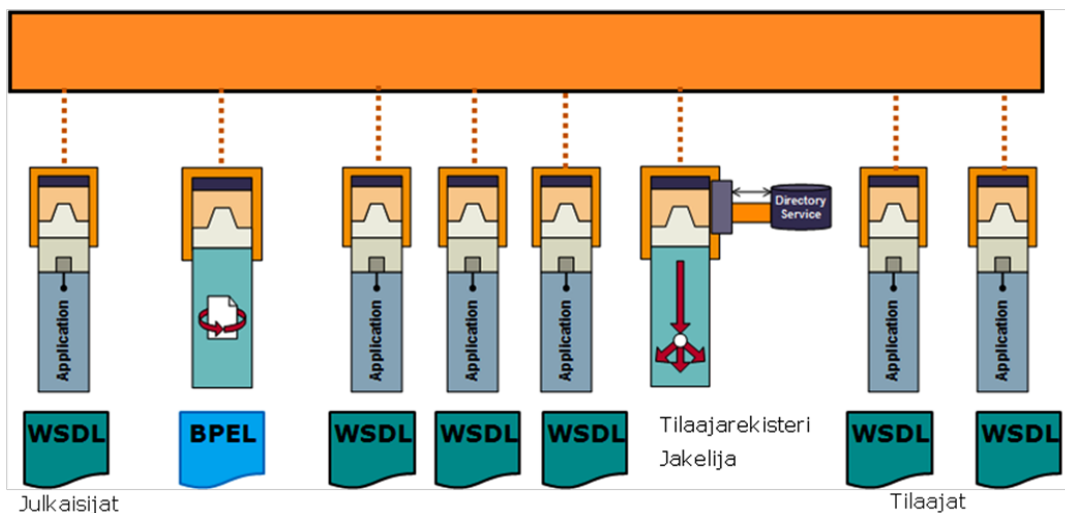


Publish-Subscribe -malli

Kuva 17: Publish - Subscribe malli

Satakunnan päivystyksen tapauksessa tapahtumia tuottavat osapuolet voivat olla palveluita tuottavia yksiköitä, kuten laboratorio, kuvantamisyksikkö ja myös muut osastot, joiden erikoislääkäreitä käytetään konsultaatioihin vaativissa tapauksissa.

Käytännön toteutuksena Satakunnassa hahmoteltiin palveluväylän käyttämistä oheisen kuvan mukaisesti.



Kuva 18: Palveluväylän käyttö Publish-Subscribe mallin toteutuksessa.

Eri tietojärjestelmät voidaan liittää palveluväylään järjestelmäkohtaisilla adaptoreilla. Niiden tehtävänä on sovittaa eri järjestelmät näkymään Web Service -palveluina, jotka voivat olla keskenään vuorovaikutuksessa palvelurajapintojen välityksellä. Yksi palveluväylän toiminto voisi olla tapah-

tumien jakelija, jolla on tehtäväänsä varten on käytössään tilaajarekisteri. Palveluväylään voi liittyä prosessimoottori, jonka tehtävänä on suorittaa BPEL-kielellä kuvattua prosessia.

Palveluväylään liitettyjä järjestelmiä voisivat olla potilashallinnon järjestelmä (tekee palvelupyynnöt), potilaskertomusjärjestelmä (vastaanottaa palveluiden tulokset) sekä eri palveluihin liittyvät järjestelmät kuten laboratorion ja kuvantamisen järjestelmät. Palveluväylään voisivat liittyä myös edellä mainitut huoneentaulua ohjaava järjestelmä sekä tapahtumien seurantatietoa keräävä tietovarastoinnin keruujärjestelmä.

Omana haasteenaan tulee olemaan tapahtumien jakelijan reititystietojen määrittely ja ylläpito. Tätä varten tarvitaan hakemisto, jossa on tarpeen yksilöidä eri osapuolet ja vastaavat palveluosoitteet.

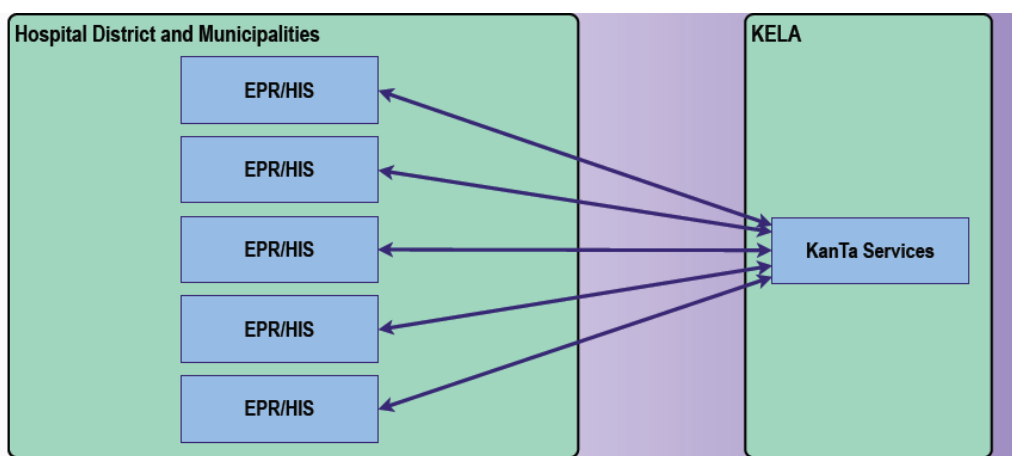
5.3 KanTa-liityntäpisteen toteuttaminen

Kansallinen Terveysarkisto (KanTa) on yhteinen nimitys terveydenhuollon, apteekkien ja kansalaisten valtakunnallisille terveydenhuollon tietojärjestelmäpalveluille. Palvelut tulevat käyttöön vaiheittain koko Suomessa ja niiden toimeenpanosta vastaavat Kela ja THL. Tässä luvussa käsitellään esimerkkinä integraatoratkaisujen määrittelystä KanTa-palveluihin kuuluvan eReseptin liittämistä erityisesti paikallisen tai alueellisen liityntäpisteen määrittelyn ja toteutuksen näkökulmasta.

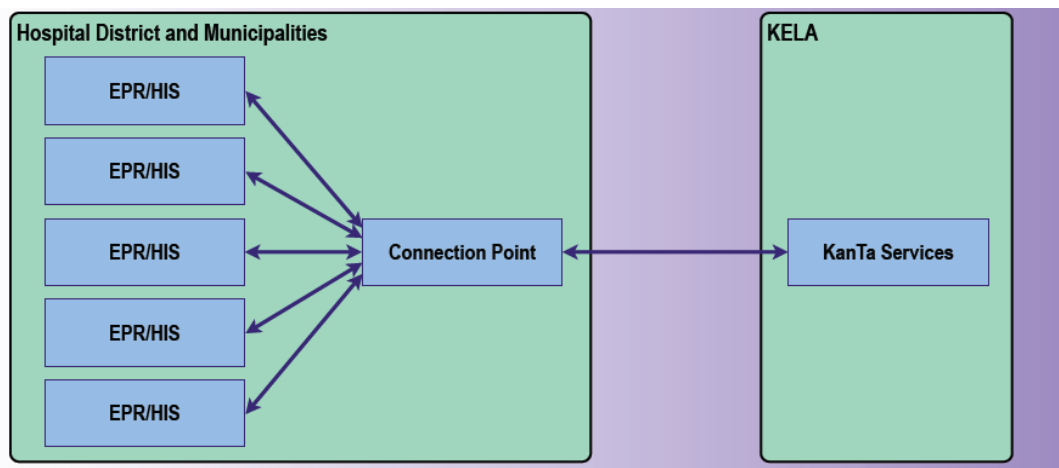
Tietojen ylläpidon ja tietoliikenneyhteyksien valvonnan kannalta on suositeltu, että KanTa-liittyjä keskittää KanTa-palveluihin tapahtuvan tietoliikenteensä yhteen liityntäpisteeseen tai enintään kahteen, jos se tarjoaa vastaanottopalveluja (Kela 2011, KuntaIT 2010).

KanTa-liityntäpisteellä tarkoitetaan pistettä, josta organisaation tietojärjestelmä liittyy KanTa-palveluihin terveydenhuollon varmentajan palvelinvarmenteella salattua ja tunnistettua tietoliikenneyhteyttä pitkin. KanTa-liityntäpisteeseen on asennettu VRK:n tai Valviran myöntämä palvelinvarmenne (Kela 2011). Erilaisia toteutustapoja liityntäpisteelle ovat esimerkiksi kuntatoimijan oma palveluväylä, erillisen välittäjätahon toteuttama liityntäpalvelu, järjestelmätoimittajan sovelluskohdainen integraatoratkaisu, tai yhdistelmät näistä (KuntaIT 2010).

Kuvissa 19 ja 20 on esitetty Kanta-palvelujen käyttöä ilman liityntäpistettä ja liityntäpisteen kautta.



Kuva 19: Kanta-palvelujen käyttö ilman liityntäpistettä (Kauppi 2010)



Kuva 20: Kanta-palvelujen käyttö liityntäpisteen kautta (Kauppi 2010)

Liityntäpisteen toteuttamiseen ja testaamiseen liittyviä hankkeita/projekteja

SOLEA-hankkeen integraatioon liittyvissä työpajoissa käsiteltiin Ensemble Kanta-palveluiden liityntäpisteenä Pohjois-Savon sairaanhoitopiirissä -testausprojektia (Korhonen 2010). Projektin taustalla olivat KuntaIT:n eReseptin integraatioarkkitehtuurin perusteet -projektin tulokset keväältä 2010 (KuntaIT 2010), PSSHP:n kokemukset keskitetyn viestinvälityksen ja integraatioalustan käytöstä sekä muu alueellinen yhteistyö.

Tavoitteena oli toteuttaa KYS:in erityisvastuualueen sairaanhoitopiirien ja muiden sidosryhmien kanssa yhteistyöprojekti, jossa selvitettäisiin jo käytössä olevan palveluväylän / integraatioalustan soveltuvuus keskitetyksi liityntäpisteeksi Kelan Reseptikeskukseen. Liityntäpisteestä tehtiin esimerkkitoteutus käytössä olevan InterSystems Ensemble -alustan avulla. Testauksessa ja dokumentaatioissa oli kuitenkin tavoitteena pyrkiä tuoteriippumattomuuteen siltä osin kuin se on mahdollista.

Projekti suunniteltiin toteutettavaksi siten, että Pohjois-Savossa testataan keskitettyä liittymistä ja Ensemble-toteutusta eReseptin osalta, vaikka erityisvastuualueella Itä-Savo olikin eReseptin ensimmäinen käyttöönottaja. Jos tulokset olisivat positiivisia, testaus jatkuisi Kuopion kaupungin Karkisto2-projektissa (keskitetyn potilastietoarkiston käyttöönotossa) ja ko. pilotti toteutettaisiin keskitetyllä liityntämallilla.

Projektin ensimmäisessä vaiheessa oli tavoitteena toteuttaa Ensembleen tarpeelliset rajapinnat sairaanhoitopiirin alueen potilaskertomusjärjestelmiin päin, joilla se voi välittää alueelta vastaanottamansa sanomat Reseptikeskukseen. Lisäksi toteutus suunniteltiin verifioitavaksi ja testattavaksi Kelan testiympäristön kanssa sekä tarvittava dokumentaatio ja toimintamallien kuvaus tuotettavaksi. Lähtökohtana oli, että perusjärjestelmät näkevät Ensemblen kuten näkisivät Reseptikeskuksen. Perusjärjestelmätoimittajien ei tällöin tarvitse toteuttaa erillisiä rajapintoja keskitettyä liityntäpistettä varten, ellei tätä projektin aikana erikseen toimittajien kanssa nähdä järkeväksi ja kustannustehokkaaksi.

Projektin toisen vaiheen tavoitteena oli PSSHP:n kokemusten perusteella testauksen (ja käyttöönottomallien arvioinnin) laajentaminen muiden toimijoiden käyttöön sekä Kuopion sosiaali- ja terveyskeskuksen eArkisto-käyttöönottoprojektissa tapahtuva testaus Ensemble liityntäpisteenä.

Liityntäpisteen käytännön toteutusta testattiin kahden eri potilastietojärjestelmän kautta kahdessa kunnassa vuodenvaihteessa 2010-2011. Samalla kokeiltiin myös eArkiston palvelukutsuja, koodistokyselyjä kansallisesta koodistopalvelimesta sekä ammattioikeuskyselyjä Valviran attribuuttipalve-

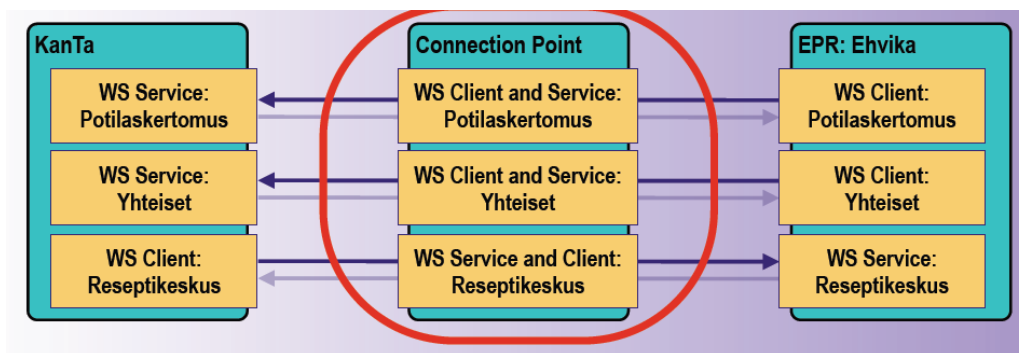
lusta. Testeissä ei ilmennyt erityisiä ongelmia (Kauppi 2011). Liityntäpisteen tuotannollinen käyttöönotto lähti liikkeelle Itä-Savon sairaanhoitopiirissä vuonna 2011, ja useita muita sairaanhoitopiirikohtaisia käyttöönottoja on aikataulutettu vuosille 2011-2012.

eResepti-liityntäpisteen esimerkkitoetus

KanTa tarjoaa kolme Web Services -pohjaista rajapintaa potilastietojärjestelmille. Nämä rajapinnat on kuvattu WSDL-dokumentteina, jotka kuvaavat reseptikeskuksen ja potilastietojärjestelmien rajapintoja. Potilastietojärjestelmien (EPR) rajapinnat on mahdollista toteuttaa käyttämällä näitä samoja WSDL-dokumentteja. Potilastietojärjestelmäesimerkkinä kuvissa 21-22 toimii ”Ehvika”.



Kuva 21: Potilaskertomus- ja Yhteiset- rajapinnat tarjoavat palvelumetodeja, joita potilastietojärjestelmä voi kutsua, esim. uuden reseptin lisääminen arkistoon. Reseptikeskus-rajapinta tarjoaa rajapinnan, jota KanTa voi kutsua (reseptin uusimispyyntö) (Kauppi 2010).



Kuva 22: Liityntäpisteen toteuttamat Web Service-rajapinnat (Kauppi 2010).

WSDL-dokumenteista on mahdollista luoda automaattisesti oliopohjaisia luokkia liittyvien järjestelmien Web Service ja Web Service client -toteutuksia varten. Toteutuksen yksinkertaistamiseksi nämä oliopohjaiset rajapinnat muutettiin kuitenkin XML-rajapinnoiksi.

Kehitetty tekninen perusratkaisu on saatettu vapaaseen ja vastikkeettomaan käyttöön (Kauppi 2011). Edellytyksenä on kuitenkin, että hyödyntäjillä on tarvittava tekninen alusta käytettävissään. Liityntäpalvelun kehittämistä tehdään yhteisen toteutuksen kautta, ja eri osapuolet voivat muuttaa yhteistä toteutusta omilla projekteillaan. Paikallisista muutoksista esiin nousevien kehitystarpeiden kohdalla käydään keskustelu, otetaanko ne mukaan myös yhteiseen toteutukseen.

eReseptin liityntäpisteen integraatoratkaisut ja niiden lähteet

Monet muut eReseptin integroinnin ratkaisut perustuvat joko valtakunnallisiin määrittelyihin, tuotekohtaisiin käytäntöihin tai paikalliseen sopimiseen. Liityntäpiste-projektissa ja topologian valtakunnallisten vaihtoehtojen suosituksissa (mm. KuntaIT 2010) on hyödynnetty luvun 3.1 kuvan 6 mukaista tähtimallia, jossa paikallisesta näkökulmasta integrointialustan kautta muodostetaan hub and spoke-arkkitehtuuri. Palveluväylän (ks. luku 3.3) tarjoamista mahdollisuuksista ei kuitenkaan käytetä esimerkiksi muunnoksia, vaan liittyjät noudattavat valtakunnallisia teknisiä protokollia.

Yllä kuvatuissa eReseptin integraatioiden toteutuksissa tehtyjä integraatoratkaisuja voidaan tarkastella käyttäen luvussa 2.3 esitettyä mallia. Taulukossa 6 esitetään eReseptin integraatioissa eri integrointitasoilla tehtyjä ratkaisuja sekä käytetyissä ratkaisuisa käytettyjä lähteitä sekä se, onko ratkaisu valtakunnallisesti yhteinen, usean käyttäjäorganisaation yhteinen vai organisaatiokohtainen. Taulukossa eivät ole mukana monet apteekkien ja Kelan kannalta olennaiset seikat, vaan siinä keskitytään eReseptiin liittyvien terveydenhuollon organisaatioiden näkökulmaan. Taulukko ei sisällä kaikkia liittymisen yksityiskohtaisia sopimisen aiheita. Monia muista mainituista valtakunnallisista projekteista on erikseen käsitelty ja tutkittu mm. sen suhteen, kuinka erilaisten kokonaisarkkitehtuurikuvausten kohteet muuttuvat projektien välillä (Mykkänen, Virkanen 2012). Taulukosta näkyy selvästi, kuinka yhteentoimivuuksratkaisuisa ja niitä kehittämissä projekteissa on huomioitava runsaasti muitakin seikkoja kuin tyypillisesti standardoinnissa huomioitavat rajapinta- ja semanttiset määrittelyt. Samoin näkyy laajoissa kehittämishankkeissa tyypillinen laaja toimijakenttä ja huomioitavien määrittelyjen runsas määrä, vaikka taulukko ei kata läheskään kaikkia huomioitavia seikkoja.

Taulukko 6. eReseptin integraatoratkaisujen määrittelyjä.

Integrointitaso	Ratkaisun kohde	Määrittelykset / projektit	Määrittysten kohdealueen laajuus
Prosessit	eReseptin prosessit (lääkkeen määrääminen, toimitus, uusiminen, korjaaminen, varaukset, annosjakelu, lukitukset jne.)	Kela määrittelyt ja käyttötapaukset	valtakunnallinen, myös lainsäädäntöpohja
Kehitysprosessin liittymät	menettelyt kuinka eResepti liitetään valmiisiin potilastietojärjestelmiin	tuote- ja käyttäjäorganisaatiokohtaiset dokumentit, ohjeistusta ja vaatimuksia KuntaIT ja Kela määrittelyistä	paikallinen, osin usean organisaation yhteinen
	liittymisratkaisujen palvelutaso	Kela määrittelykset (valtakunnallinen), KuntaIT määrittelykset (paikallinen), paikalliset projektit ja liittymissopimukset (paikallinen)	valtakunnallinen, paikallinen
Toiminnallinen viitemalli	käytössä olevien järjestelmien tukemat tietosisällöt	tuotekohtaiset määrittelyt	tuotekohtainen / paikallinen
	käytettävien koodistojen saanti järjestelmiin	tuotekehitysprojekti tai paikallinen käytäntö, ratkaisu vaikuttaa useisiin alempiin tasoihin	usean organisaation yhteinen / paikallinen
Semantiikka	eReseptin sanomien tietokenttien merkitykset	Kela- ja HL7-määrittelyt	valtakunnallinen
	käytettävät koodistot	Kela- ja HL7-määrittelyt, THL koodistopalvelu	valtakunnallinen
Toiminnalliset rajapinnat	eReseptin integraatiossa käytetyt sanomatyyppit ja sanomat	Kela- ja HL7-määrittelyt	valtakunnallinen, pohjautuu valtakunnallisiin määrittelyisiin ja kv-standardeihin
Sovellusinfrastruktuuri	liityntäarkkitehtuurin topologia vaihtoehdot	KuntaIT- ja Kela-määrittelyt	valtakunnallinen
	liityntäpisteen topologian valinta ja tarkennus	liityntäpiste-projekti	usean organisaation yhteinen / paikallinen
	eReseptin integrointipisteet käytettävässä EPR:ssä	tuotekehitysprojekti	usean organisaation yhteinen
Tekninen infrastruktuuri	käytettävä integrointialusta / paikallinen ESB ja sen konfigurointi	liityntäpiste-projekti, vaatimuksia mm. KuntaIT-ohjeistuksesta	usean organisaation yhteinen, paikallinen
	turvallisuussertifikaation hankkiminen ja käyttöönotto	tuote- ja käyttäjäorganisaatiokohtaiset dokumentit, sopiminen Kela ja Valvira kanssa	paikallinen, perustuen valtakunnallisiin ohjeisiin
	liittyjien osoitehakemisto	Kela ohjeistus ja toteutus, huomioitava liittymisprojekteissa	valtakunnallinen ja paikallinen
	porttien, IP-osoitteiden ja URL:ien sopiminen	käyttäjänorganisaatiokohtaiset dokumentit	paikallinen
Tekniset liittymät	sanomaliikenteen rajapintatekniikat	HL7-määrittelyt	valtakunnallinen, pohj. kv. standardiin
	liikenteen salaus-, allekirjoitus- ja tunnistustekniikat	Kela- ja HL7-määrittelyt	valtakunnallinen, pohj. kv. standardeihin
Verkot ja laitteet, palomuurit	laite- ja verkkoympäristön hankinta ja konfigurointi	käyttäjänorganisaatiokohtaiset dokumentit, KuntaIT ohjeistusta	paikallinen

5.4 HL7 Service-Aware Interoperability Framework (SAIF)

HL7-organisaatio on kehittänyt vuodesta 2008 lähtien HL7-standardien tuottamiseen uutta kokonaisarkkitehtuuri- ja palvelukeskeistä menettelytapaa. Kehikon nimenä on tällä hetkellä SAIF (Service-Aware Interoperability Framework) (HL7 2011). Tavoitteena on ollut tuottaa "kokonaisarkkitehtuurilähestymistapa" HL7-määrittelyjen ja standardien kehittämiseen. Määrittelyssä on pyritty tuottamaan joukko malleja ("kieliä"), joilla standardien ja yhteentoimivuuksratkaisujen kehittämistä voidaan hallita ja yhdenmukaistaa. Keskeinen taustamalli SAIF:n kehittämiseksi on ollut ISO:n RM-ODP-malli ja sen viisi näkökulmaa (ks. mm. Itälä ym. 2012). Erityisesti tavoitteena on ollut pystyä tukemaan organisaatioiden välisen semanttisen yhteentoimivuuden toteuttamista ja yhdenmukaiseen kehikkoon perustuvien standardien soveltamisoppaiden (implementation guide, IG) määrittelyä eri organisaatioissa.

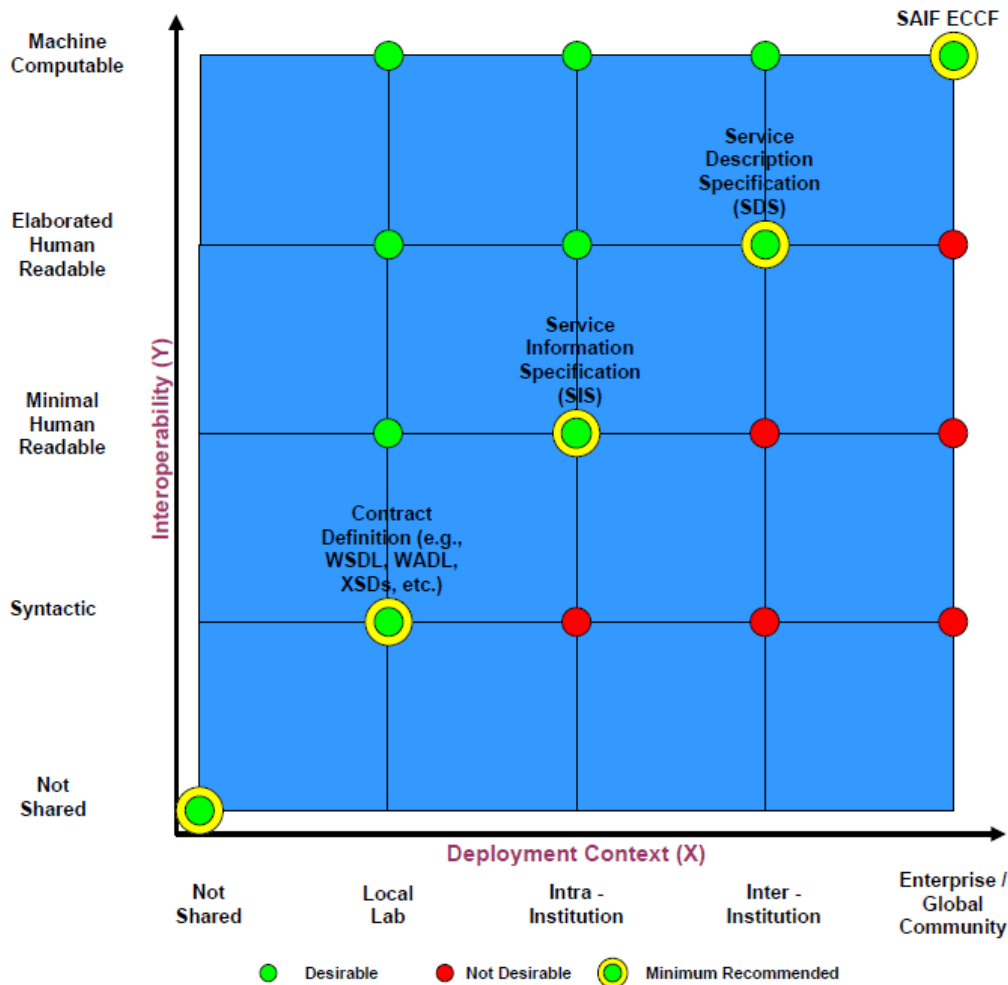
SAIF-kehikon neljä pääosaa ovat:

- Hallinnointikehys (Governance Framework, GF): määrittelee kielen ja rakenteet, joiden avulla organisaatioissa voidaan ilmaista yhteisiä tavoitteita ja hallintasuhteita erityisesti suhteessa erilaisiin yhteentoimivuuden tavoitetasoihin. Hallinnointikehys perustuu Thomas Erl'in malliin, jossa hallintamalli koostetaan neljästä elementistä: ohjeista (precepts, sisältää tavoitteet, toimintasäännöt, standardit ja suositukset), ihmisistä (ja heidän rooleistaan), prosesseista ja mittareista.
- Toiminnallinen kehys (Behavioral Framework, BF): määrittelee kielen sille kuinka ilmaistaan ratkaisuihin mukana olevien tahojen tai osien käyttäytymistä yhteentoimivuuteen liittyen. Keskeisiä määriteltäviä seikkoja ovat sopimukset, operaatiot ja prosessit. Aiemmissa HL7-viestistandardointimenetelmissä nimenomaan käyttäytymisen ja vuorovaikutuksen määrittely on nähty haasteellisena.
- Tietokehys (Information Framework, IF): määrittelee kielen sille kuinka kohdealueen staattinen tieto (information) tunnistetaan, esitetään ja tarkennetaan toteutettavaksi. Tietokehysten pääosat ovat käsitteet, tietotyypit, luokat ja tietomallit. Käsitteiden määrittelyyn voidaan käyttää terminologioita ja koodistoja tai vapaata esitysmuotoa. Viitetietomalli kuten HL7 RIM-malli (ks. dokumentti "Kokonaisarkkitehtuurin ja palveluarkkitehtuurin menetelmät ja välineet") toimii pohjana käsitteiden ja tietojen esittämiselle. Tietokehysten mallit voivat olla myös validoitavia ja suoritettavia.
- Yhdenmukaisuuskehys (Enterprise Consistence / Conformity Framework, ECCF): "määrittelee kielen sille kuinka yhteentoimivuuksmäärittelymatriisin solujen välisten suhteiden merkitykset määritellään" eli kuinka samaan tarkoitukseen liittyvät yhteentoimivuuksmäärittelyt liitetään toisiinsa. Yhdenmukaisuuden tarkistuspisteitä voidaan määrittellä joko havainnoitavaksi tai ohjelmallisesti testattavaksi.

SAIF määrittelee suuren joukon käsitteitä ja käsitelmalleja, ja lisäksi joukon käyttökelpoisia jäsenymalleja yhteentoimivuuden ja siihen liittyvien määritysten, sopimusten ja vastuiden tarkentamiseen. Alla olevassa kuvassa esitetään kehikon yhteentoimivuuksastot, käyttöönottotasot sekä suositeltu "hallinnan taso" näiden leikkauspisteissä.

SAIF-mallissa yhteentoimivuuksmatriisi ja ECCF pyrkivät tarjoamaan keinot määritysten dokumentointiin ja hallintaan. Yhteentoimivuuksmatriisi (Interoperability Specification Matrix, ISM) kokoaa kuvaukset, joilla määritellään tietyn toimijan tai komponentin tiedonvaihto- ja vuorovaikutusominaisuudet. Yhteentoimivuuden soveltamisoppaita ja määrittelyksiä tuotetaan kokoamalla ISM:n eri

osiin kuuluvia kuvauksia ja niihin kuuluvia määrittysten mukaisuuden määrittelyjä (conformance statements) yhteen. Yhteentoimivuusmatriisissa hyödynnetään neljää RM-ODP-kehikon näkökulmaa (Enterprise, Information, Computational ja Engineering) sekä OMG:n MDA-lähestymistapaan perustuvia abstraktiotasoa (Conceptual, Platform-independent, Platform-specific).



Kuva 23. HL7 SAIF-kehikon yhteentoimivuustasot (y-akseli), käyttöönotto- ja tarvittava hallinnoinnin taso (HL7 2011).

Yhteentoimivuusmatriisi (ISM) määrittelee SAIF-kehikon mukaisuuskriteerit soveltamisoppaiden malleille (Interoperability Specification Template, IST). IST on profiili, joka määrittää yksittäisissä soveltamisoppaissa ja joka rajoittaa ISM:n osoittamaa aluetta. IST puolestaan toimii pohjana ja määrittelee mukaisuuskriteerit yhteentoimivuuden toteutusmäärittelylle (Interoperability Specification Instance, ISI). ISI on kuvausten kokoelma, jossa määritellään tietty yhteentoimivuusratkaisun osa kuten sovelluspalvelu-, sanoma- tai asiakirjamäärittely. Sovelluksiin tehtävät toteutukset puolestaan voidaan todeta yhteentoimivuuden toteutusmäärittelyjen mukaisiksi.

SAIF-mallin tavoitteena on tukea kolmea erityyppistä yhteentoimivuuden toteutusmallia (interoperability paradigm) siten, että samoja rakenteita ja sisältöjä voitaisiin hyödyntää erityyppisissä integraatiomäärittelyissä:

<i>Topic Specification</i>	Enterprise / Business Viewpoint	Information Viewpoint	Computational Viewpoint	Engineering Viewpoint
<i>Conceptual</i>	Business Context, Reference Context	Concepts from Domain Analysis (Information) Model	Collaboration Analysis, Functional Profile(s), Service Roles and Relationships	Existing Platform capabilities, essential requirements regarding compatibility
<i>Platform Independent (logical)</i>	Business Governance	Project-oriented Domain Information Model, Constrained Information Model, Localized Information Model, Hierarchical Message Definition	Collaboration Types, Interface Specification and Functional Groups, Interaction Types and Collaboration Participations, Contracts Parts	Existing Platform models, libraries, intended transparencies of the platform.
<i>Platform Specific (implementable)</i>	Rules, Procedures	Localized Information Model, Schema	Collaboration scripts, Orchestrations, Realized Interfaces	Execution Context (channels, stubs, binders, protocols, and interceptors), Transforms, Deployment Model

Kuva 24. Esimerkki populoidusta SAIF-mallin yhteentoimivuusmatriisista: Interoperability Specification Template-esimerkki (HL7 2011).

- sanomapohjainen integraatio kuten HL7 versio 2- ja 3-sanomanvälitys pohjautuen tiedon- siirrossa käytettävien sanomien määrittelymenetelmiin,
- palvelupohjainen integraatio, kuten SoaML- tai OASIS SOA Reference Model-pohjaisesti mallinnetut palvelut joita voidaan toteuttaa Web services- tai REST-tekniikoilla,
- dokumenttipohjainen integraatio, jossa hyödynnetään pysyviä ja määriteltyjä dokumentti- muotoja kuten HL7 CDA (Clinical Document Architecture) standardi.

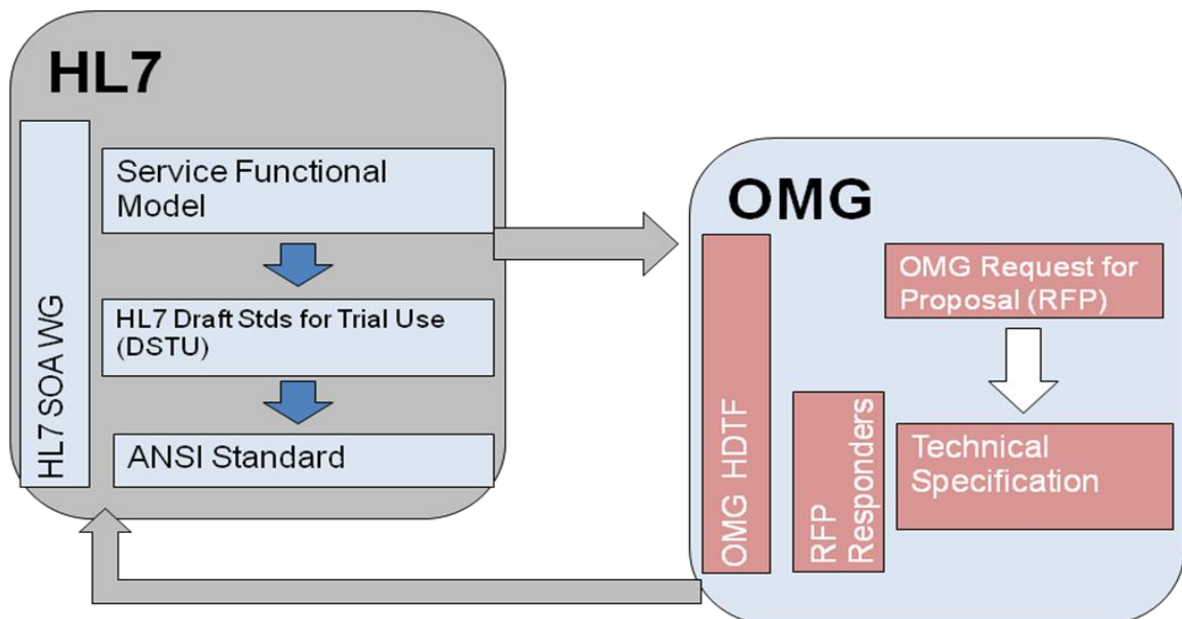
Eri toteutusmalleissa on eroja mm. toiminnallisen kehyksen osalta tehtävissä valinnoissa, mutta esimerkiksi yhteisten viitetietomallien käyttöä pyritään edistämään eri toteutusmallien välillä.

HL7-organisaatiossa useat aktiiviset työkohteet vuodesta 2009 lähtien ovat olleet SAIF-menetelmän mukaisesti kehitettäviä (nk. alpha projects). Varsinaisen SAIF-kehikon dokumentaatioissa (HL7 2011) käytetään suhteellisen monimutkaista käsitteistöä ja kieltä ja pitkiä sekä runsaasti käsitteitä sisältäviä lauseita, mikä hankaloittaa keskeisten seikkojen löytämistä ja ohjeistuksen hyödyntämistä. Lisäksi lukuisien esitettyjen käsittemallien hyödyntäminen jää helposti lukijalle epäselväksi. Kehikon hyödyntämiseen onkin suunniteltu tarkempien soveltamisoppaiden tuottamista. Kehikko ei vielä ole levinnyt laajasti HL7-organisaation lukuisten eri määrittelykomiteoiden käyttöön, mutta esimerkiksi rokatustietojen kansainvälisten soveltamisoppaiden kehittämisessä yhdistellään eri toteutusmalleihin pohjautuvia valmiita määrittelyjä kehikkoa soveltaen eri standardointijärjestöjen yhteistyössä.

5.5 Healthcare Services Specification Project (HSSP)

Healthcare Services Specification Project (HSSP) on HL7 (Health Level Seven) ja OMG (Object Management Group) -standardointikonsortioiden yhteisprojekti, joka pyrkii määrittelemään terveydenhuollon palvelukeskeisiä yhteentoimivuusstandardeja. HSSP käynnistettiin vuonna 2005, kun useilta tahoilta (mm. Kaiser Permanente ja veteraaliasiain ministeriö VA USA:ssa, Suomessa toteutettu PlugIT-projekti) tuotiin esiin terveydenhuollon palvelukeskeisten integraatoratkaisujen standardoinnin tarpeellisuus. Projekti toimii HL7-organisaation SOA technical committee- ja OMG-organisaation Healthcare Domain Task Force-ryhmien kautta. Sillä on lisäksi läheiset yhteistyösuhteet IHE-organisaatioon sekä Open Health Tools-yhteisöön, jotka tuottavat standardien soveltamisoppaita sekä avoimen lähdekoodin välineitä yhteentoimivuusratkaisujen tuottamiseen.

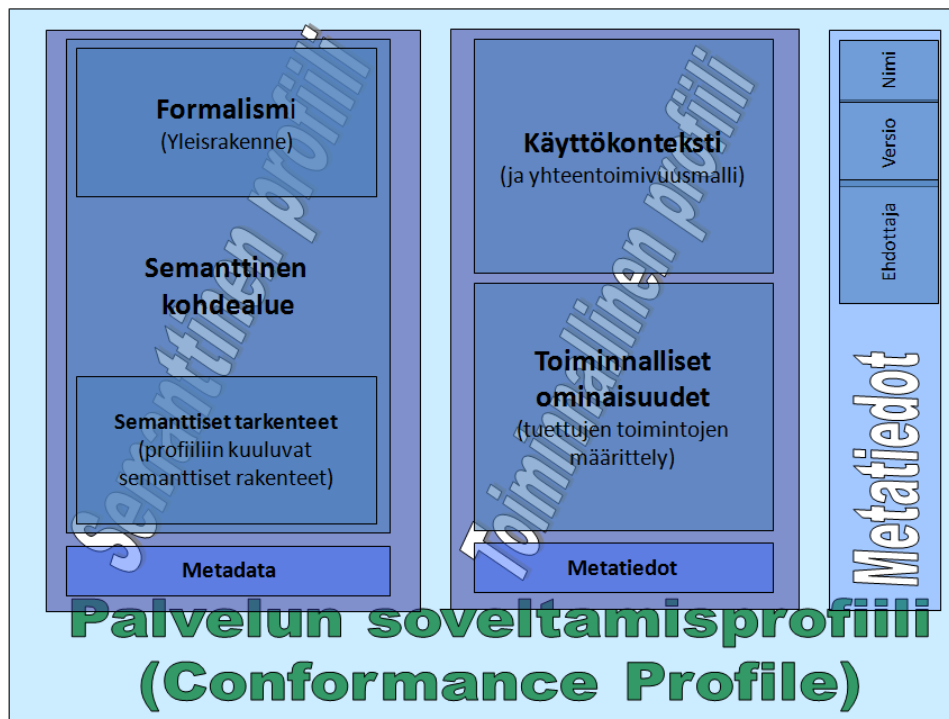
HSSP-projektissa on määritelty prosessi avointen palvelurajapintamäärittelyjen tuottamiseen ja hyväksymiseen. Määrittelyjen perusdokumenteja ovat toiminnalliset palvelumäärittelyt (Service Functional Model Specification), joissa määritellään palvelujen käyttötapaukset, vaatimukset sekä toiminnalliset ominaisuudet ja rajapinnat tekniikkariippumattomalla tasolla. Näiden määrittelyjen pohjalta voidaan tuottaa tekniset, toteutus- tai rajapintatekniikkaan sidotut määrittelyt joihin varsinaiset palvelujen toteutukset pohjautuvat. Malli seuraa OMG:n MDA-lähestymistapaa ja tämän dokumentin kuvassa 2 esitetyn määrittelyprosessin tekniikkariippumattomien ja teknisten rajapintojen määrittelytapaa. Standardointijärjestöjen yhteistyömalli muodostuu siten, että palvelujen toiminnalliset mallit (SFM) tuotetaan HL7-organisaatioissa huomioiden sisällöllisten erityiskomiteoiden tarpeet ja vaatimukset (ks. Kuva 25). Toiminnalliset mallit voidaan hyväksyä HL7-hyväksymisprosessin mukaisesti kokeilustandardeiksi ja lopullisiksi standardeiksi. Ne toimivat myös syöteenä ja alustariippumattomina malleina OMG:n standardointiprosessiin. OMG:n prosessissa lähetetään järjestön jäsenille ehdotuspyyntö (RFP), jossa viitataan HL7:n määrittelemään toiminnalliseen malliin. OMG:n jäsenet tekevät ehdotuspyynnön pohjalta ehdotuksia teknisiksi määrittelyiksi joissa on toiminnallisen määrittelyn mukaiset seikat toteutettuna valitulle alustalle tai rajapintatekniikalle. Nämä ehdotukset harmonisoidaan ja hyväksytään osana OMG:n prosessia, joka myös edellyttää että lopulliset hyväksyttävät standardit on toteutettu tuotteisiin (Kawamoto ym. 2009).



Kuva 25. HL7- ja OMG-standardointijärjestöjen yhteistyö HSSP-projektissa.

HSSP:n prosessin tavoitteena on hyödyntää sekä HL7:n sisällöllistä että OMG:n teknistä osaamista, tuottaa toiminnallisuuden, semantiikan sekä tekniset ratkaisut sisältäviä standardeja sekä pyrkiä hyödyntämään olemassa olevia standardeja (kuten HL7 EHR-S toiminnallisten ominaisuuksien standardit, HL7 RIM-viitetietomalli ja IHE-integraatioprofiilit).

HSSP-määrittelyjä on tehty sekä tyypillisen API-rajapintadokumentaation että yleistety, toiminnalliset ja semanttiset ominaisuudet erottavan lähestymistavan mukaisesti. Kuvassa 26 on jälkimmäisen lähestymistavan mukainen kaavio palvelun soveltamisprofiilin sisältämistä määrittelyistä.



Kuva 26. HSSP:n palvelumäärittelyn soveltamisprofiilin toiminnalliset ja semanttiset osat.

HSSP on tuottanut joukon avoimia määrittelyksiä palvelupohjaista terveydenhuollon sovellusintegraatiota varten. Taulukossa 7 esitetään keskeisimmät valmistuneet tai käynnissä olevat HSSP-määrittelykset.

HSSP-projektissa kehitettyjä palvelujen toiminnallisten määrittelyjen pohjia on suunniteltu yleisesti käytettäväksi luvussa 5.4 esitetyn SAIF-mallin palvelupohjaisissa määrittelyissä. SAIF-kehikossa ei kuitenkaan oleteta, että eri abstraktiotasojen määrittelyt kulkisivat läpi eri standardiorganisaatioiden.

Standardimäärittelyjen lisäksi HSSP-projekti on tuottanut mm. "Practical guide to SOA in Health-care"-dokumentaation, jossa havainnollistetaan esimerkin kautta palvelupohjaista terveydenhuollon organisaatioiden tietojärjestelmäympäristön määrittelyä (Rubin 2009). Taulukossa 7 mainittujen määrittelyjen lisäksi mm. useisiin muihin turvallisuuspalveluissa on tunnistettu vaatimuksia ja palveluja, joihin ei ole vielä tuotettu tarkempia määrittelyjä.

HSSP:ssä määrittelyjen palvelujen toteutuksia löytyy mm. USA:n veteraaniasiain ministeriön (VA) rokotustietorekisterin toteutuksessa (IXS, RLUS, CTS2), Mayo Clinicilta (CTS2), OpenCDS-projektista (DSS), Open Health Tools-välineistä (PASS Audit), Italiasta sekä kansainvälisestä eSOS-hankkeesta. Myös luvussa 5.6 kuvataan eräs toteutus esimerkki, jossa palvelumäärittelyjä yhdistetään vakiintuneita standardeja hyödyntäviin integraatioprofiileihin.

Taulukko 7. HSSP-määrittelyt.

Nimi	Kohdealue	Tila
IXS (Identity Cross-reference Service) aikaisemmin EIS (Entity Identification Service)	Henkilön (tai muun kohteen) tunnistaminen	Hyväksytty normatiivinen HL7 standardi, hyväksytty OMG-standardi 2011, avoimen lähdekoodin välineitä saatavilla
RLUS (Resource Location and Update Service)	Resurssien kuten dokumenttien hakeminen ja ylläpito	Hyväksytty normatiivinen HL7 standardi, hyväksytty OMG-standardi 2011
DSS (Decision Support Service)	Kliininen päätöksentuki ja tietämysmoduulien liittäminen päätöksentukipalveluihin	Hyväksytty HL7 kokeilustandardi, hyväksytty OMG-standardi 2011, avoimen lähdekoodin välineitä saatavilla
CTS II (Common Terminology Services II)	Terminologia- ja koodistorajapinnat	Hyväksytty HL7 kokeilustandardi, etenemässä OMG:n prosessissa yhdistäen kolme toteutusmäärittelyä 2011
PASS (Privacy, Accountability and Security Services) Audit	Turvallisuuslokipalvelut	HL7-äänestyksen kommentit käsittelyssä 2010
PASS Access Control Conceptual Model	Pääsynhallintapalvelujen käsitelmä	Läpäissyt HL7 kokeilustandardi (DSTU) äänestyksen 2010
HCSPDIR (Healthcare and Community Services Provider Directory)	Hyvinvointipalvelujen hakemiston rajapinnat	Hyväksytty normatiivinen HL7 standardi
hData -määrittelyt	terveydenhuollon tietojen yleiskäyttöiset web-tietorakenteet, RLUS-palvelun REST-toteutusrajapinta, sisältöprofiilit eri tietosisältöjä varten	aktiivinen kehitysvaihe

5.6 IHE-profiilien hyödyntäminen palveluarkkitehtuurissa

Tässä luvussa kuvataan esimerkki valmiiden ja vakiintuneiden standardien ja niiden tarkennettujen soveltamisoppaiden hyödyntämisestä SOA-tyyppisesti. Esimerkillä pyritään havainnollistamaan sitä, kuinka olemassa olevia integraatiomäärittelyjä voidaan tuoda SOA-arkkitehtuurin mukaisiin ympäristöihin ja hankkeisiin.

IHE (Integrating the Healthcare Enterprise) on kansainvälinen yhteisö, joka määrittelee standardeihin perustuvia profiileja terveydenhuollon tietojärjestelmien yhteensovittamiseen (Mykkänen & Virkanen 2009). IHE määrittelee kansainvälisiä *integraatioprofiileja* osana teknisiä viitemalleja (technical framework) monilla eri sovellusalueilla sekä järjestää alueellisia (Eurooppa, Pohjois-Amerikka, Aasia-Tyynenmeren alue) testaus- ja esittelytapahtumia. Järjestöllä on useissa maissa toimivia maakohtaisia ryhmiä. Integrintiprofiilit koostuvat *aktoreista* (tietystä roolissa toimivat sovellukset) sekä yhteentoimivuusstandardeihin viittavista *transaktioista*, joissa määritellään toimijoiden välisiä sanomia tai kutsuja. Integrintiprofiileissa pyritään suosimaan vakiintuneita ja jo käytössä olevia standardeja, joiden käyttö rajoitetaan profiilissa määriteltyihin tarkkoihin käyttötapauksiin. Näitä rajattuja profiileja (soveltamisoppaita) käyttäen voidaan nopeuttaa integrintiratkaisujen

toteuttamista siten, että järjestelmien integraatiot pystytään toteuttamaan nopeasti (jota demonstroidaan esimerkiksi testaus- ja esittelytapahtumissa). IHE-integraatioprofiileihin on mahdollista viitata tarjouspyynnöissä ja järjestelmiä hankittaessa ja järjestelmätoimittajat voivat julkistaa *integration statement* -dokumenteja, joissa kuvataan tuotteiden tukemat IHE-profiilit, aktorit ja optiot.

IHE on määritellyt työnkulkuprofiileja, infrastruktuuriprofiileja sekä sisältöprofiileja. Työnkulku-profiilit keskittyvät tietyn toimintokokonaisuuden integraatioihin, esimerkiksi radiologian työnkulku (SWF) perusjärjestelmän, radiologian toiminnanohjausjärjestelmän, kuva-arkiston ja kuvantamislaitteiden välillä. Infrastruktuuriprofiilit sisältävät määrittelyjä keskeisten monissa tilanteissa tarvittavien seikkojen ratkaisuun, kuten XDS-profiilin määrittelyt dokumenttien jakamiseen. Sisältöprofiilit keskittyvät nimensä mukaisesti tietosisältöihin, kuten CDA- ja CCD-dokumenttistandardeja soveltava XPHR-profiili henkilökohtaisten terveystaltioiden tietojen vaihtoon potilaskertomusjärjestelmien kanssa.

IHE-profiileja on pyritty hyödyntämään myös SOA-pohjaisissa hankkeissa eri maissa ja organisaatioissa. Tästä syystä IHE IT Infrastructure -komitea on koostanut näkemyksen siitä, kuinka IHE-integraatioprofiileja voidaan hyödyntää SOA-hankkeissa ja -ympäristöissä (Painter ym. 2009).

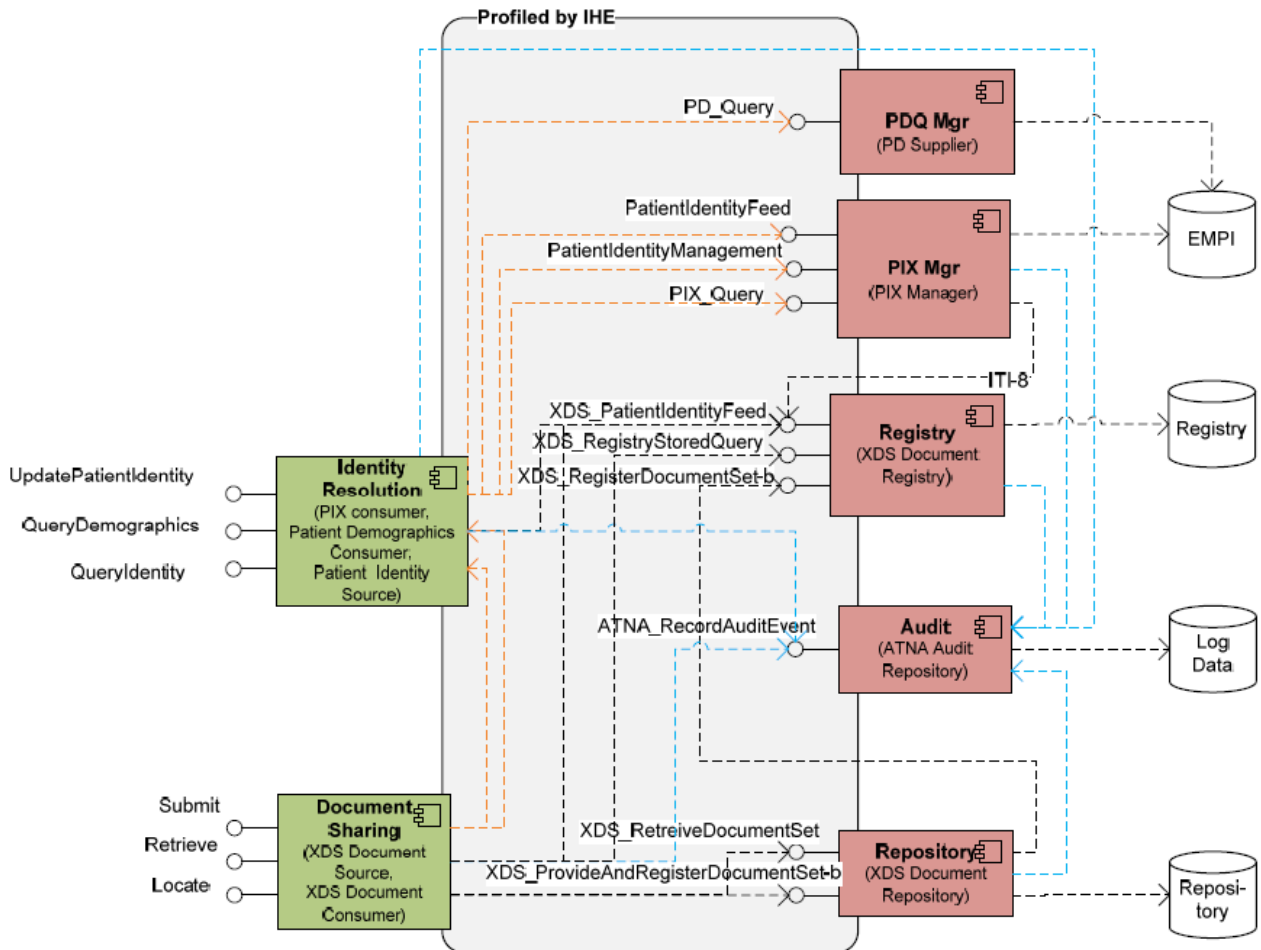
SOA- ja IHE-lähestymistavoissa on kuitenkin myös selviä eroja. SOA-palvelujen määrittelyt foku-soituvat tyypillisesti suhteellisen korkealla abstraktiotasolla määriteltäviin palvelujen toiminnallisuuksiin, joita IHE-mallissa vastaavat profiili-dokumenteissa kuvattava yhteentoimivuuden peruskuvaukset. IHE-määrittelyissä taas korostuu tarkkojen transaktioiden määrittely, joka vastaa SOA-toteutuksissa tehtäviä teknisiä valintoja. IHE pyrkii tiukkaan yhdenmukaisuuteen ajonaikaisten integrointiratkaisujen tasolla, kun taas SOA pyrkii lisäämään joustavuutta siten, että palvelu olisi eri konteksteissa ja usein myös eri rajapintatekniikoiden avulla uudelleenkäytettävissä. IHE-ratkaisujen painotus on integraatiolähtöinen, alhaalta-ylös tyyppinen, kun taas SOA-määrittelyissä korostuu usein palvelujen tunnistaminen prosesseista (ylhäältä-alas). Taulukossa 8 esitetään keskeisten IHE- ja SOA-käsitteiden vastaavuuksia.

Taulukko 8. IHE- ja SOA-käsitteiden vastaavuuksia.

IHE	SOA
Aktori	Palvelun tarjoaja
Aktori	Palvelun kutsuja
Integraatioprofiili vastaa osittain (IHE Technical Framework Volume 1)	Abstrakti palvelurajapinta
Transaktio vastaa osittain (IHE Technical Framework Volume 2)	Tekninen rajapintamäärittely
Transaktio vastaa osittain (IHE Technical Framework Volume 2)	Kutsuparametri, "etäohjelmakutsu"-tyyppinen
IHE-sisältöprofiili (esimerkiksi CDA-dokumentin soveltamisopas)	Kutsuparametri, "dokumentti"-tyyppinen

HSSP:n (ks. luku 5.5) toiminnallisia sovelluspalvelumäärittelyjä on toteutettu IHE-profiilien avulla. Perusmalli pohjautuu kolmitasoiseen palvelumalliin (Painter ym. 2009): käyttötapauksen vaatimuksia ja toiminnallisuuksia toteutetaan task service-kerroksessa. Tämä kerros kutsuu entity service-kerrosta, joka edelleen hyödyntää Utility-kerrosta. Task-tason palvelu voisi tarjota esimerkiksi potilaan lääkityshistorian hakemistoiminnon. Tämä toiminto hyödyntäisi kahta Entity-tason palvelua 1) henkilön tunnistamiseen ja perustietojen hakuun sekä 2) dokumenttien hakuun. Utility-service-kerroksesta taas löytyisi palveluita, jotka vastaavat täysin IHE-profiileissa esitettyjä aktoreita. Näiden aktorien välinen sanomaliikenne taas voisi toteutuksessa nojautua tarkkoihin teknisiin transak-

tioihin. Alla olevassa kuvassa esitetään yllä kuvatun esimerkin mukaisten Entity- (palvelut vasemmalla) ja Utility-tason (palvelut oikealla) palvelujen toteutus IHE-profiilien avulla. Esimerkissä toteutetaan toiminnallisten IXS- ja RLUS-sovelluspalveluiden (ks. luku) toiminnallisuutta IHE:n PIX (potilaan tunnistaminen), PDQ (potilaan perustiedot) ja XDS (asiakirjojen tallennus ja noutaminen) avulla.



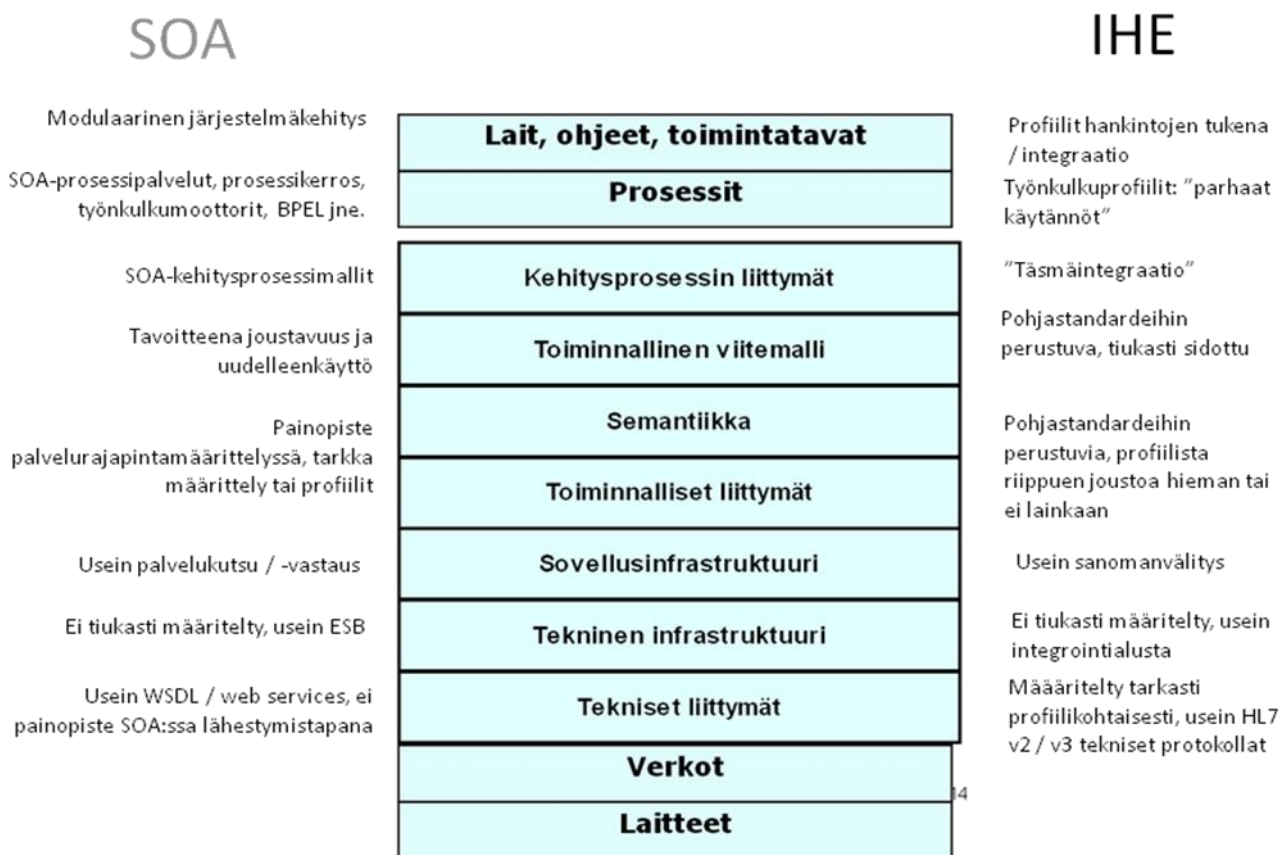
Kuva 27. Esimerkin Entity- (vihreä) ja Utility-tason (punainen) palvelut ja niiden vastaavuus IHE-profiileihin (Painter ym. 2009).

Esitetty kerrostaminen ei ole ainoa mahdollinen. Riippuen eri standardien ja profiilien sisällöistä on mahdollista joissakin tilanteissa muodostaa myös matalampia hierarkioita ja vähemmän abstraktiokerroksia tarkkojen teknisten standardien ja toiminnallisten SOA-palvelumäärittysten väliin. Tässä esitellyllä mallilla on tavoiteltu toiminnallisten palvelumäärittysten ja standardiprofiilien yhdistämisen lisäksi monimutkaisuuden vähentämistä loogisten palvelujen tasolla sekä lisääntynyttä teknistä ketteryttä. Malli antaa myös mahdollisuuden toteuttaa samoja palveluja eri tekniikoilla eri ympäristöissä tai muuttaa tietyn palvelukerroksen toteutus uuteen tekniikkaan vaikuttamatta palvelujen kutsujiin. Malli myös havainnollistaa sitä, millä tavoin valmiita teknisiä rajapintamäärittämiä voidaan sovittaa osaksi loogisella tasolla määriteltävää palvelupohjaista arkkitehtuuria.

Kuten yleisiä standardeja, myös tarkemmin määritellyjä integraatioprofiileja hyödyntämällä ei välttämättä ole helppoa saavuttaa minkään yksittäisen käyttötilanteen kaikkia paikallisia vaatimuksia (Mykkänen, Tuomainen 2012). Yleisten standardien ja määrittysten pohjalta voidaan kuitenkin usein tunnistaa SOA-palveluja jotka täyttävät standardin tai profiilin mukaisesti osan vaatimuksista, ja

SOA-periaatteen mukaisesti tuottaa muita palveluja ja sisältö- tai käyttöoikeusmäärittelyjä, joilla muut vaatimukset saadaan täytettyä. Tämän tyyppistä lähestymistapaa on sovellettu mm. IHE:n BPPC-suostumusprofiiliin toteuttamisessa palvelupohjaisesti avoimen lähdekoodin välineitä hyödyntäen osaksi henkilökohtaista terveystietoa (Heinze ym. 2011).

Sekä IHE-profiileissa että SOA-lähestymistavassa pyritään standardoimaan ratkaisujen yleiset ja yhteiset osat. IHE-profiileissa käytetään SOA-tyyppistä komponentti / aktorijajattelua ja työnkulkuun keskittyvissä IHE-profiileissa koreografia-tyyppistä työnkulkuajattelua. Joissakin IHE-profiileissa hyödynnetään SOA-ratkaisuissa yleisesti käytettyjä web services- ja XML-tekniikoita. Alla olevassa kuvassa esitetään IHE- ja SOA-lähestymistapojen painotuksia ja eroavaisuuksia luvun 2.3 viitemallin kautta.



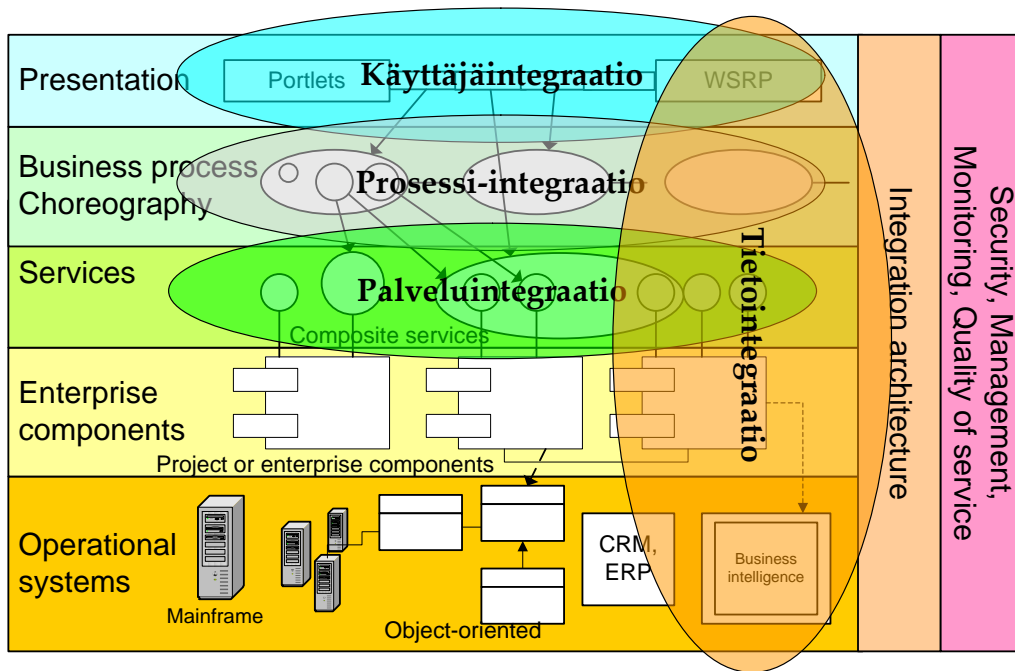
Kuva 28. SOA- ja IHE-lähestymistavat seitsentasoisessa viitemallissa.

6 Yhteenveto

Tämän dokumentin viitemallit, arkkitehtuurimallit, tutkitut määritykset sekä esimerkit valottavat monesta näkökulmasta palveluarkkitehtuuriin liittyvää yhteentoimivuutta ja sitä tukevaa standardointia. Samoin ne nostavat esiin erityisesti huomioitavia seikkoja ja jatkokehitystarpeita yhteentoimivuuden kehittämiseen. Kokonaisarkkitehtuurissa yhteentoimivuuden monet kriittiset tekijät sijoittuvat suhteellisen hienojakoiselle ja yksityiskohtaiselle tasolle. Tästä huolimatta monet kokonaisarkkitehtuurin peruselementtien ja myös ylempien abstraktiotasojen kuvaukset ovat erittäin tarpeellista pohjaa prosessien ja toiminnan tavoitteista lähtevälle sovellusten ja palvelujen integroinnille. Yhteentoimivuuksien elinkaaren aikana on käytännössä huomioitava monia vaatimuksiin, käytössä oleviin järjestelmiin, sekä käytettävissä oleviin sisällöllisiin ratkaisuihin liittyviä tekijöitä, mikä vaatii järjestelmällistä lähestymistapaa ja viitemalleja joiden avulla voidaan jäsentää tarpeita ja ratkaisuja. Standardeja on saatavilla runsaasti, ja niiden soveltaminen on usein perusteltua, mutta tämä vaatii potentiaalisten määrittelyjen tehokasta löytämistä ja arviointia. Organisaatiossa tai kohdealueella valituista kehittämissuunnitelmista riippuu, missä määrin pyritään "plug and play"-yhteensopivuuteen tuotteiden välillä, missä määrin taas korostetaan toiminnallista ja teknistä joustavuutta ja alustariippumattomuutta. Valmiiden yhteentoimivuuden kuvausten valinta ja yhdistely uusien tuottamisen sijaan on yleisesti valittu strategia SOA-pohjaisissa hankkeissa. Vastaava uudelleenkäyttöä korostava malli on nousemassa entistä keskeisemmäksi myös standardien sekä niiden soveltamisoppaiden määrittelyissä. Yleisiä pohjastandardeja rajoittavat profiilit ja soveltamisoppaat ovat keskeisiä välineitä riittävän tarkkuuden saavuttamisessa.

Integrointiratkaisujen määrittely lähtee perusvaatimusten tunnistamisesta ja potentiaalisten ratkaisujen luokittelusta. Kuva 29 havainnollistaa luvun 2.2 integrointitapojen käyttöä tähän tarkoitukseen: vaatimusten luonteesta riippuen voidaan päätyä hyvin erityyppisiin ratkaisuihin ja standardeihin. Kattavassa kokonaisarkkitehtuurissa ja laajoissa projekteissa on sovittava yhteisten periaatteiden ja peruselementtien lisäksi myös yksityiskohtaisista ratkaisuista. Kannatettavaa on esimerkiksi organisaatiossa pyrkiä vakioimaan tietyyppisten integraatioiden toteuttamisen tavat valmiiksi ja uudelleenkäytettäviksi "sablunoiksi". Olennaista on myös ratkaista suhde "ylätason" arkkitehtuurinhallintaan ja se, käytetäänkö integraatoratkaisuissa samaa muutostenhallintamekanismia kuin arkkitehtuurin hallinnassa. Tähän vaikuttaa se, soveltuuko vaatimusten ja muutosten hallintamalli sekä korkean että yksityiskohtaisen abstraktiotason kehittämiseen. Tarpeiden, vaatimusten ja ratkaisujen hallinnassa käytettävissä menetelmissä (ks. esim. Tiihonen ym. 2012) on voitava huomioida, että esimerkiksi yksittäisten integrointiratkaisujen yksityiskohdista päättämisessä vaaditaan erityyppistä osaamista ja näkemystä kuin vaikkapa organisaation strategian tai toiminta-arkkitehtuurin tavoitteiden määrittelyssä.

Palvelukeskeisen SOA-lähestymistavan käytöllä saavutettuja uudelleenkäyttö- ja integraatiohyötyjä on jonkin verran näkyvissä eri kohdealueille toteutettujen kehittämissuunnitelmien, yksittäisten organisaatioiden ja kohdealueiden arkkitehtuuriohjauksen sekä standardointijärjestöjen suunnalta. SOA-lähestymistavan strategia joustavuus- ja ketteryyshyötyjä on kuitenkin todennettu ja raportoitu paljon vähemmän. Samaten tietojärjestelmien kehittämisessä kulttuurimuutos kertahankinnoista jatkuvaksi kehittämiseksi on pääosin kesken. SOA-ratkaisuista on jo tullut arkipäivää etenkin integraationäkökulmasta, mutta etenkin prosessimallinnuksen ja järjestelmäkehityksen välissä on vielä runsaasti aukkoja, mitä kuvaa myös SOLEA-työpajoissa siteerattu kehittämissuunnitelmassa esitetty kommentti "heitetäänkö prosessikuvaukset menemään ja tehdään käyttötapauksista?".



Kuva 29. Integrointitapojen sijoittuminen SOA-viitearkkitehtuurissa.

HL7 SAIF-mallin tyyppinen pohdinta integraation tavoitetasosta on osa tarpeiden ja vaatimusten varhaista analyysia joka ohjaa vahvasti määriteltäviä ratkaisuja. Sisäisessä integraatiossa yksinkertaisuus, nopeus ja joustavuus ovat tärkeitä tavoitteita, jolloin voidaan tähdätä suhteellisen vaivattomiin ja yksinkertaisiin ratkaisuihin. Monenvälisessä ja standardi-integraatiossa kaikkia olennaisia seikkoja ohjaavat sopimukset ovat tärkeämmässä roolissa. Standardien hyödyntämisestä voidaan kuitenkin saavuttaa hyötyjä molemmissa tapauksissa sekä yksityiskohtaisten rajapinta- ja semantiikkamäärittelyjen että yhteisten viitemallien tasolla. SOA-standardoinnissa korostuvat erityisesti myös palvelujen toiminnallisten mallien määrittelyt.

Sekä sisäisessä että ulkoisessa integraatiossa on profiili-ajattelun lisäksi näkyvissä pyrkimys yksinkertaistaa integraatoratkaisuja ja standardeja. Samalla haetaan mm. selkeämmän eri näkökulmien tai ratkaisun piirteiden erottamisella entistä modulaarisempia ja joustavampia ja toisiaan tukevia standardeja. Näkyvissä on kehitystä, jossa integraatoratkaisut tiukasti "sementoivia" määrittelyjä kuten Web Services-määrittelyjen ja XML Scheman käyttöä haastetaan joustavammilla ja yksinkertaisilla teknisillä ratkaisuilla kuten REST-protokollat sekä näyttömuotoisten dokumenttien täydentäminen semanttisilla merkkauksilla (esimerkiksi XHTML + RDFa / HTML5+Microdata).

Vastaavasti alustaratkaisut kuten palvelualustan käyttö lisäävät joustavuutta ja soveltamismahdollisuuksia integraation ja palvelupohjaisen kehittämisen kannalta. Palveluarkkitehtuuria on mahdollista toteuttaa ilman integrointi- tai palvelualustoja, mutta tämä vaatii tiukempaa arkkitehtuuriohjausta ja voi asettaa myös teknisiä rajoitteita toiminnallisesti sopivien komponenttien hyödyntämiselle tietojärjestelmäkokonaisuudessa. Etenkin moniprotokolla- ja monitoimijaympäristössä palvelualustojen avulla saavutettavat hyödyt voivat olla merkittäviä. Arkkitehtuuriohjausta tarvitaan joka tapauksessa, esimerkiksi on pyrittävä huolehtimaan siitä, etteivät toimintapolitiikka- ja turvallisuusyksityiskohdat sotkeudu rajapinnoissa toiminnallisiin ja prosessitason ominaisuuksiin.

Sekä kokonaisarkkitehtuurissa että yhteentoimivuuden kuvauksissa yleisistä tavoitteista ja ylätasoinen kuvauksista on yleensä helppo olla samaa mieltä ja löytää yhteisiä näkemyksiä. Ratkaisujen käytännön toteuttamiseen ja standardien soveltamiseen tarvitaan kuitenkin suuri määrä yksityiskohtaista työtä. Integraation tarpeet kasvavat edelleen yhteiskunnan ja organisaatioiden sekä prosessien

verkottuessa ja myös yksilöiden oman tiedonhallinnan tarpeiden ja välineiden kehittyessä. Edellä mainitut joustavuuteen ja ratkaisujen nopeuttamiseen liittyvät tarpeet asettavat uusia vaatimuksia sekä ohjauksen ja kehittämisen tavoille että teknisille ratkaisuille. Yhteentoimivuuden elinkaaren nopeuttaminen vaatimuksista käyttöönottoon sekä yhteentoimivuusratkaisujen toteutuskynnyksen madaltaminen ovat edellytyksiä sille, että organisaatioiden palvelut pystytään pitämään kilpailukykyisinä ja vastaamaan niille asetettuihin yhä kasvaviin vaatimuksiin. Palveluarkkitehtuuri ja standardit tarjoavat näiden tavoitteiden saavuttamiseen joitakin välineitä, mutta niiden keskeisten toteuttamis- ja määrittelytapojen tarkentaminen paikallisiin tarpeisiin, yhteentoimivuusratkaisujen ja -kuvausten tehokas määrittely ja toteuttaminen sekä yleiskäyttöisyyden ja tarkkuuden yhteensovittaminen ovat edelleen keskeisiä tutkimuksen ja kehittämisen kohteita.

Lähteet

APQC 2008. Process Classification Framework. Version 5.0.3-en-XI, April 2008, American Productivity & Quality Center (APQC). <http://www.apqc.org/process-classification-framework>

Arsanjani A., Liang-Jie Zhang, Ellis, M., Allam, A., Channabasavaiah K.: S3: A Service-Oriented Reference Architecture. IT Professional, issue 3, 2007, s. 10-17.

Chappell D. Enterprise Service Bus, O'Reilly, 2004.

EC 2010. European Interoperability Framework for European Public Services (EIF). Annex 2 to the Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of Regions 'Towards interoperability for European public services'. European Commission, 2010. (Version 2.0).

Erl T. SOA Principles of Service Design. Prentice Hall, 2007.

Friedrich J. Vivid public debate around the EIF and openness. Jochen Friedrich's Open Blog, 12 November 2009, <http://jfopen.blogspot.com/2009/11/vivid-public-debate-around-eif-and.html>

Heinze O, Birkle M, Köster L, Bergh B. Architecture of a consent management suite and integration into IHE-based regional health information networks. BMC Medical Informatics and Decision Making 2011, 11:58. doi:10.1186/1472-6947-11-58

Herzum P, Sims O. Business Component Factory. Wiley, 2000.

HL7 2011. HL7 Service-Aware Interoperability Framework - Canonical Definition, Release 1. HL7 Informative Document, Architectural Review Working Group, Health Level Seven International, 2011.

Itälä T, Mykkänen J, Virkanen H, Tiihonen T, Hiekkänen K, Luukkonen I, Sammelvuola I, Melleri I, Han Y. Kokonaisarkkitehtuurin ja palveluarkkitehtuurin menetelmät ja välineet. SOLEA-hanke, Itä-Suomen yliopisto, Aalto-yliopisto, 2012.

JHS 2011. JHS 181 Julkisen hallinnon standardisalkku, versio 1.0. JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta, 2011. <http://www.jhs-suositukset.fi/suomi/jhs181>

Kawamoto K, Honey A, Rubin K. The HL7-OMG Healthcare Services Specification Project: Motivation, Methodology, and Deliverables for Enabling a Semantically Interoperable Service-oriented Architecture for Healthcare. Journal of the American Medical Informatics Association 2009;16(6):874-881.

Kela 2011. Liittymismallit KanTa-palveluihin. KanTa-palveluryhmä, Kela, 3.2.2011. https://www.kanta.fi/c/document_library/get_file?uuid=df9dcc8e-0cf7-4b8d-b585-ebdb1aa844b4&groupId=10206

Kauppi A. Connectivity to the National Service. SOLEA-työpajaseminaari 2.9.2010, Kuopio. Inter-Systems Nordic, 2010.

Kauppi A. KanTa liityntäpiste - tilannepäivitys. Ensemble käyttäjätapaaminen 20.10.2011, Intersystems Nordics.

Korhonen M. Ensemble Kanta-palveluiden liityntäpisteenä Pohjois-Savon sairaanhoitopiirissä, Testausprojekti. SOLEA-työpajaseminaari 2.9.2010, Kuopio.

Kotinurmi P, Mykkänen J, Itälä T. Conflicts of local adaptation and vertical standards in SOA. Submitted, 2008.

Kreger H, Estefan J (toim.). Navigating the SOA Open Standards Landscape Around Architecture. The Open Group, Object Management Group and OASIS, November 2009. <http://pubs.opengroup.org/onlinepubs/7699909399/toc.pdf>

KuntaIT 2010. eReseptin integraatioarkkitehtuurin perusteet. Versio 1.0. KuntaIT Arkkitehtuuri, 19.4.2010. https://www.kanta.fi/c/document_library/get_file?uuid=ac92858c-27df-4241-92c2-b87514394376&groupId=10206

Laki julkisen hallinnon tietohallinnon ohjauksesta 10.6.2011/634.

Linthicum D. Next Generation Application Integration: From Simple Information to Web Services. Addison-Wesley, 2003.

Mykkänen J. Specification of Reusable Integration Solutions in Health Information Systems. Doctoral dissertation. 2007. Kuopio University publications H. Business and Information Technology 6.

Mykkänen J, Porrasmäe J, Rannanheimo J, Tikkanen T, Sormunen M, Korpela M, Häyrinen K, Eerola A, Häkkinen H, Toivanen M. Terveysteknologioiden sovellusintegraatoratkaisujen määrittely. Kuopion yliopisto, Savonia-ammattikorkeakoulu, 2004. PlugIT-hankkeen selvityksiä ja raportteja 4. <http://www.uku.fi/plugit/julkaisut/04-Mykkanen-ym.pdf>

Mykkänen J, Savolainen S, Virkanen H, Itälä T, Kortekangas P. *Palvelutapahtumien hallinta. Arkkitehtuuritarkennuksia terveydenhuollon valtakunnallisten, alueellisten ja paikallisten tietojärjestelmäratkaisujen kannalta*. SOLEA-hanke, Itä-Suomen yliopisto, Aalto-yliopisto, 2012.

Mykkänen JA, Tuomainen MP. An evaluation and selection framework for interoperability standards. Inform Software Tech 2008;50(3):176-197.

Mykkänen J, Tuomainen M. Balancing between Interoperability Standards, Local Requirements and SOA principles - Case eBooking of Health Services. Submitted, 2012.

Mykkänen J, Virkanen H. Governance in Extended Lifecycle of Large-Scale eHealth Initiatives: Analyzing Variability of Enterprise Architecture Elements. Submitted, 2012.

Mykkänen J, Virkanen H. Integrating the Healthcare Enterprise (IHE) -hyödyntämisselvitys. Tekes, Terveysteknologian liitto FIHTA, Kuopion yliopisto, 2009.

Mykkänen J, Virkanen H, Kortekangas P, Savolainen S, Itälä T. Task Analysis and Interoperable Application Services for Service Event Management. In: Moen A, Andersen SK, Aarts J, Hurlen P, eds. *User Centred Networked Health Care - Proceedings of MIE 2011*, p. 295-299. Amsterdam: IOS Press, 2011. Studies in Health Technology and Informatics 169.

Nelson ML, Shaw MJ. The adoption and diffusion of interorganizational system standards and process innovations. MISQ Special Issue Workshop: Standard Making: A Critical Research Frontier for Information Systems, Seattle, Washington, USA, 258-301, 2003.

Open Group 2009. TOGAF 9 - The Open Group Architecture Framework. Document Number: G091, The Open Group, 2009.

OSIMM 2011. Information technology — The Open Group Service Integration Maturity Model (OSIMM). International Standard Final Draft ISO/IEC FDIS 16680:2011(E).

Painter J, Kirnak A, Moehrke J. A Service-Oriented Architecture (SOA) View of IHE Profiles. IHE IT Infrastructure White Paper, Public Comment. IHE International, 2009.

Rubin K. Healthcare Architecture for the Real World - Making EA, SOA, and Localization more than buzzwords. SOLEA Symposium, March 2009, Espoo.

Sosmeta 2011. Sosmeta.fi - sosiaaliuollon asiakastietomalli. Sosiaalialan tietoteknologiahanke Tikesos, 2011. <http://sosmeta.fi/>

Viinikainen H, Paakkanen E, Komulainen J, Aholainen J, Mykkänen J, Hyppönen K, Suhonen M, Tuomainen M, Leinonen P. Sosiaaliuollon kansallinen kokonaisarkkitehtuuri - Standardisalkku. Versio 1.4. Sosiaalialan tietoteknologiahanke, STM, 2011.

VM2011a. Julkisen hallinnon kokonaisarkkitehtuuri - Julkisen hallinnon arkkitehtuurilinjaukset ja -kuvaukset - Määrittely, versio 0.95, 4.4.2011, Valtiovarainministeriö.

Liite 1. Sanasto SOLEA-hankkeen keskeisistä käsitteistä

Käsite	Kuvaus
Ajurit	Ulkoiset toimintaympäristön tekijät, jotka vaikuttavat toimintaan tai sen tavoitteisiin (erityisesti kontekstivastauksen tarkastelu), kuten lainsäädäntömuutokset, markkinoiden tai palvelujen kysynnän kehitys, mille ei tunnustettavissa sisäistä omistajaa.
Aktiviteetti-kaavio	Kaaviotyyppi, joka on tarkoitettu erityisesti aktiviteettien, kuten työkulkujen, liiketoimintaprosessien sekä rinnakkaisia toimintoja sisältävien järjestelmien sekä niiden välisten suhteiden sekä UML versiossa 2 prosessien kuvaamiseen (Fowler 2004, KuntaIT).
Aliprosessi	Aliprosessi (esim. yksittäinen SOA-palvelu) on pääprosessin osa. Vrt. Osaprosessi.
Arkkitehtuuri-periaate	Yleinen kokonaisarkkitehtuuria tai jotain sen osa-aluetta yli useiden eri projektien ohjaava periaate, jonka perusteella voidaan tehdä valintoja erilaisten vaihtoehtojen välillä. Periaatteet ovat yleisempiä kuin linjaukset eli niitä ei ole välttämättä kohdistettu mihinkään yksittäiseen kehittämiskohteeseen.
Asiakasprosessi	Prosessi, jossa kuvataan asiakkaalle annettava palvelu. Lähellä ydinprosessia joissakin tapauksissa.
Asiantuntijatyön prosessi	Prosessi, jonka vaiheiden järjestys tai sisältö perustuu tyypillisesti jossain vaiheessa prosessia asiantuntijan hiljaisen tiedon tai kokemuksen hyödyntämiseen ja asiantuntijuuteen, jota on vaikea automatisoida. Usein dynaaminen.
Automatisointi	Manuaalisten työvaiheiden tuottaminen tietoteknologian avulla.
Dynaaminen prosessi	Prosessi, jonka vaiheet tai niiden järjestys eivät ole tarkalleen etukäteen määriteltyjä; prosessin osat voivat olla suunnilleen samoja, mutta suoritusjärjestys vapaampi. Vrt. Staattinen prosessi.
EA	Ks. Kokonaisarkkitehtuuri (engl. Enterprise Architecture)
Ei-toiminnalliset vaatimukset	Määrittelee rajoitukset ja reunaehdot toiminnallisille vaatimuksille. Ei-toiminnalliset vaatimukset eivät liity suoraan palveluihin vaan kertovat, mitä ehtoja järjestelmän on täytettävä, jotta toiminnalliset vaatimukset voidaan toteuttaa (JHS 173). Esim: vasteaika vaatimus ja saumattomuus.
IT-järjestelmä	IT-järjestelmällä tarkoitetaan organisaation koko teknistä järjestelmää sisältäen laitteistot, tietoverkot sekä ohjelmistot. Vrt. Tietojärjestelmä.

JHS	Julkisen hallinnon suositukset (www.jhs-suositukset.fi).
Järjestelmävaatimus	Ilmaisee mitä, millä ehdoin ja kuinka hyvin järjestelmän on tehtävä (jotain) tai millainen sen on oltava (reunaehto) sidosryhmien tarpeiden tyydyttämiseksi (JHS 173).
Kehittämistavoitteet	Organisaation sisäiset tietyn kokonaisuuden kehittämiseen liittyvät tavoitteet, esim. tietyn toiminnon tehostaminen, tietojärjestelmäkokonaisuuden hankinta tai kehittäminen, uuden toimipisteen tai kumppanin hankinta, uuden prosessin kehittäminen, prosessin uudelleensuunnittelu.
Kohdearkkitehtuuri	Kohdearkkitehtuuri on yhteenkuuluvan rajatun alueen arkkitehtuurikokonaisuus kattaen toiminnan, tiedon, järjestelmät ja teknologiat. Se luo puitteet kyseisten keskitettyjen palveluiden tarkemmalle suunnittelulle ja toteuttamiselle jäsentäen ja määrittäen arkkitehtuurin keskeiset rakenneosat.
Kokonaisarkkitehtuuri	Synonyymi: Yritysarkkitehtuuri, joka on yksityisellä sektorilla käytetty nimitys kokonaisarkkitehtuurista (suom. KA, engl. Enterprise Architecture, EA). <i>Menetelmäpainotteinen määritelmä:</i> Kokonaisarkkitehtuurilla tarkoitetaan toiminnan, tietotarpeiden, tietojärjestelmien ja teknologiaratkaisujen mallintamista, kuvaamista ja suunnittelemista yhtenäisen mallin mukaisesti (KuntaIT). <i>Tuotospainotteinen määritelmä:</i> Kokonaisarkkitehtuuri on toiminnan, prosessien ja palvelujen, tietojen, tietojärjestelmien ja niiden tuottamien palvelujen muodostaman kokonaisuuden rakenne (JHS 171).
Kuvaus	Ks. Malli.
Kuvaustapa	Kuvaustapa on kuvaustyyppin tarkennus, esim. TOGAF:in prosessitaulukko. Samaa kuvastapaa (esim. uimaratakaavio) voidaan tarvittaessa toteuttaa erilaisten notaatioiden avulla.
Kuvaustaso	Kuvaustaso kertoo, miten tarkalla tasolla kuvauksen kohdetta, kuten prosesseja tai toimintaa, kuvataan/mallinnetaan, kuinka suuren (organisatorisesti) ja yleistettävän (yksityiskohtien abstrahointi) kokonaisuuden kuvaus kattaa; esim. organisaatiotason yleiskuva, yksi prosessi, henkilön tai yksikön toiminta, palvelu tai sen operaatio.
Kuvaustyyppi	Kuvaustyyppi on tuotettavien kuvausten perusmuoto/-rakenne, kuten matriisi, kaavio, taulukko, teksti, hakemisto tai lista.
Käyttjävaatimus	Määrämuotoinen ilmaisu siitä mitä, kuinka hyvin ja millä rajoituksin käyttäjä (tai muu sidosryhmä) haluaa järjestelmällä tehdä tai aikaansaada, tai mitä ominaisuuksia järjestelmän on omattava. Vaatimuksella on varottava ilmaisemasta erityistä ratkaisua tarpeeseen tai ongelmaan (JHS 173).

Liiketoiminta-strategia	Liiketoiminta- eli kilpailustrategian voidaan sanoa olevan se toimintatapa, jolla yritys kilpailee markkinoillaan, ja kuinka se yrittää luoda kilpailijoihinsa nähden kilpailuetua luoden edellytykset yrityksen olemassaololle (Simons, 1990). Lähdeettäessä määrittelemään yrityksen kilpailustrategiaa on edellytyksenä se, että on ratkaistu, missä liiketoiminnassa ja millaisin päämäärin ollaan liikkeellä. Tämä määrittely lukitsee samalla toimialan, missä yritys operoi, tuotteet mitä se asiakkailleen tarjoaa ja markkinat missä se toimii.
Linjaus	Linjaus kohdistuu johonkin määriteltyyn kehittämisen kohteeseen. Linjaus on vastaavan tahon hyväksymä ja se on ainakin jossain määrin sitova. Se kuvaa, mitä tullaan tekemään tai mitä kuvausta tai tarkempaa määrittelyä pitää käyttää tiettyyn määriteltyyn kehittämisen kohteeseen. Linjauksia voi olla myös suosituksissa, standardeissa, asetuksissa ja laeissa.
Malli	Malli koostuu kuvauksen sisällöstä ja kuvaustavasta. Se, mille tasolle yksittäinen malli kuuluu, määräytyy sisällön perusteella.
Notaatio	Notaatiolla tarkoitetaan mallinnuskielen graafisia komponentteja. Notaatio tarkoittaa sääntöä, jonka mukaan menetelmän käsitteistöä mallinetaan. Notaatio ottaa kantaa siihen, kuinka esim. kaaviossa esiintyvä luokka esitetään, esim. suorakaiteella vai ympyrällä (KuntaIT).
Ohjelmisto	<i>Ohjelmisto</i> tai <i>tietokoneohjelmisto</i> on useista tietokoneohjelmista, niiden käyttämistä tiedostoista ja niihin liittyvästä dokumentaatiosta muodostuva kokonaisuus.
Oletus	Tiettyyn kehittämisen kohteeseen tai ympäristötekijään liittyvä oletus tulevasta kehityksestä tai nykytilasta, jota ei voida pitää varmana. Voidaan käyttää perusteluina erilaisille tavoitteille ja ratkaisuvaihtoehdoille.
Organisaatio	Hallinnollinen yksikkö, esim. Kuopion yliopistollinen sairaala.
Organisaatio-yksikkö	Organisaation sisällä oleva yksikkö, esim. ihotautipoliklinikka.
Osaprosessi	Osaprosessit ovat ydin- tai tukiprosessien osia (JHS 152). Kuvaamistavat kuten prosessilla, mutta yksityiskohtaisemmin. Voi olla myös aliprosessi.
Osatoiminta	Toimintaan kuuluva, pienempi kokonaisuus. Kuvataan kuten toiminta.
Palvelu	<i>Toiminnallisessa arkkitehtuurissa:</i> Sopimuksen avulla kuvattu joukko ominaisuuksia, joiden avulla palvelun tarjoaja tuottaa haluttuja tuloksia palvelun käyttäjälle. <i>Sovellusarkkitehtuurissa:</i> Sopimuksen / määrittelyn avulla kuvattu joukko palvelun tarjoajan tarjoamia tietoja ja toimintoja, joiden avulla palvelun käyttäjä pystyy kokoamaan prosesseja tai sovelluksia. SOA-arkkitehtuurissa palvelut toimivat keskiössä ja ovat näin järjestelmien toiminnan edellytys. Palvelut toimivat toiminnallisuuden ja sovelluslogiikan rakennusosina (KuntaIT).

Palvelukeskeisyys	Palvelukeskeiseen avoimeen arkkitehtuuriin kuuluu perusinfrastruktuuri ja valmiita yleiskäyttöisiä tukipalveluja, komponentteja ja rajapintoja, joita voidaan suoraan hyödyntää palvelujen rakentamisessa (KuntaIT).
Palvelupohjainen arkkitehtuuri (SOA, service-oriented architecture)	Tietojärjestelmien kehittämisen lähestymistapa, jossa sovelluksia tai toimintaprosesseja muodostetaan pienemmistä, määriteltyjä osatehtäviä toteuttavista palveluista. Tietojärjestelmäkokonaisuus hahmotetaan joukkona palveluita (sovelluspalveluita), joita tarpeen mukaan yhdistelemällä voidaan entistä helpommin toteuttaa tai mukauttaa sovelluksia eri käyttötarpeisiin (Mykkänen 2004).
Poikkeus	Prosessissa tai palvelussa tapahtuva normaaliin työn tai suorituksen kulkuun kuulumaton tapahtuma, joka tyypillisesti estää etenemisen tai odotettujen tulosten tuottamisen.
Prosessi	Sarja toisiinsa liittyviä toistuvia toimintoja tai tapahtumia, joiden avulla prosessiin liittyviä resursseja käyttäen päästään lisäarvoa tuottavasti toivottuun tulokseen. Prosessilla on selkeä alku ja loppu.
Prosessiaskel	Prosessiaskel tarkoittaa toiminnan etenemistä eli prosessin tai sen osan siirtymistä vaiheesta toiseen (JHS 152). Vrt. Prosessin vaihe.
Prosessikaavio	Prosessikaavio on tapa kuvata prosessin toiminnot graafisesti. Prosessin toiminnot, tietovirrat ja tuotteet kuvataan sovitulla symboleilla. Prosessikaavio auttaa ymmärtämään toimintojen järjestystä ja niiden välisiä riippuvuuksia (JHS 152). Voi olla esimerkiksi vuokaavio.
Prosessikartta	Organisaation tasolla tehty yleinen, usein graafinen kuvaus organisaation tärkeimmistä prosesseista ja niiden välisistä yhteyksistä (JHS 152).
Prosessin frekvenssi	Miten usein prosessi toistuu.
Prosessin omistaja	Prosessin omistaja on prosessin toiminnasta, tuloksesta, tuloksellisuudesta ja kehittämisestä vastuussa oleva henkilö (JHS 152). Prosessiin nimetty vastuullinen toimija, jonka tehtävänä on koordinoida omistamansa prosessin kuvausta, käyttöönottoa, vakiinnuttamista ja kehittämistä sekä seuranta.
Prosessin vaihe	Prosessille lisäarvoa tuottava toimijan (ihminen/sovellus) aliprosessi toiminto tai tapahtuma. Prosessin vaiheesta toiseen siirrytään prosessiaskelten kautta.
Sekvenssikaavio	Kaaviotyyppi, joka esittää uimaradoilla kuvattujen osapuolten väliset kutsut ja niiden välisen järjestyksen tai tietoliikenteen.

Rajoite	Selkeästi ja tarkoituksella rajoittaa suunnittelua, toteutusta, käyttöä, elinkaarta tai päätöksentekoa; kehittämistä ohjaavat projektin linjaukset, joiden avulla rajoitetaan mahdollisten suunnittelupäätösten joukkoa. Esim. tietyn teknologian käyttö, lain asettama vaatimus, yhteisen tietomallin määrittelemien tietojen käyttö projektikohtaisen sijaan.
Simulointi	Simuloinnissa toiminta kuvataan mallin avulla, johon liitetään toimintaa kuvaavia parametreja ja luodaan matemaattinen malli. Simulointimallin avulla (parametreja muuntamalla) voidaan arvioida toiminnan sujuvuutta eri tilanteissa. Simulointi toteutetaan usein tietokoneella.
SOA	Ks. Palvelupohjainen arkkitehtuuri.
Sovellus	Tietokoneohjelma/-ohjelmisto (engl. application).
Staattinen prosessi	Pysyväluonteinen prosessi; prosessin osat ja suoritusjärjestys määritellyt. Vrt. Dynaaminen prosessi.
Standardi	Tunnustetun osapuolen hyväksymä dokumentti, jossa on määritelty yleistä ja toistuvaa käyttöä varten sääntöjä, ohjeita tai piirteitä tuotteille, prosesseille tai palveluille (Project Management Institute 2000).
Strategia-prosessi	Liiketoiminnan tavoitteellinen kehittäminen tai laadunhallinnallinen jatkuva parantaminen. Kokonaisarkkitehtuurin suunnitteluprosessissa tärkeimpänä syöteenä toimii organisaation johdon asettama strategia (JHS 179).
Strategiset tavoitteet	Organisaation sisäiset, koko organisaation toimintaa ja erityisesti kehittämistä ohjaavat tavoitteet, esim. markkinaosuus, toiminnan volyymin tavoitearvot, uusien palvelualueiden kehittäminen ja organisaatioiden yhdistäminen.
Syöte	Syöte on tietoja ja materiaalia, joka syötetään prosessiin, palveluun tai sovellukseen (Laamanen & Tinnilä 2009). Syöte ei ole sama asia kuin raha, laitteet tai ihmisten osaaminen, jotka ovat resursseja ja siten osa prosessia (JHS 152).
Tarve	Jonkin halutun, tarvitun tai käytännöllisen ominaisuuden/asian puuttuminen. Kuvatut ja määritellyt tarpeet voidaan nähdä vaatimuksina.
Tehtävä	Yksittäisen toimijan määritelty joukko toimenpiteitä halutun tuloksen aikaansaamiseksi. Prosessikuvauksen sisällä käytettävä käsite tehtävä (engl. task) tai osatehtävä tarkoittaa käsittelyvaihetta. Nämä tehtävät ovat yleensä yksilön tai ryhmän suorittamia käytännön toimenpiteitä (JHS 152).
Teko	Yksittäisen toimijan lyhykestoinen toimenpide (operation) halutun tuloksen aikaansaamiseksi.

Teknologia-arkkitehtuuri	Kokonaisarkkitehtuurin näkökulma, joka kuvaa organisaation teknologista infrastruktuuria ja järjestelmäarkkitehtuurin teknologiavalintoja. Teknologia-arkkitehtuurin suunnittelussa linjataan käytettävät tekniset järjestelmien ja ICT-infrastruktuurin ratkaisuvaihtoehdot, standardit ja rakenteet siten, että kokonaisuus tukee parhaalla mahdollisella tavalla organisaation tavoitteita (JHS 179).
Tieto-arkkitehtuuri	Kokonaisarkkitehtuurin näkökulma, joka kuvaa informaation rakenteistamista, organisointia ja luokittelua sekä välitystä. Tarkastellaan organisaation tietotarpeita, tietopääomaa, tietojen välisiä suhteita, informaatioarvoketjua, tietojen rakenteita sekä informaation organisointia ja hallintaa (JHS 171).
Tietojärjestelmä	<i>Sosiotekninen määritelmä:</i> Tietojärjestelmä (engl. Information System) käsittää siihen sisältyvän tiedon (data, informaatio, tieto, tietämys), teknologian, työtoiminnan, kommunikaation, organisaation ja ihmiset. Siten tietojärjestelmän perimmäinen tarkoitus on palvella sitä työtä ja sitä työtoimintaa, mihin se on tarkoitettu käytettäväksi palvelun tuottamiseksi (Mursu & Ikävalko 2007). <i>Välinekeskeinen määritelmä:</i> Tietojärjestelmän avulla tietoa tallennetaan, säilytetään, välitetään ja hyödynnetään. Voi sisältää sekä manuaalisia, sähköisiä että toiminnallisia elementtejä. Tietoa käsittelevät toimijat (ihmiset ja ohjelmistot) sekä itse tieto (data) ovat osa järjestelmää.
Tieto-järjestelmä-arkkitehtuuri	Kokonaisarkkitehtuurin näkökulma, joka kuvaa organisaation keskeiset järjestelmät sekä niiden arvioidun elinkaaren, kriittisyyden, niiden käyttämät/tuottamat tiedot ja suhteet muihin järjestelmiin. Organisaation järjestelmäpääoma (JHS 171).
Tietokokonaisuus	Toisiinsa kiinteästi liittyvä joukko tietoja, esim. asiakirja tai samassa tehtävässä tarvittavien tietoelementtien muodostama kokonaisuus.
Tietovirta	Tiedon liikkumisen suunta ja sisältö kahden tai useamman toimijan tai prosessin vaiheen/toiminnon välillä.
Tietoväline	Väline, jonka avulla tietoa välitetään. Voi olla manuaalinen, sähköinen tai toiminnallinen esim. paperilomake, ilmoitustaulu, ohjelmisto, puhelin tai palaveri.
Toiminnalliset vaatimukset	Määrittelee kehitettävän tai hankittavan järjestelmän käyttäytymistä tai toiminnallisuutta, kuten mitä palveluja ohjelmiston on tarjottava, miten ohjelmisto reagoi syötteisiin ja miten se käyttäytyy annetuissa tilanteissa. Voi olla käyttäjä tai järjestelmävaatimus. Vrt. Ei-toiminnalliset vaatimukset.
Toiminta	Yksittäisen tai kollektiivisen toimijan tavoitteellinen joukko tekoja, joilla pyritään haluttuun lopputulokseen.
Toiminta-arkkitehtuuri	Kokonaisarkkitehtuurin näkökulma, joka kuvaa organisaation strategisiin vaatimuksiin liittyvää ydintoimintaa ja sitä tukevia tukiprosesseja, resursseja sekä palvelutarjontaa (JHS 171).

Toiminta-kokonaisuus	Useiden toimintojen muodostama joukko, joista kullakin toisiinsa liittyvää toimintaa. Joukko tehtäviä, joilla saadaan aikaan tietty tulos.
Toimintatarina	Tyypillisesti yhden toimijan näkökulmasta tehty sanallinen kuvaus tai esimerkki prosessin etenemisestä tai toiminnasta.
Toiminto	Toiminnoksi kutsutaan joukkoa tehtäviä, joiden avulla saadaan aikaan tietty tulos (JHS 152). SOA-kehittämisessä: ohjelmiston tai sovelluspalvelun tarjoama toiminnallinen ominaisuus.
Tapahtuma (event)	Havainnoitava lyhytkestoinen sisäinen tai ulkoinen ilmiö, joka voi esim. käynnistää prosessin tai prosessin vaiheen. Esimerkiksi tilauksen saapuminen on heräte tilauksen käsittelylle.
Tukiprosessi	Tukiprosessit avustavat ydinprosesseja ja luovat edellytykset niiden toiminnalle. Tukiprosesseilla voi olla sisäisiä asiakkaita. Tukiprosesseja ovat muun muassa hallinnolliset toiminnot, kuten henkilöstöhallinto sekä osaamisen ja toimintojen kehittäminen (JHS 152), sekä johtamisen, viestinnän, talous-, henkilöstö-, tieto- ja kiinteistöhallintojen sekä materiaalihuollon prosessit.
Operaatio	Ohjelmiston toteuttama toiminta, jonka tarkoituksena on aiheuttaa tietokoneen muistiin, tietokantaan tai tietojärjestelmään tallennettujen tietojen haku sekä mahdollisesti käsittely ja käsittelyn tuloksena olevien tietojen palauttaminen.
Työnkulku	Manuaaliset ja automatisoidut toimintoketjut, jotka voivat koostua yhden tai useamman toimijan toiminnoista tai teoista (manuaaliset) tai operaatioista (automatisoidut IT-palvelut).
Työnkulku-kaavio	Kaavio, jolla työnkulku kuvataan. Esim. uimaratakaavio.
Työtoiminta	Työtoiminta (work activity) tarkoittaa kokonaisuutta (toimintajärjestelmä), jossa <i>joukko ihmisiä työskentelee järjestäytyneellä tavalla jonkin yhteisen kohteen parissa – ei välttämättä samassa paikassa ja yhtä aikaa – tuottaakseen jonkin yhteisen lopputuloksen</i> (Korpela 1994).
Työtoiminnan analyysi ja kehittäminen	(engl. ActAD: Activity Analysis and Development) Toiminnan teoriaan ja kehittävään työn tutkimukseen perustuva malli, jonka avulla voidaan jäsentää monimutkaista, monen ihmisen yhteistoimintana tapahtuvaa toimintaa (Korpela 1994). Ks. Työtoiminta.
Uimarata	Uimarata on visuaalisessa prosessin kuvaamisessa käytetty tapa ilmaista eri rooleja. Kukin rooli (organisaation tai henkilön) kuvataan omana uimaratanaan (pysty- tai vaakasuoraan), ja tällöin sille sijoitetut prosessiasteet kuuluvat tämän roolin vastuulle (JHS 152).

SOLEA

Vaatus	Tiettyyn kehittämisen kohteeseen kohdistuva dokumentoitu tarve, jonka toimeenpanosta ja johon liittyvästä ratkaisusta voidaan tehdä päätöksiä.
Viite-arkkitehtuuri	Viitearkkitehtuuri on rajatun arkkitehtuurikonaisuuden abstrakti toimittaja- ja toteutusneutraali rakenne. Se on esitys arkkitehtuurikonaisuuden loogisista osista ja niiden välisistä suhteista. Viitearkkitehtuurilla ohjataan arkkitehtuurisuunnittelua halutunlaiseen toteutusrakenteeseen. Viitearkkitehtuuri voi olla organisaation sisäinen, toimialaan liittyvä tai yleinen looginen rakennemalli (JHS 179).
Vuokaavio	Tietojenkäsittelyn toimintasarjaa kuvaava, toimintoja esittäviä sovittuja symboleja käyttäen laadittu piirros, josta käy selville suoritusjärjestys (JHS 152). Vuokaaviosymbolit on Suomessa standardoitu (SFS-3204).
Ydinprosessi	Organisaation ydinprosessit kehittävät, tuottavat ja toimittavat (ulkopuolisten) asiakkaiden tarvitsemia materiaaleja, tavaroita ja tietoja (tuotteita) (JHS 152).
Yritys-arkkitehtuuri	Ks. Kokonaisarkkitehtuuri

Sanaston lähteet

- Fowler M. UML Distilled Third Edition. A Brief Guide to the Standard Object Modeling Language. Pearson Education Inc. USA, 2004.
- Gotel O.C.Z, Finkelstein A.C.W. An Analysis of the Requirements Traceability Problem, Proceedings of the 1st International Conference on Requirements Engineering (ICRE 1994), s. 94-101, Colorado Springs, U.S.A., April 18-22, 1994.
- Grant R.M. (1998). Contemporary strategy analysis (3. painos).
- JHS 152. Prosessien kuvaaminen. Julkisen hallinnon tietohallinnon neuvottelukunta. Versio 6.6.2008. Saatavissa: <http://docs.jhs-suositukset.fi/jhs-suositukset/JHS152/JHS152.pdf>
- JHS 171. ICT-palvelujen kehittäminen: Kehittämiskohteiden tunnistaminen. Julkisen hallinnon tietohallinnon neuvottelukunta. Versio 1.1. Saatavissa: <http://docs.jhs-suositukset.fi/jhs-suositukset/JHS171/JHS171.pdf>
- JHS 173. ICT-palvelujen kehittäminen: Vaatimusmäärittely. Julkisen hallinnon tietohallinnon neuvottelukunta. Versio 1.0. Saatavissa: <http://docs.jhs-suositukset.fi/jhs-suositukset/JHS173/JHS173.pdf>
- JHS 179. ICT-palvelujen kehittäminen: Kokonaisarkkitehtuurin kehittäminen. Julkisen hallinnon tietohallinnon neuvottelukunta. Versio 1.0. Saatavissa: <http://docs.jhs-suositukset.fi/jhs-suositukset/JHS179/JHS179.pdf>
- Korpela M. Nigerian practice in computer systems development. A multidisciplinary theoretical framework, applied to health informatics. Doctoral dissertation. Helsinki University of Technology. Department of Computer Science. Reports TKO-A31. s. 273, 1994.
- KuntaIT. Kuntasektorin SOA-tekniikan linjaukset. Versio 1.0, 16.12.2008.
- Laamanen K, Tinnilä M. Prosessijohtamisen käsitteet. 4. painos. Teknologiateollisuuden julkaisu, 2009.
- Mursu A, Ikävalko P. Tietojärjestelmät palvelemaan työtä - Apuna toimintalähtöinen kehittämismalli. Systemityö 1/2007, s. 9-12, 2007.
- Mykkänen J, Pöyhölä A, Toroi T, Riikonen P, Riekkinen A. Palveluarkkitehtuurin soveltaminen terveydenhuollossa - Osa 1: hyödyt, kustannukset, arviointi ja hankinnat. Kuopio: SerAPI-projekti, Kuopion yliopisto, 2007.
- Project Management Institute 2000. Guide to the Project Management Body of Knowledge (PMBOK Guide). Project Management Institute, 2000.
- Simons R. The role of management control systems in creating competitive advantage: New perspectives. Accounting, Organizations and Society. Vol. 15, nro 1/2, s. 127-143, 1990.
- Sprott D. Best Practice Report The Business Case for SOA. CBDI Journal, June, 2006.