

**Tapaustutkimus kansallisen sähköisen arkiston ja viitetietokantaan
perustuvan aluetietojärjestelmän yhteensovittamisesta**

Ilkka Mäki

Tampereen yliopisto
Tietojenkäsittelytieteiden laitos
Tietojenkäsittelyoppi
Pro gradu -tutkielma
Ohjaaja: Pirkko Nykänen
Huhtikuu 2008

Tampereen yliopisto

Tietojenkäsittelytieteiden laitos

Tietojenkäsittelyoppi

Ilkka Mäki: Tapaustutkimus kansallisen sähköisen arkiston ja viitetietokantaan perustuvan aluetietojärjestelmän yhteensovittamisesta

Pro gradu -tutkielma, 75 sivua, 6 liitesivua

Huhtikuu 2008

Tässä tutkimuksessa suoritettiin dokumentaatioaineistoon perustuva case-tutkimus, jossa arvioitiin joidenkin sairaanhoitopiirien käyttämien viitetietokantoihin perustuvien aluetietojärjestelmien mahdollisuuksia integroitua tulevan terveydenhuollon valtakunnallisen tietojärjestelmäarkkitehtuurin kanssa. Valtakunnallisen kokonaisvaltaisen tietojärjestelmäarkkitehtuurin integrointitarpeet koskettavat lähinnä Kansaneläkelaitoksen tilauksesta rakennettavaa kansallista sähköistä terveydenhuollon potilasarkistoa.

Tutkimuksessa etsittiin niin sovellusintegraation kuin palvelukeskeisen arkkitehtuurin tutkimuksista malleja, joilla voidaan kuvata hajautettuja viitetietokantoja, keskitettyä terveydenhuollon valtakunnallista tietojärjestelmäarkkitehtuuria ja näiden kahden tietojärjestelmäarkkitehtuurin sovellusintegraatioprosessia. Etsinnässä löydettiin listoja ja rakenteita, joilla kuvata sovellusintegraation ongelmakenttää tarvittavalla tarkkuudella.

Kirjallisuusanalyysin jälkeen löydettyjä malleja sovellettiin käytännön kohteeseen, tutkimuksen kohdealueeseen ja mietittiin miten kumpainenkin sovellusintegraation osapuoli vastasi näitä teoreettisia malleja. Tämän pohjalta kehitettiin useampia jaotteluryhmiä, joihin niin kansallinen sähköinen arkistopalvelu kuin alueelliset viitetietokannat kuuluivat. Toisaalta kirjallisuudesta löydetyt mallit selittivät reaali maailman ilmiöiden rakennetta, mutta kuten olettaa saattoi, eivät reaali maailman ilmiöt olleet kovinkaan helposti kategorisoitavissa.

Case-tutkimuksen tuloksina mietittiin ratkaisuja, joilla viitetietokannat ja kansallinen arkisto voisivat liittyä yhteen käyttäen palvelukeskeistä arkkitehtuuria. Tutkimusta jatkettiin miettimällä konkreettisia toimintoja, joiden tarpeellisuus ja sijoittelu tarvitsee miettiä tapauskohtaisesti. Näistä voidaan muodostaa monia vaihtoehtoja, joiden mukaisesti sovellusintegraation jälkeinen palvelukokonaisuus voisi toimia.

Lopuksi mietittiin mitä sovellusintegraatio vaatii onnistuakseen, mitkä ovat sen hyödyt, haitat ja tarpeet kohdealueen tietojärjestelmien järjestelyissä. Tätä varten kehitettiin yksinkertaistettu malli, jolla tarpeellisuuskartoitusta voisi mahdollisesti tehdä.

Avainsanat ja -sanonnat: Sovellusintegraatio, terveydenhuollon tietojärjestelmät, palvelukeskeinen arkkitehtuuri, viitetietokannat, kansallinen arkisto.

Sisällys

1.	Johdanto	1
1.1.	Alueellinen viitetietojärjestelmäarkkitehtuuri	2
1.2.	Kansallinen terveydenhuollon tietojärjestelmäarkkitehtuuri.....	4
2.	Tutkimusalue	7
2.1.	Tutkimusongelma	7
2.2.	Tutkimuksen taustaa	7
2.2.1.	Terveydenhuollon tietojärjestelmien käytöstä Suomessa vuonna 2007.....	7
2.2.2.	Terveydenhuollon tietojärjestelmien käytöstä ulkomailla.....	8
2.2.3.	Toiminnallisuutta ohjaavat lait	9
2.2.4.	Aikataulusta	10
2.2.5.	Osapuolet.....	10
2.3.	Tutkimuksen tavoitteet	10
2.4.	Tutkimuksen metodit	11
2.5.	Tapaustutkimuksesta	11
2.6.	Tutkimuksen rakenne.....	11
3.	Tietojärjestelmäarkkitehtuurista	13
3.1.	Sovellusintegraatiosta.....	13
3.1.1.	Sovellusintegraation käsitteistä	13
3.1.2.	Sovellusintegraation jaottelu	13
3.1.3.	Yhteistoiminnallisuuden ongelmat	14
3.1.4.	Sovellusintegraation tavoitemalleja.....	16
3.1.5.	Sovellusintegraation vaiheistuksesta ja kypsyydestä	17
3.2.	Palvelukeskeinen arkkitehtuuri	19
3.3.	Keskitetty vai hajautettu: Tietojärjestelmäarkkitehtuurin taistelukenttä 20	
3.3.1.	Keskitetty tietojärjestelmäarkkitehtuuri	21
3.3.2.	Hajautettu tietojärjestelmäarkkitehtuuri	22
3.3.3.	Hajautetun ja keskitetyn tietojärjestelmäarkkitehtuurin integraatio 22	
4.	Tapaustutkimus KANTA-järjestelmän ja aluetietojärjestelmien yhteensovittamisesta.....	24
4.1.	KANTA-potilastietojen arkistopalvelun määrittelyt	24
4.1.1.	Vaatimusmäärittelyistä	25
4.1.2.	Arkkitehtuurimäärittelyistä.....	26
4.1.3.	Määrittelyiden asettamista vaatimuksista.....	26

4.2.	KANTA-arkkitehtuurin käyttämiä standardeja ja teknologioita	28
4.2.1.	HL7-standardi	28
4.2.2.	Muista standardeista	29
4.2.3.	Palvelukeskeinen arkkitehtuuri KANTA-arkkitehtuurissa	30
4.3.	KANTA-järjestelmän palvelut	30
4.3.1.	Arkistopalvelu	31
4.3.2.	Hakemisto- ja rekisteröintipalvelu	32
4.3.3.	Viestinvälitys	32
4.3.4.	Tunnistus, varmennus ja sähköinen allekirjoitus	33
4.3.5.	Koodistopalvelu	34
4.3.6.	Suostumuksenhallintapalvelu	34
4.3.7.	Loki- ja valvontapalvelu	34
4.3.8.	Kansalaisen katseluyhteys	35
4.4.	Viitetietokannat hajautettuna tietojärjestelmänä	35
4.5.	KANTA-arkkitehtuuri keskitettynä tietojärjestelmänä	36
5.	Tulokset	39
5.1.	Viitetietokantojen ja KANTA-arkkitehtuurin yhdistämisestä	39
5.1.1.	KANTA-arkkitehtuurin ja viitetietokantojen yhdistäminen sovellusintegraation näkökulmasta	40
5.1.2.	Tietojärjestelmien integraation mahdollisuudet palvelukeskeisen arkkitehtuurin näkökulmasta	43
5.2.	Mahdollisia käyttötapoja	44
5.2.1.	Jatketaan kuten ennenkin	46
5.2.2.	Väliaikainen tallennus	47
5.2.3.	Hakutietojen talletus	47
5.2.4.	Kuvantamispalvelut jätetään	47
5.2.5.	Koodistopalvelu	48
5.2.6.	Resursoinnin apuvälineet sairaanhoitopiirin sisällä	48
5.2.7.	Alueellinen toiminnanohjaus ja data warehouse-ratkaisut	48
5.2.8.	Aito integrointi noudattaen KANTA-vaatimuksia	49
5.2.9.	Viitetietokanta aktiivisena toimijana	49
5.3.	Viitetietokantojen tarpeellisuus- ja hyödyllisyyskartoitus	52
6.	Yhteenveto ja pohdintaa	56
	Viiteluettelo	58
	Liitteet	

1. Johdanto

Tässä tutkimuksessa tutustutetaan lukija käytössä oleviin terveydenhuollon sairaanhoitopiirikohtaisiin alueellisiin viitetietojärjestelmiin ja tulevaan terveydenhuollon valtakunnalliseen tietojärjestelmäarkkitehtuuriin ja sen puitteissa toteutettavaan kansalliseen sähköiseen potilastietojen arkistopalveluun. Tästä suunnitellusta arkistopalvelusta ja siihen liittyvästä arkkitehtuurista käytetään yleisnimitystä KANTA ja tätä termiä tullaan käyttämään myös tämän tutkimuksen puitteissa. Varsinaisena tarkoituksena on tehdä tapaustutkimus, jolla arvioidaan näiden tietojärjestelmäkokonaisuuksien yhteensovittamisen mahdollisuuksia. Taustalla yhteensovittamisen ongelmassa on se, että viitetietojärjestelmät perustuvat tiedon hajauttamiseen, kun taas kansallinen tietojärjestelmäpalvelu perustuu monen tietojärjestelmän tiedon keskittämiseen yhteen, valtakunnalliseen tietojärjestelmään.

Viitetietokantojen ja KANTA-tietojärjestelmän tarkoituksena on lisätä sairaanhoidon palveluiden tehokkuutta kehittämällä käytettäviä tietoteknisiä palveluita. Tehokkuuden lisäys on tarpeen terveydenhuollon alalla, jossa painitaan ikääntyvän väestön ja toisaalta määrärahaongelmien ja tehokkuusvaatimusten kanssa. Ongelmat juontavat juurensa terveydenhuollon palveluiden hallinnoinnin ja rahoituksen hajaantuneisuudesta, joka vuorostaan on johtanut tietojärjestelmien heikkoon standardointiin ja yhteistoiminnallisuuteen [Harno *et al.*, 2008]. Toisaalta kuitenkin tutkimukset ovat osoittaneet, että kaikista ongelmista huolimatta juuri toimivilla klinisillä potilasjärjestelmillä on suurin vaikutus terveydenhuollonorganisaatioiden (tarkemmin sairaaloiden) toimintaan [Bhattacharjee *et al.*, 2007]. Kuitenkin näiden klinisten järjestelmien tulisi olla nimenomaan jokapäiväistä käyttöä tukevia järjestelmiä niin, että niistä saadaan todellista mitattavaa hyötyä.

Tietojärjestelmien integraatio ja organisaatiot ylittävä informaation kulku ovat yleisiä trendejä myös kansainvälisesti. Tämä kehityssuunta, yhdistettynä vahvasti asiakaslähtöisyyteen, on saanut kansainvälisen yleisnimikkeen eHealth. Sillä voidaan tarkoittaa ylipäätään terveydenhuollon tietojärjestelmien tutkimusta ja kehittämistä, mutta tällöinkin termi viittaa yleensä kansalaisille tarjottaviin palveluihin, joiden avulla hän voi itse päästä näkemään tietojaan ja vaikuttamaan omaan hoitoonsa. Hallinnon läpinäkyvyys on siis osa tätä kokonaisuutta [Nykänen, 2007] [Commission of the European Communities, 2007].

Seuraavissa johdannon luvuissa käsitellään karkealla tasolla kahden edellämainitun tietojärjestelmäarkkitehtuurin sisältöä niin, että ensiksi kerrotaan viitetietokantoihin perustuvista aluetietojärjestelmistä ja tämän jälkeen taas keskitetystä kansallisesta terveydenhuollon arkistosta.

1.1. Alueellinen viitetietojärjestelmäarkkitehtuuri

Alueellisilla tietojärjestelmäratkaisulla tarkoitetaan usean eri terveydenhuollon organisaation yhteisesti käyttämiä tietojärjestelmiä. Niillä on yleisesti pyritty ratkaisemaan terveydenhuollon ongelmia nimenomaan sairaanhoitopiiritasolla. Tunnistettuja ongelmia ovat mm. palveluiden tehottomuus, tiedonkulun ongelmat organisaatioiden tasolla ja asiakaslähtöisyyden puute. Palveluiden tehottomuus ei niinkään tarkoita puutteita hoidon laadussa, vaan resurssien käytössä ja niiden priorisoinnissa; asiakkaan hoitoa ei ole ollut mahdollista ajatella kokonaisuutena hänen tarpeidensa lähtökohdista [Nykänen ja Karimaa, 2002a] ja toisaalta tiedonkulun ongelmista johtuen priorisoidut tutkimukset ja hoidot ovat voineet perustua väärin olettamuksiin, jäädä tekemättä tai tulla tehtyä useita kertoja. Sen vuoksi tehokkuus tuo myös mukanaan kustannussäästöjä. Yleisenä ratkaisuna näihin ongelmiin on otettu käyttöön ajatus mahdollisimman sujuvasta, asiakaslähtöisestä palveluiden muodostamasta kokonaisuudesta eli palvelukokonaisuudesta. Palvelukokonaisuuden on terminä todettu olevan asiakkaalle ”Yhden tai useamman terveydenhuollon palvelunantajan tuottamien palvelutapahtumien yksilöity kokonaisuus” [L159, 2007]. Se vastaa aiempaa saumattomien palveluketjujen käsitettä kun kyse on terveydenhuollon palveluiden antajien välisestä palvelukokonaisuudesta [STM, 2007e]. Periaate on, että asiakkaiden hoito tapahtuu heidän omista tarpeistaan ja lähtökohdistaan eikä sitä rajoita organisaatorajat tai hoidon tuottavan organisaation tietojärjestelmät tai yleiset tietoresurssit. Kirjallisuudessa löytyy tukea sille, että terveydenhuollon asiakkaiten tyytyväisyys paranee ja toisaalta koko hoitoprosessi tehostuu, mikäli asiakas pääsee itse vaikuttamaan palvelukokonaisuutensa suorittamiseen [Smith, 2007].

Taustalla aluetietojärjestelmien kehityksessä vaikuttaa jo pitempiaikainen kehitys terveydenhuollon tietomäärien kasvusta ja käsittelyn kompleksisuuden lisääntymisestä. Tähän taas on johtanut erikois- ja perusterveydenhuollon erkaneminen (erikoissairaanhoidon voidaan nähdä spesialisoituneen) ja etenkin kansanterveyslain (66/1972) säätäminen ja avoterveydenhuollon laitostaminen terveyskeskuksiin [Ohtonen, 2002]. Nämä toimet yhdessä ovat johtaneet organisaatiotasojen pirstaloitumiseen ja sitä kautta edellä mainittuun, ongelmalliseen tilanteeseen. Näihin ongelmiin vastaamaan perustettiin Satakunnan Makropilotti-hanke, jossa rakennettiin alueellinen viitetietokanta toteuttamaan saumatonta palveluketjun käsitteen mukaista tietojärjestelmäkokonaisuutta. Hanke käynnistyi vuonna 1999 ja sen tuotokset ovat edelleen käytössä Satakunnan sairaanhoitopiirin alueella.

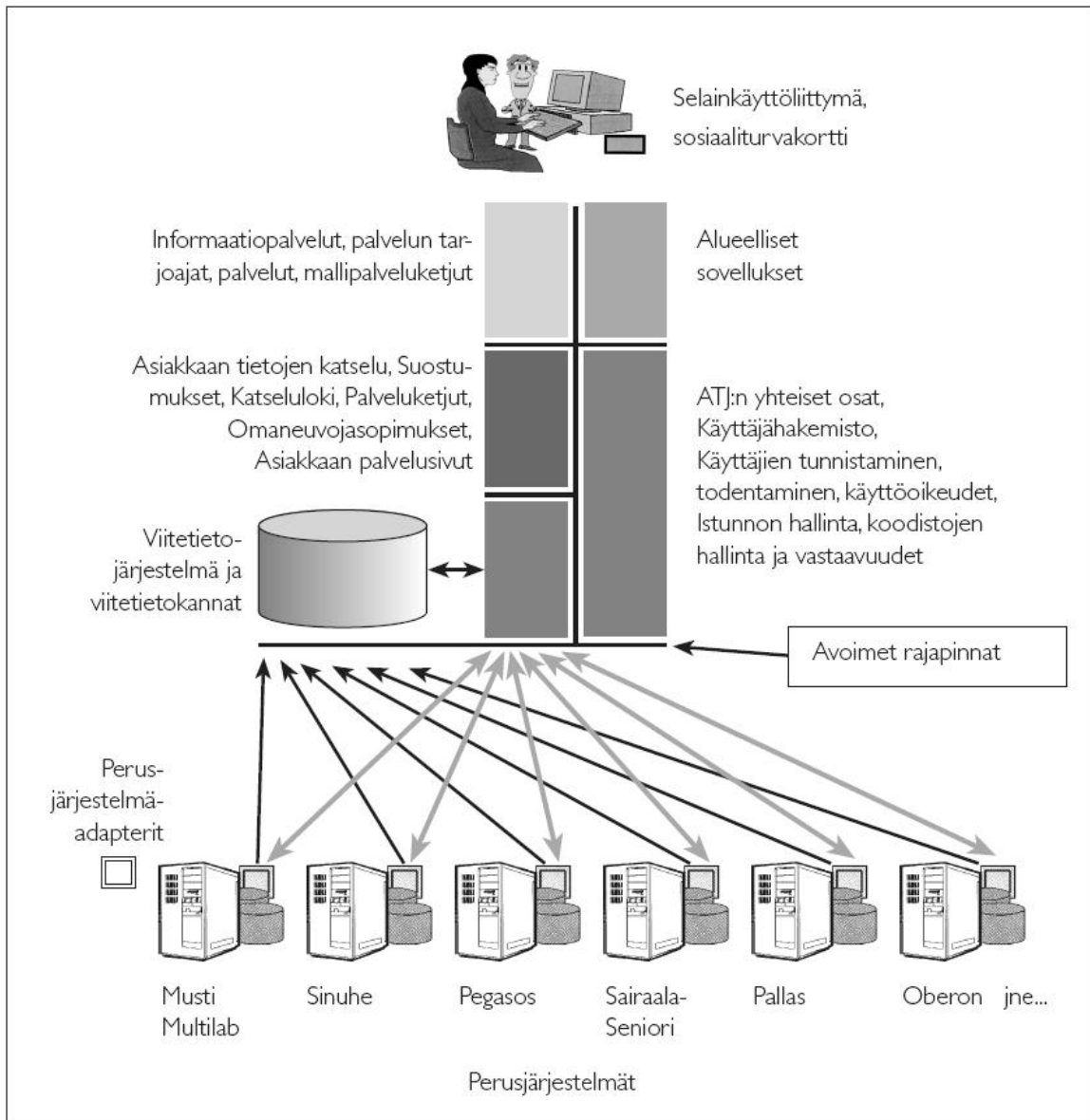
Teknisesti viitetietokannat perustuvat hajautettuun tietojärjestelmäarkkitehtuuriin eli asiakkaan tiedot sijaitsevat perusjärjestelmissä ja niillä on jokaisella yhteys perusjärjestelmäadapterin välityksellä viitetietokantaan, johon on tallennettu viite terveydenhuollon asiakirjoista (viite sisältää tiedon asiakirjan sisällöstä ja sen

sijainnista) ja asiakkaan perustiedoista. Tämän takia mikäli käyttäjällä on asiakkaan suostumus (tietojen hakua tehtäessä suullinen, itse potilasasiakirjoja katsellessa kirjallinen) hän saa nähtäväkseen toisessakin perustietojärjestelmässä olevan asiakirjan kuitenkin niin, ettei hänellä ole mahdollisuuksia tallentaa tai muuttaa sitä [Nykänen ja Karimaa, 2002a]. Perusjärjestelmäadapterit ovat ohjelmistokomponentteja, jotka toisaalta lähettävät luoduista potilastiedoista viitteet viitetietokantaan, mutta myös toisaalta lähettävät viitetietokannan kautta tehdyt käyttäjien potilastietojen kyselyt viitetietokannalle. Viitetietokanta sitten välittää potilastietoja sitä kysyneelle käyttäjälle. Kuvassa 1 on esitetty viitetietojärjestelmään perustuvan Makropilotin osakokonaisuudet.

Satakunnan Makropilottiin kehitetyn viitetietokannan toimitti Fujitsu Services ja siitä käytetään nimeä Fiale. Sen lisäksi myös muille sairaanhoitopiireille on tullut käyttöön viitetietokantoja; Sekä Varsinais-Suomen että Pirkanmaan sairaanhoitopiirit ovat ottaneet käyttöön Fiale-järjestelmän ja toisaalta Uudenmaan sairaanhoitopiirissä on käytössä Elisa oy:n kehittämä Navitas-viitetietokantajärjestelmä [Nykänen, 2007]. Sittemmin Pirkanmaan sairaanhoitopiirin kunnat ovat luopuneet Fiale-viitetietojärjestelmän käytöstä helmikuussa 2007, mutta eivät ole ottaneet korvaavaa tuotetta tilalle, vaan odottelevat kansallisen sähköisen arkiston käyttöönottoa.

Makropilotissa käyttöön otetusta toimivasta ja yleisestä terveydenhuollon organisaatiotasot ylittävästä palvelusta voidaan mainita esimerkiksi lähete-palauteratkaisut, joissa perusterveydenhuollon organisaatio antaa potilaalle lähetteen erikoissairaanhoidon palveluita tarjoavalle, toiselle organisaatiolle. Tämä organisaatio taas tekee tilatun palvelun ja antaa palautteen työstään lähetteen antajalle. Tilattu palvelu voi olla niin röntgen- tai laboratoriotutkimus, leikkaus kuin vaikka konsultaatiopalvelu. Tarkoituksena on, että asiakkaan tiedot ovat löydettävissä asiakkaan suostumuksella siellä, missä niitä tarvitaan.

Alueellisille tietojärjestelmille on kehitetty myös muita ratkaisumalleja viitetietokantojen lisäksi, mutta tässä tutkimuksessa keskitytään arvioimaan nimenomaan viitetietokantoihin perustuvien aluejärjestelmien ja KANTA-arkkitehtuurin mahdollisia yhteistoiminnan malleja. Muut ratkaisut perustuvat erilaisiin integroitiratkaisuihin organisaatioiden potilastietojärjestelmien kesken.



Kuva 1. Aluetietojärjestelmän osakokonaisuudet [Itälä, 2000] [Virtanen, 2000].

1.2. Kansallinen terveydenhuollon tietojärjestelmäarkkitehtuuri

Vaikka Suomessa tällä hetkellä 95% potilasvirroista tapahtuukin aluetason sisällä, on suuntaus kuitenkin kohti valtakunnallisia palveluita. Tähän ajavat potilaiden suurempi liikkuvuus, lisääntyvä palveluiden osto (hoitoja ja konsultaatiota) toisista sairaanhoitopiireistä ja yksityiseltä sektorilta ja toisaalta myös Kelan ja vakuutuslaitosten tarpeet tietojen saatavuudesta ja siirrettävyydestä [STM, 2006].

Kansallisella terveydenhuollon sähköisellä tietojärjestelmällä pyritään ratkaisemaan juuri niitä samoja ongelmia, joiden ratkaisuun alueelliset viitetietokannatkin ovat kehitetty. Erona on fokuksen laajentaminen kansalliselle tasolle. Niinpä Sosiaali- ja terveysministeriö on käynnistänyt yhteishankkeen, jossa on määritelty terveydenhuollon valtakunnallinen tietojärjestelmäarkkitehtuuri ja toteutetaan kansallinen sähköinen potilastietojen arkistopalvelu. Tästä arkistopalvelusta käytetään

yleisnimitystä KANTA, mutta itse arkistopalvelutietojärjestelmän lisäksi termi viittaa myös sen ympärille liittyvään infrastruktuuriin ja viestinvälitysrajapintoihin.

KANTA-arkkitehtuuri toteutetaan keskitettynä kansallisena tietojärjestelmäpalveluna käyttäen palvelukeskeistä arkkitehtuuria [STM, 2007a], joten se rakennetaan pienistä itsenäisistä palveluista, jotka löyhästi liitetään toisiinsa. Osa näistä palveluista on julkisia ja osa sisäisiä, ts. ei ulospäin näkyviä.

Kansallisella arkistokomponentilla on julkisina palveluina potilasasiakirjoille seuraavat palvelut:

- Asiakirjojen arkistointi ja rekisteröinti
- Asiakirjojen haku
- Asiakirjojen käyttö ja luovutus.

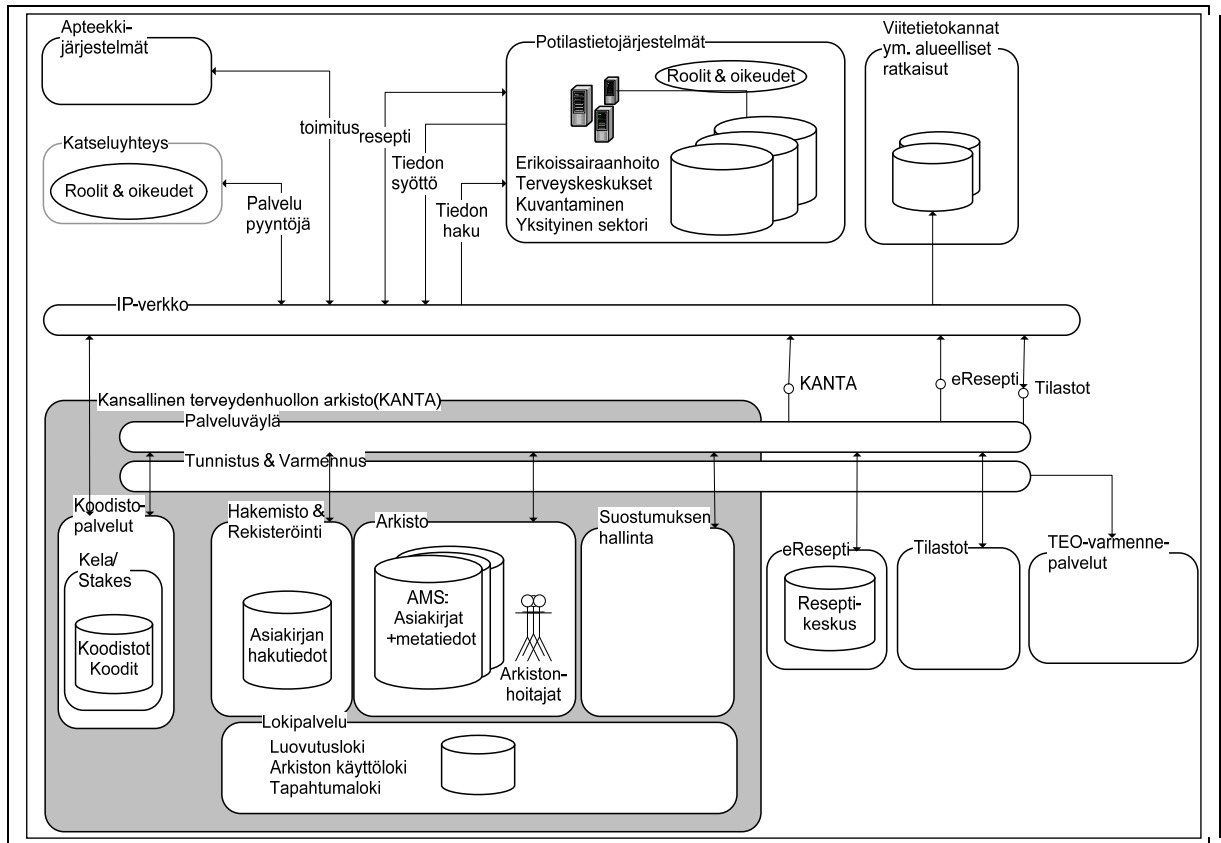
Nämä palvelut ovat KANTA-arkkitehtuuriin liittyneillä terveydenhuollon tietojärjestelmillä käytettävissä.

Lisäksi ulkoiseksi palveluksi voidaan ajatella arkistojärjestelmään luotava suora katseluyhteys kansalaisen omiin tietoihin, niin potilas- kuin lokitietoihinkin.

Yksinkertaistaen kansallisen arkistopalvelun tarjoamat palvelut voidaan nähdä ohjelmistokomponentteina. Tässä komponentti ei tarkoita teknologista ohjelmistokomponenttia joita voisi toteuttaa esim. CORBA:lla tai Microsoftin .NET-teknologialla, joista kokonaisuus muodostuu. KANTA-arkkitehtuurin komponentit/sisäiset palvelut ovat:

1. Arkistopalvelu
2. Hakemisto- ja rekisteröintipalvelu
3. Viestinvälitys
4. Tunnistus, varmennus ja sähköinen allekirjoitus
5. Koodistopalvelu
6. Suostumustenhallinta
7. Loki- ja valvontapalvelu.

Jokainen näistä muodostaa oman kokonaisuutensa ja sen vuoksi niistä on myös jokaisesta kirjoitettu oma määrittelynsä täydentämään kokonaisarkkitehtuurin määrittelyä. Kuvassa 2 näkyy kokonaisuudessaan terveydenhuollon kansallinen arkkitehtuuri niin kuin se on kuvattu määrittelyissä [STM, 2007b].



Kuva 2: Kansallinen terveydenhuollon arkkitehtuuri [STM, 2007b]

Arkistopalvelun lisäksi samaan aikaan ja käyttäen samaa viestinvälitysarkkitehtuuria kuin KANTA:kin tehdään myös palvelu sähköiselle lääkemääräykselle eli eResepti-palvelu, jossa rakennetaan järjestelmä reseptien sähköiselle välittämiseksi apteekkeille. Yhdessä KANTA-palvelu ja eResepti muodostavatkin Kelan vastuulla olevan Kansallinen Toimija-palveluklusterin (lyhennetään muotoon KanTo) [STM, 2007d]. Tämä eResepti-palvelu jätetään kuitenkin tämän tutkimuksen ulkopuolelle.

2. Tutkimusalue

2.1. Tutkimusongelma

Tutkimuksen tarkoituksena on selvittää, mitä käyttötarkoituksia viitetietokantoihin perustuvilla aluetietojärjestelmillä voi olla, kun kansallinen sähköinen terveydenhuollon arkisto tulee käyttöön. Teoreettiselta kannalta kyseessä on keskitetyn (KANTA) ja hajautetun (aluetietojärjestelmät) tietojärjestelmän yhteenliittämisestä. Ongelman haastavuutta lisää se, että keskitetty, kokoava toimija on tässä tapauksessa hierarkisesti hajautetun aluetietojärjestelmän yläpuolella. Vaatimusmääritysten mukaan aluetietojärjestelmän saa liittää KANTA-palveluun [STM, 2007a], jos se osaa toimia läpinäkyvästi siten, että KANTA näkee sen terveydenhuollon perusjärjestelmänä ja toisaalta aluejärjestelmään liitetyt perusjärjestelmät näkevät vain KANTA-palvelun, ainakin niiltä osin kuin mitä tässä tapauksessa liittyy potilastietojen hakuun, luovutukseen ja käyttöön.

2.2. Tutkimuksen taustaa

Seuraavissa luvuissa käsitellään lyhyesti tähän tutkimukseen liittyviä taustoja ennen KANTA-arkkitehtuurin julkaisua. Ensiksi kerrotaan hiukan yleisestä terveydenhuollon tietojärjestelmien tilasta Suomessa. Tämän jälkeen esitellään tilannetta ulkomailla ja lopuksi kerrotaan niistä realiteeteista, joiden puitteissa KANTA-kehitystä tehdään.

2.2.1. Terveydenhuollon tietojärjestelmien käytöstä Suomessa vuonna 2007

Suomessa on terveydenhuollon tietojärjestelmien palveluissa paljon kehitettävää. Tietojärjestelmälähtöiset ongelmat ovat yleisiä, mutta huomiota on viime aikoina kiinnitetty myös siihen, että vaikka tietojärjestelmät toimisivatkin, ei niistä saada odotettua apua tehokkuuteen, mikäli itse palveluprosessit ovat tehottomia eivätkä organisaatioiden nykyiset rakenteet tue tietojärjestelmien optimaalista käyttöä [Harno, 2007]. Kansallisella tasolla informaatioteknologian tuomia, oletettuja hyötyjä ei voida nähdä vielä täysin saavutetun, olkoonkin että rahalliset panostukset ovat olleet pieniä suhteessa koko terveydenhuollon budjettiin (noin 2%) [Harno *et al.* 2008].

Sähköiset terveydenhuollon tietojärjestelmät ovat yleistyneet vauhdilla ja varsinkin sähköinen potilaskertomus oli vuonna 2007 käytössä lähes kaikissa terveydenhuollon palveluja tarjoavissa yksiköissä [Hallitus, 2006]. Erityisesti yleisiä olivat potilaskertomuksen, laboratorion ja ajanvarauksen tietojärjestelmät niin julkisella kuin yksityisellä puolella.

Yksikkökohtaisten tietojärjestelmien välille on viime vuosina rakennettu yhteyksiä niin, että varsinaisia alueellisia tietojärjestelmäratkaisuja oli vuonna 2007 käytössä jo kolmessa neljästä sairaanhoitopiiristä. Lähinnä käytetyin palvelumuoto on lähete-palaute-järjestelmä, jossa potilaan lähetteet ja niistä johtuneet hoidot ylittävät

palveluyksiköiden rajat. Lisäksi muita potilastietoja jaellaan erinäisissä mittakaavoissa [Hallitus, 2006]. Nämä alueelliset viitetietokannat on rakennettu aina tietyn sairaanhoitopiirin toimiyksiköiden väliseen tiedonsiirtoon, mutta sairaanhoitopiirien rajat ylittäviä ratkaisuja on ollut toistaiseksi vain vähän.

Yleisenä ongelmana on tähän saakka ollut teknisten haasteiden (esim. tietoliikenneyhteydet, tietoturva, järjestelmärajapinnat) ja yksiköiden tietosisällön vastuukysymysten lisäksi ongelmat käytössä olevien tietojärjestelmien käsitteistön ja tietosisällön yhtenäisyyden kanssa. Terveystietojärjestelmien ei välttämättä ole ollut esimerkiksi käytössä samoja koodistoja tai käsitteet kuten asiakirja ovat voineet tarkoittaa semanttisesti eri asioita. Lisäksi ongelmia on ollut asiakirjojen rakenteellisuudessa, toisin sanoen sähköisillä potilaskertomuksilla ei ole aikaisemmin ollut standardoitua rakennetta. Sähköisille potilaskertomuksen asiakirjoille on kuitenkin olemassa maailmalla useita standardoituja malleja [Eichelberg *et al.*, 2005], joista Suomen terveydenhuollossa lähes ainoana on yleistynyt HL7-standardi. Näin ollen on luontevaa, että KANTA-palvelussa käytettäväksi asiakirjapohjaksi on valittu HL7-standardin kuvaama CDA R2-rakenne (CDA, Clinical Document Architecture). HL7-standardin laaja levinneisyys voidaan nähdä etuna Suomessa terveydenhuollon tietojärjestelmien yhteistoiminnallisuutta rakennettaessa, sillä asiakirjarakenteiden moninaisuus on nähty suurena uhkana sähköisten potilasasiakirjojen jakelun ja yhteistoiminnallisuuden kehittämisessä [Grimson *et al.*, 2000] [Hristidis *et al.*, 2006]. Vaikka koodistoja on useita ja käsitteissä on tietojärjestelmien ja organisaatioiden välisiä eroja, tarjoaa saman viestistandardin käyttö oletusarvoisesti mahdollisuuden organisaatioiden sovelluksille keskustella keskenään, vaikka HL7 onkin vain yksi pala organisaatioiden tiedonsiirron palapelissä.

2.2.2. Terveystietojärjestelmien käytöstä ulkomailla

Myös monissa muissa maissa ollaan yleisen teknologisen kehityksen myötä ottamassa käyttöön kansallisia sähköisiä terveydenhuollon järjestelmiä. Ongelmat terveydenhuollossa ovat hyvin pitkälle samoja kuin Suomessa huolimatta siitä, kuinka sosiaali- ja terveydenhuolto on ylipäättään järjestetty [Grimson *et al.*, 2000] [Sousa *et al.*, 2006]. Ongelmat kuten hajaantuneet markkinat, liian pienet investoinnit, yhteisten käytössä olevien standardien puutteet vaivaavat niin vahvasti julkiseen terveydenhuoltoon nojaavaa Suomea kuin terveydenhuollon yksityispainotteisesti hoitavaa USA:ta. Tavoitteina kaikissa kehitysprojekteissa on kansalaisten terveyden parempi hallinta informaatioteknologian avulla, tiedon ympärivuorokautinen saatavuus paikasta riippumatta sekä korkealuokkainen yksityisyyden suoja ja tietoturva [Hallitus, 2006] [Grimson *et al.*, 2000]. Lähes jokaisessa EU-maassa ja monissa teollisuusmaissa on jonkunlainen strategia terveydenhuollon sähköisten potilaskertomusten saatavuuden parantamiseksi kansallisella tasolla. Vaikka tavoitteet ovatkin yhteneviä, on suunnitelmissa joitain eroja. Vaikka esimerkiksi yksityisyyden suoja ja tietoturva on

nostettu kaikissa hankkeissa tärkeään asemaan, on eroja kuitenkin siinä, käytetäänkö palvelussa julkista internetiä vai tehdäänkö terveydenhuollon toimijoille oma, suljettu verkko. Myös yleisessä arkkitehtuurissa on eroja, esimerkiksi osassa kansallisista projekteista lähdetään liikkeelle hajautetusta arkkitehtuurista, toisin sanoen arkistot sijaitsevat jokaisessa toimipaikassa ja kansallinen ratkaisu tarjoaa vain mahdollisuuden hakea tieto nähtäville. Toisissa taas noudatetaan samaa keskitettyä periaattetta kuin KANTA-järjestelmässä.

Eräs laajimpia ja kunnianhimoisimpia näistä kansainvälisistä projekteista on varmasti Iso-Britannian kansallisen terveydenhuollon (NHS) ajama kansallinen ohjelma (NpfiT), jolla pyritään saamaan kansallinen terveydenhuollon tietotekninen infrastruktuuri vuoteen 2010 mennessä [Hallitus, 2006]. Luotavan tietojärjestelmän toimintaperiaatteet ovat hyvin samanlaisia kuin Suomen KANTA-järjestelmässä, senkin ollessa vahvasti valtiollisessa ohjauksessa toimiva, keskitetty potilasasiakirjojen tietokanta. NpfiT:n tietojärjestelmäarkkitehtuuri on vielä kunnianhimoisempi kuin KANTA, siinä kun on lisäksi erillinen, suojattu sähköpostiyhteys terveydenhuollon ammattihenkilöille ja ajanvaraustoimintojen tarjoaminen potilaille [Avison and Young, 2007]. Toki tietojärjestelmän mittakaava jo pelkän Iso-Britannian asukasmäärän kautta nousee massiiviseksi. Projekti on ollut käynnissä jo vuodesta 2002 ja myös sitä edeltäviä projekteja löytyy. Kehitystyö on ollut varsin työlästä ja kallista, mutta tavoitteisiin pääsemisellä on kuitenkin ollut erittäin suuri valtiovallan tuki Iso-Britanniassa. Käytettävä rahasumma-arvio, 12 – 20 miljoonaa puntaa seuraavien vuosien aikana, nostaa sen yhdeksi suurimmista terveydenhuollon tietoteknisistä kehitysprojekteista [Avison and Young, 2007].

2.2.3. Toiminnallisuutta ohjaavat lait

Kansallisen terveydenhuollon tietojärjestelmäarkkitehtuurin kehitys- ja määrittelytyön ohessa on valmisteltu myös kehitystä ohjaavaa lainsäädäntöä. KANTA-kehityksen taustalla vaikuttaakin nyt kaksi varsin tuoretta lakia; laki sosiaali- ja terveydenhuollon asiakastietojen sähköisestä käsittelystä (159/2007) ja laki sähköisestä lääkemääräyksestä (61/2007). Toisaalta laissa sosiaali- ja terveydenhuollon asiakastietojen sähköisestä käsittelystä todetaan, että sen määräämä potilasasiakirjan käsittely noudattaa aikaisemmin säädettyjä lakeja. Tarkemmin ottaen tämä koskee lakia potilaan asemasta ja oikeuksista (785/1992), lakia sosiaalihuollon asiakkaan asemasta ja oikeuksista (812/2000), henkilötietolakia (523/1999), lakia viranomaisten toiminnan julkisuudesta (621/1999), lakia sähköisestä asioinnista viranomaistoiminnassa (13/2003), lakia sähköisistä allekirjoituksista (14/2003) ja arkistolakia (831/1994).

Alueellisten viitetietojärjestelmien toiminnallisuutta varten luotiin vuonna 2000 voimaan tullut laki sosiaali- ja terveydenhuollon saumattoman palveluketjun kokeilusta (811/2000). Alun perin laki säädettiin kestäväksi vuoden 2003 loppuun [Ohtonen, 2002], mutta sitä on jatkettu aina eteenpäin niin, että vuoden 2007 lainmuutoksen

(160/2007) myötä voimassaoloaikaa on jatkettu vuoteen 2011 [L811, 2000], jotta lain nojalla toteutettuja aluetietojärjestelmäkokeiluja voitaisiin jatkaa, kunnes siirrytään valtakunnallisen käsittely- ja arkistointipalvelujen käyttöön [STM, 2007k].

2.2.4. Aikataulusta

Aikataulu kansallisen terveydenhuollon tietojärjestelmäarkkitehtuurin käyttöönotolle on varsin tiukka. eResepti-hanke pitäisi olla käytössä jo vuoden 2008 aikana [L61, 2007]. Sähköinen potilasarkisto pitää olla valmiina ja kaikkien sähköisesti potilastietoa tallentavien palvelun antajien (niin julkisten kuin yksityistenkin) pitää olla liittyneitä tähän järjestelmään viimeistään 1. päivä toukokuuta vuonna 2012 [L159, 2007].

2.2.5. Osapuolet

Varsinaisen kansallisen arkistopalvelun toteuttajana ja ylläpitäjänä toimii kansaneläkelaitos. Määrittelyprojektissa vetovastuu on sosiaali- ja terveysministeriöllä, joka hallinnoi myös käydyt tarjouskilpailut niin määrittelyprojektista kuin itse toteutusprojektistakin. Määrittelyprojekti annettiin WM-Datan johtamalle konsortiolle, johon kuului myös Medici Data Oy, Commit Oy, Avain Technologies Oy ja CSW Limited [STM, 2007f]. Varsinaisen toteutusprojektin tarjouskilpailun voitti Fujitsu Services Oy, jonka kanssa Kela aloitti sopimusneuvottelut arkistopalvelun toteuttamisesta [Kelan tiedote, 2007].

Muina palveluita tarjoavina osapuolina toimivat mm. Sosiaali- ja terveysalan tutkimus- ja kehittämiskeskus (Stakes) jonka tehtävänä on huolehtia koodistopalvelusta ja Terveydenhuollon oikeusturvakeskus (TEO) joka huolehtii sähköisistä allekirjoituksista ja varmenteista.

Oleellisimpina osapuolina toimivat kaikki ne terveydenhuollon järjestelmät, jotka KANTA:an liittyvät; Ne kuitenkin ovat varsinaisia tiedon tuottajia. Nämä liittyvät perusjärjestelmät edustavat niin yksityisiä lääkäriasemia kuin julkisia sairaaloita ja terveyskeskuksia. Lisäksi kansalliseen järjestelmään voivat liittyä myös sairaanhoitopiirien omat alueelliset viitejärjestelmät. Tällöin viitetietojärjestelmä toimisi kuten yksittäinen terveydenhuollon yksikkökin samoine toiminnallisuuksineen. Tulevaisuudessa on myös mahdollista, että järjestelmään liittyy myös organisaatioita jotka eivät liity sosiaalitoimen tai terveydenhuollon toimialaan kuten vakuutusyhtiöitä tai tilastoja kerääviä valtiollisia organisaatioita.

2.3. Tutkimuksen tavoitteet

Tämän tutkimuksen tavoitteena on perehtyä hajautettujen ja keskitettyjen tietojärjestelmien muodostamien kokonaisuuksien yhteenliittämisen mahdollisuuksiin. Tavoitteena on tutkia asiaa sekä teoreettiselta kannalta että käyttäen tapaustutkimuksen kohteena mahdollista KANTA-arkkitehtuurin ja viitetietokantoihin perustuvien

aluejärjestelmien yhteen liittämistä. Tapaustutkimuksessa mietitään teorioiden pohjalta ratkaisuja, miten voitaisiin edetä todellisessa ongelmassa.

2.4. Tutkimuksen metodit

Päämetodinä tässä tutkimuksessa käytettiin tutustumista julkaistuihin dokumentteihin ja määrityksiin. Pohjan tutkimukselle muodosti julkaistujen KANTA-arkkitehtuurin määritysten analysointi. Määrityksiä peilattiin varsinaiseen tutkimusongelmaan ja kirjallisuudesta löytyviin siihen liittyviin artikkeleihin. Kirjallisuudesta etsittiin sekä yleisiä artikkeleita sovellusintegraatiosta että viittauksia erilaisilla tietojärjestelmän rakentamisen metodeilla toteutettujen sovellusten integraatiosta. KANTA-arkkitehtuurin määrityksiä vertailtiin viitetietokannoista löydettyihin dokumentteihin tarkoituksena löytää mahdollinen yhteyspinta. Perustuen kirjallisuuteen ja kummastakin tietojärjestelmästä kerättyyn tietoon mietittiin ratkaisuja yhteistoiminnallisuuden toteutukseen.

2.5. Tapaustutkimuksesta

Tämä tutkimus perustuu tapaustutkimuksen metodiin niin, että aihealueen kirjallisuutta ja teorioita tutkitaan yleisesti ja pyritään arvioimaan niiden toteutumista tässä case-tapauksessa. Toisin sanoen, tutkimuksessa pyritään soveltamaan ja vertailemaan saatua tietoa reaali maailman ilmiöön eli tässä tapauksessa mahdolliseen sovellusintegraatioon hajautettujen alueellisten viitetietokantojen ja keskitetyn KANTA-arkkitehtuurin välillä. Tutkimus on Järvisen ja Järvisen [2004] esittelemä teorioita testaava case-tutkimus. Tutkimuksen rakenteena käytetään hiukan sovellettua Järvisen ja Järvisen [2004] teorioita testaavan tutkimuksen raportin rakenteen mallia.

2.6. Tutkimuksen rakenne

Luvussa 1 eli johdannossa esitellään lyhyesti tutkimuksen aihe ja kartoitetaan taustaa, johon tutkimus perustuu. Siinä myös esitellään lyhyesti tapaustutkimuksen sovellusintegraation osapuolet.

Luvussa 2 käydään läpi tutkimuksen ongelmatilannetta ja taustaa tarkemmin. Tapaustutkimuksen ongelmatilanteeseen johtaneet taustat kartoitetaan ja toisaalta kerrotaan vertailun vuoksi esimerkkejä myös muiden maiden vastaaviista tilanteista. Tutkimusalueen taustoja kerrotaan myös mm. lakitekstin ja tavoitteiden osalta.

Luku 3 käsittelee tutkimuksen teoreettista taustaa. Luvussa kartoitetaan kirjallisuutta tämän tapaustutkimuksen laajemman kontekstin eli sovellusintegraation ja sen mallien osalta. Lisäksi annetaan lyhyt katsaus palvelukeskeiseen tietojärjestelmäarkkitehtuuriin ja kahteen tapaustutkimuksen eri tietojärjestelmäparadigmaan, keskitettyyn ja hajautettuun tietojärjestelmäarkkitehtuuriin.

Luvussa 4 käydään varsinaista tapaustutkimuksen kohdetta läpi. Käsittely aloitetaan kansallisen arkkitehtuurin määräyksistä ja kerrotaan standardeista ja suunnitteluparadigmoista joita sen kehittämisessä on käytetty. Lukua jatketaan kertoen KANTA-järjestelmän eri palveluista siltä osin kuin ne nähdään tämän tutkimuksen osalta tarpeelliseksi. Lopuksi kerrotaan siitä, kuinka viitetietokannat edustavat hajautettua tietojärjestelmäarkkitehtuuria ja toisaalta, kuinka KANTA-järjestelmä edustaa keskitettyä tietojärjestelmäarkkitehtuuria.

Luvussa 5 kerrotaan tapaustutkimuksen tuloksista. Viitetietokantojen ja KANTA-arkkitehtuurin yhdistämisen mahdollisuuksia analysoidaan sovellusintegraation näkökulmasta. Luvussa mietitään myös KANTA-integraation mahdollisia vaikutuksia aluejärjestelmien organisaatioon ja toisaalta palvelukeskeisen arkkitehtuurin tarjoamia apuvälineitä sovellustason integraatioon. Tämän jälkeen mietitään konkreettisia käyttötapoja viitetietokannalle sovellusintegraatioprosessin läpi käyneissä organisaatioissa. Vaikka ehdotukset ovat konkreettisia, ne on silti kuvattu vain karkealla tasolla. Lopuksi mietitään vielä viitetietokantojen tarpeellisuus- ja hyödyllisyyskartoitusta, jolla järjestelmävastaavat voisivat arvioida omien viitetietokantojensa mahdollista tarvetta tulevaisuudessa.

Viimeinen luku eli luku 6 sisältää yhteenvedon tutkimuksen tuloksista sekä tulosten merkityksen pohdintaa.

3. Tietojärjestelmäarkkitehtuurista

3.1. Sovellusintegraatiosta

Tässä luvussa tutustutaan sovellusintegraatioon niin käsitteenä kuin siitä tehtyjen tutkimusten muodossa. Sovellusintegraation käsite esitellään lyhyesti, kuten myös siihen liittyviä lähestymistapoja ja ongelmakohtia. Lisäksi esitellään myös sovellusintegraation tavoitemalleja ja integraatioprosessin vaiheistusta.

3.1.1. Sovellusintegraation käsitteistä

Sovellusintegraatio terminä on varsin epämääräinen. Tässä luvussa kerrotaan käsitteen sisällöstä niistä useista lähtökohdista, joita sille voidaan kirjallisuudesta löytää. Kirjallisuuskartoituksessa tutkitaan termin käyttöä nimenomaan terveydenhuollon tietojärjestelmien lähtökohdista.

Sovellusintegraatio käsitteenä tarkoittaa useamman kuin yhden sovelluksen yhteistoiminnallisuutta niin, että ne osaavat jakaa, lukea ja käyttää toisiltaan saamaansa tietoa tai palveluita. Tarkempi sovellusintegraation alueen käsitteistön määrittelyn analysointi jätetään tässä tarkastelun ulkopuolelle. Käsitteitä ovat pohtineet muun muassa Vasconcelos *et al.* [2007] ja Siltanen [2004]. Tutkimuksessa huomioidaan nimenomaan sovellusintegraation tietojärjestelmätasolla, toisin sanoen integraatioprosessissa on mukana pelkkien sovellusten lisäksi myös niiden käyttäjät, heidän toimintatapansa kuin se tietotekninen laitteisto, jossa sovellukset toimivat.

3.1.2. Sovellusintegraation jaottelu

Sovellusintegraation voi jakaa monella tavalla erilaisiin alikäsitteisiin. Linthicum [2003] jakaa sovellusintegraation käytännönläheisesti neljään eri ryhmään sen suuntautumisen mukaan. Havaitut ryhmät hänen mukaan on:

- Tieto-orientoitunut
- Liiketoimintaorientoitunut
- Palveluorientoitunut
- Portaaliorientoitunut

Näistä tieto-orientoitunut näkökulma on yleisin lähestymistapa sovellusintegraatiota tutkivassa kirjallisuudessa. Tämä on ymmärrettävää, sillä sovellusten tietojen sisällön merkitykset ovat oleellinen osa koko niiden toimintaa ja sitä kautta tietokokonaisuuksien merkitysten yhteensovittaminen sovellusten välisessä integraatiossa koko sovellusintegraation perusta. Muut Linthicumin [2003] mallin esittämät orientaatiot ovat enemmän käytännönläheisiä. Näistä liiketoimintaorientoitunut malli ajattelee sovellusintegraatiota yritysten liiketoimintaprosessin kannalta kokonaisvaltaisesti niin, että integraation tuotos on yritysten liiketoimintaprosessien yhdistyminen. Liiketoimintaorientoitunut integraatio

voi koostua useasta perinteisestä sovellusintegraation läpikäyneestä sovelluskokonaisuudesta. Esimerkkinä voidaan ajatella yritysten yhteiskäyttöisiä toimitusketjunhallinnan tietojärjestelmiä eli SCM-tietojärjestelmiä (SCM, Supply Chain Management). Tässä ratkaisussa samassa tuotantoketjussa samoja sovellusintegraation läpikäyneitä tietojärjestelmiä voi siis käyttää monta yritystä. Palveluorientoitunut sovellusintegraatio taas perustuu palvelukeskeisen tietojärjestelmäarkkitehtuurin ajatukselle, jossa käyttäjä käyttää tietojärjestelmää, joka koostuu useista yhteen liitetystä palveluista. Portaaliorientoituneessa mallissa yhden suoran käyttöliittymän taakse on koottuna useita eri palveluita ja järjestelmiä. Linthicum [2003] tunnistaa sen eroavan esimerkiksi palveluorientoituneesta mallista siinä, ettei tässä kyse ole niinkään tosiaikaisesta ja tapahtumakeskeisestä toiminnasta vaan esimerkiksi eräajotyypisistä ratkaisuista, joita vaan hoidetaan yhden portaalin ja käyttöliittymän läpi. Linthicumin [2003] jaottelu on varsin erilainen kuin mitä muussa kirjallisuudessa esiintyy, koska siinä ajatellaan asioita enemmänkin sovellusintegraation läpikäyneiden organisaatioiden ja niiden tietojärjestelmien näkökulmasta, eikä niinkään itse sovellusintegraatioprosessin näkökulmasta.

Toinen, kirjallisuudessa yleisempi jaottelu perustuu enemmän tietoon. Siinä karkealla tasolla sovellusintegraation kenttä voidaan myös jakaa kahteen osaan, organisaatioiden sisäiseen ja ulkoiseen sovellusintegraatioon. Terveystieteiden tutkimuksissa on perehdytty näihin molempiin sovellusintegraation lajeihin. Sisäisestä sovellusintegraatiosta esimerkkinä voidaan mainita muun muassa HL7-organisaation CCOW- eli työpöytäintegraation [Hristidis, 2006], jossa on pyritty optimoimaan useiden eri sovellusten käyttöä samalle käyttäjälle. Se ei kuitenkaan mahdollista varsinaista datan siirtoa käyttäjien kesken, joten sitä ei voi pitää automaattisena ratkaisuna sovellusintegraation pulmaan organisaation sisäisessä kontekstissa. Tämän tutkimuksen lähtökohdissa keskitytään kuitenkin lähinnä organisaatioiden väliseen sovellusintegraatioon, toisin sanoen organisaatioväliseen sovellusten yhteistoiminnallisuuteen (englanniksi interoperability). Sovellusintegraatiossa monet teoriat kuitenkin pätevät huolimatta siitä, peilaako niitä sisäiseen vai ulkoiseen sovellusintegraatioon.

3.1.3. Yhteistoiminnallisuuden ongelmat

Sovellusintegraatiota voidaan lähestyä, riippumatta siitä, onko se organisaation sisäistä vai organisaatioiden välistä, joko puhtaasti data ja alustalähtöisesti tai enemmän prosessilähtöisesti huomioiden myös muut kuin tekniset lähtökohdat [Linthicum, 2003] [Stumptner *et al.*, 2004]. Prosessilähtöinen lähestymistapa keskittyy enemmän analysoimaan toimialaympäristöä ja käytäntöjä miettien integraatiota joko enemmän reaali maailman tapahtumana [Linthicum, 2000] [Mendoza *et al.*, 2006] tai toiminnallisena (behavioristisena) ongelmana [Stumptner *et al.*, 2004], eikä niinkään

teknisenä ja tiedollisena ongelmana. Linthicum [2003] taas määrittelee lähestymistapansa enemmänkin organisatorisina malleina.

Oli lähestymistapa mikä tahansa, on kirjallisuudessa yleisimpiä tutkimuksen aiheita järjestelmien yhteistoiminnallisuus keskenään. Yhteistoiminnallisuudesta tietotasolla on tehty useita jakoja kirjallisuudessa, mutta periaatteessa karkea jako tehdään semanttiseen ja syntaktiseen [Park and Ram, 2004] [Beyer *et al.*, 2004]. Syntaktinen yhteistoiminnallisuus käsittää sovellus- ja käyttöjärjestelmätasot niin, että sovellukset pystyvät toimimaan keskenään riippumatta käytetyistä käyttöjärjestelmistä, ohjelmointikielestä tai rajapinnoista. Syntaktisen yhteistoiminnallisuuden ongelma ei enää ole yhtä suuri kuin aikaisemmin. Tähän muutokseen ovat olleet vaikuttamassa komponenttipohjaisen ohjelmistokehityksen yleistyminen ja erityisesti palvelukeskeisen tietojärjestelmäarkkitehtuurin ja web-pohjaisten, hajautettujen ratkaisujen yleistyminen [Park ja Ram, 2004] [Matsunaga *et al.*, 2007].

Yleisesti kirjallisuudessa tunnustetaan myös semanttisesta yhteistoiminnallisuudesta eri tasoja. Beyer *et al.* [2004] jakaa semanttisen tason tyyppitasoon (ontologiat) ja instanssitason (terminologiat) ja erottaa nämä kaksi tasoa omikseen syntaktisen tason rinnalle. Samaa jakoa noudattaa myös Linthicum [2000] miettiessään organisaatioiden välistä sovellusintegraatiota. Park ja Ram [2004] puhuvat taas semanttisen tason *konflikteista*, joita voi olla data-tasolla tai malli-tasolla (schema level). Periaatteessa näkemys on sama, mallien välisissä vertailuissa data-taso vastaa instanssitason ja malli-taso tyyppitasoa.

Tyyppimallilla eli ontologisella mallilla voidaan sovellusintegraatiossa ajatella tarkoitettavan sovelluksen sisäisesti käyttämän käsitteistön mukaista mallia tai mallijoukkoa sovelluksen kohdealueesta [Abels *et al.*, 2005]. Toisin sanoen, ontologiatasolla sovellusintegraatiossa ongelmana ovat sovellusten erilaiset, epäyhteensopivat mallinnukset kohdealueen reaali maailman käsitteistä. Ongelmia voivat aiheuttaa esimerkiksi erilaiset termit samasta asiasta tai käsitteiden tai tiedon jaottelun ero. Koska terveydenhuollossa tieto on hyvin usein rakenteista ja hierarkista, on yhteisen ontologisen kielen määrittely ollut erittäin tarpeellista. Tähän tarpeeseen vastaa myös jo aikaisemmin mainittu HL7-standardi, jolla on pyritty luomaan yhteisiä käsitteistöjä toimialalle [Beyer *et al.*, 2004] [Ferranti *et al.*, 2006]. Huomioida kannattaa kuitenkin, että HL7 on kehittyvä standardi, joten siitä on olemassa eri versioita, joissa keskeisenä erona on viestien rakenteisuus, toisin sanoen ontologinen tarkkuus. Sovellusintegraatiota tehtäessä HL7-standardia käyttävien organisaatioiden kesken integraation mahdollisuudet rajoittuvat heikoimman, toisin sanoen vanhimman käytetyn HL7-standardin implementaation mukaan. Analogia tyyppimallilla ja Parkin ja Ramin [2004] mallitasolla on, että kummankin voidaan nähdä edustavan tietoa ja tietoalkioiden suhteita keskenään. Tämä tieto ja alkioden suhteet voi olla luettavissa aina tietokannan taulujen tasolle saakka. On tärkeätä huomata, että HL7-standardi ei välttämättä johda

ontologioiden yhtenevyyteen, vaan niiden mahdollisimman suureen yhteentoiminnallisuuteen, tämäkin vain usein viesti- tai asiakirjatasolla. Sovellusintegraation tutkimusalalla ontologiaintegraation voidaan nähdä käsittävän tyyppimallien (tietokantarakenteiden) yhdistämisen lisäksi myös instanssitason. Tämän vuoksi ontologioiden yhdistäminen ei välttämättä tuota samaa rakennetta, kuin mitä tietokaavioiden yhdistäminen [Abels *et al.*, 2005].

Instanssitaso ja data-taso kohdentuvat kummatkin dataan ja sen esitykseen. Instanssi- tai datatason konflikteihin voidaan ajatella kuuluvan kaikki ne ongelmat, jotka liittyvät joko datan esittämiseen, mittaamiseen tai tarkkuuteen. Esimerkkeinä näistä ongelmista voi olla vaikka päivämäärän esittämistapa (15.04.2008 vai 15-April-2008) tai jonkun arvon esittäminen toisessa tietokannassa metreinä ja toisessa kilometreinä. Koska luokituksia ja koodistoja on terveydenhuollon alueella paljon, on niillekin luotu useita kansainvälisiä luokituksia. Yleisin ja käytetyin näistä on ICD-10-tautiluokitus.

3.1.4. Sovellusintegraation tavoitemalleja

Sovellusintegraatiolle on luotu kirjallisuudessa joitain mahdollisia rakennemalleja, toisin sanoen malleja siitä millaiseksi sovellusintegraation läpikäynyt sovellusrypäs voisi muodostua. Sekä Grimson *et al.* [2000] että Aier ja Schönherr [2006] esittelevät kolme erilaista sovellusintegraation tavoitemallia. Esiteltyt mallit muistuttavat siinä mielessä toisiaan, että niistä on mahdollista tunnistaa samat tavoitemallit, vaikka itse mallit muistuttavatkin toisiaan käsitteiden tasolla vain etäisesti. Tämä ei sinällään ole yllättävää, koska niissä on kuvattuna kolme erilaista tietojärjestelmäarkkitehtuurin lähtökohtaa, joissa edetään yksinkertaisimmasta ja vanhimmasta kohti uudempaa.

Grimson *et al.* [2000] jakaa mallinsa viestipohjaiseen rajapintamalliin, dataware housing- eli tietovarastomalliin ja komponenttipohjaiseen malliin. Näistä viestipohjainen on perinteistä, vanhakantaista kahden organisaation välistä keskustelua, jossa rajapinta on luotu erikseen ja vain näiden kahden organisaation käyttöön. Tietovarastomalli taas perustuu yhteen, keskitettyyn tietovarastoon, johon organisaatiot voivat tallettaa tietoa yhteiskäyttöön. Tämä lähestymistapa on nähtävissä heidän mielestään ongelmallisena, koska niin sanottuja liitännäistietokantoja (federated database) on vaikea saada toimimaan tyydyttävästi. Sen takia tieto on enemmän yksisuuntaisesti kirjoitettavaa, eli tietovarastoon kerran siirrettyä dataa on vaikea nähdä päivitetävän reaaliaikaisesti sovellusintegraatiosta huolimatta. Kolmas malli on hajautettu komponenttipohjainen malli, jossa ohjelmistokomponentit toimivat alustariippumattomasti keskenään. Tämä taas vaatii korkeata yhteisten standardien sopimista sovellus- ja verkkoliikennetasolla, mutta onnistuessaan voi tuottaa suurimman yhteistoiminnallisuuden tason.

Aier ja Schönherr [2006] näkevät tuoreemmassa tutkimuksessaan kolme mallia, joita he nimittävät point-to-point-malliksi (pisteestä pisteeseen), keskitetyksi

hub&spoke-malliksi (pyörännapa ja pinnat) ja hajautetuksi palvelukeskeistä arkkitehtuuria käyttäväksi malliksi.

Vertailtaessa näitä kahta mallikokoelmaa, näyttävät Aierin ja Schönherrin [2006] kaksi ensimmäistä mallia vastaavan samoja mitä Grimson *et al.* [2000] esitteli omissa malleissaan. Ensimmäisten mallien vastaavuus on ilmeinen, mutta myös toisten mallien voi nähdä muistuttavan toisiaan, onhan kummankin kantavana ideana keskitetty tietoa hallinnoiva tietojärjestelmä. Siitä, kun Grimson *et al.* [2000] esittelivät mallinsa, on kulunut sen verran aikaa, että sovellusintegraatio on uusien tekniikoiden ja metodien myötä saanut uusia suuntia. Toisaalta myös jo olemassa olleet ratkaisut kuten liitännäistietokannat ovat kehittyneet toimiviksi ja yleistyneet käytössä [Linthicum, 2003]. Myös komponenttiperustainen ohjelmistokehitys hajautettujen tietojärjestelmien kehityksessä on korvautunut pääosin palvelukeskeisellä arkkitehtuurilla. Aierin ja Schönherrin [2006] lainaavat hub&spoke-mallissaan Linthicumia [2000], mutta on syytä huomata, että omassa tutkimuksessaan Linthicum [2000] puhui nimenomaan viestien sanomanvälityspalveluista (message broker), ei sovellusintegraation läpi käyneen tietojärjestelmän kokonaisarkkitehtuurista. Sama typologia tosin toimii myös tässä.

Myös Linthicumin [2003] sovellusintegraation malli sisältää vahvan näkemyksen siitä, miltä sovellusintegraation läpikäyneet tietojärjestelmäkokonaisuudet näyttävät. Idea on, että sovellusintegraation malli muodostuisi hyvin pitkälle sen mukaan, miten sovellusintegraation läpikäyneet organisaatiot (tai organisaatio) itse muodostuvat. Vaikka tiettyjä yhteneväisyyksiä valitun mallin ja organisaation/organisaatioiden rakenteen kanssa voi ollakin, on syytä välttää suoran syy-seuraussuhteen päättelyä.

3.1.5. Sovellusintegraation vaiheistuksesta ja kypsyydestä

Sovellusintegraatioprosessin vaiheistusta ovat miettineet muun muassa Linthicum [2000] ja Stumptner *et al.* [2004]. Kumpikin tutkimus esittää tietyt, rajatut vaiheet, joiden mukaan sovellusintegraatio voitaisiin tehdä. Kumpikin lista myös käsittelee enemmän organisaation sisäisten tietojärjestelmien (tai sovellusten) integroimista kuin organisaatiotasot ylittävää integrointiprosessia.

Linthicumin [2000] lista muodostuu 12 vaiheesta jonka mukaan integraatioprosessi pitäisi viedä läpi:

1. Yrityksen liiketoiminnan ja integraatio-ongelma kohdealueen tunnistus
2. Organisaation datan (ja siitä johdetun tiedon) analysointi, tunnistaminen ja sijaintien ja datavirtojen päättely
3. Organisaation liiketoimintaprosessien analysointi ja ymmärtäminen
4. Organisaation sovellusten rajapintojen tunnistaminen
5. Liiketoimintatapahtumien tunnistaminen ja tiedon tallentaminen
6. Datan (ja siitä johdetun tiedon) mahdollisten muunnostapahtumien tunnistaminen sovelluksissa

7. Tiedon varsinaisten muunnosten prosessien luonti
8. Sovellusintegraation teknologian valinta ja prosessin toteuttaminen
9. Toteutetun teknologisen sovellusintegraatioprosessin testaus
10. Integroidun järjestelmän suorituskyvyn arviointi
11. Suoritetun sovellusintegraatioprosessin hyötyjen arviointi
12. Sovelluskokonaisuuden ylläpitoprosessin luonti.

Tämä lista on varsin yleinen ja käytännönläheinen. Samalla se myös saattaa liikaa yksinkertaistaa sovellusintegraation ongelmakohtia, varsinkin tiedon semantiikan ja järjestelmien ontologioiden tunnistamista ja analysointia. Kuitenkin mielestäni varsinaisen integroimisoperaation jälkeinen prosessi on saanut tarvittavan huomion.

Stumptner *et al.* [2004] on antanut myös listan sovellusintegraation vaiheistuksesta. Heidän listassa on vain neljä vaihetta jotka ovat:

1. Järjestelmien sisäisten tietomallien yhdistäminen yhdeksi malliksi
2. Järjestelmien tietomallien semanttisten puutteiden etsiminen ja korjaaminen järjestelmien yleiseen tietomalliin
3. Tiedon semanttisten vastaavuuksien tunnistaminen
4. Varsinainen integrointiprosessi, jossa löydetyt semanttiset epäyhentevyudet korjataan ja linkataan yhteen edellisissä vaiheissa luotujen mallien ja suunnitelmien pohjalta.

Stumptner *et al.* [2004] esitti siis listan, jossa vaiheistus on vielä karkeampi kuin edellä mainittu Linthicumin [2000] lista. Huomio on selvästi enemmän tiedossa ja sen semantiikassa kuin varsinaisessa integraatioprosessissa. Ylipäätään esitelty lista on niin lyhyt ja vajavainen, että vaikka sitä tutkimuksessa tarkennetaan vaiheiden osalta, ei se siltikään ole mitenkään kattava.

Jos vertaillaan edellä esitettyjä listoja keskenään niin huomataan, että Linthicum [2000] ajattelee listassaan asioita enemmän käytännönläheisesti kuin Stumptner *et al.* [2004]. Voidaan jopa sanoa, että Linthicum [2000] palvelee omalla listallaan sovellusintegraatiotarpeiden kimpussa painivaa IT-päälliköä ja Stumptner *et al.* [2004] taas tutkijaa, joka on enemmän kiinnostunut tiedosta käsitteenä ja sen tulkinnasta kuin siitä, miten sitä käytetään työelämässä.

On varsin selvää, ettei sovellusintegraatio prosessina ole yleensä mitenkään selvä, rajattu ja yhtäjaksoinen ja -tasoinen operaatio, vaan esimerkiksi sovellusintegraation tarpeet organisaation sisällä voivat muuttua hyvinkin nopeasti ja jo tehtyjen integraatio-operaatioiden myötä voidaan törmätä uusiin tarpeisiin ja rajoitteisiin. Tämän takia näkökulman tulisi olla iteroiva ja jatkuva. Ongelmana usein on, että organisaatioissa integraatioprosessi suoritetaan kertaluonteisesti, jolloin jatkuva ja muuttuva näkemys jää huomaamatta.

3.2. Palvelukeskeinen arkkitehtuuri

Luvussa 3.1 käsiteltiin sovellusintegraatiota ja sen ongelmia. Lisäksi nopeasti muuttuvassa yritysmaailmassa ja toisaalta hajautuneissa organisaatiotasoissa ilmenee suurta tarvetta olla toimissaan mahdollisimman ketterä (*agile*) ja toisaalta tehokkaan suoraviivainen [Papazoglou and van den Heuvel, 2007]. Tietojärjestelmäkokonaisuus, on se rakennettu sitten yhden tai useamman organisaation kesken, voi muodostua useista eri komponenteista ja alijärjestelmistä, joiden tarvitsee toimia kuitenkin tehokkaasti yhteen. Tätä taustaa vasten palvelukeskeisessä arkkitehtuurissa eli SOA-arkkitehtuurissa (Service Oriented Architecture) on paljon haluttuja ominaisuuksia, koska se on alun perin juuri kehitetty auttamaan hajautettujen järjestelmien mahdollisissa ongelmatilanteissa. Tällaisia ovat muun muassa juuri sovellusintegraatio, transaktioiden hallinta (tietojärjestelmien kesken) ja tietoturva. Näin ollen palvelukeskeisestä arkkitehtuurista on luotu löyhästi kytkeytynyt, tunnettuihin standardeihin perustuva ja toisaalta protokolla-, laitealusta- ja ohjelmointikieliriippumaton [Papazoglou and van den Heuvel, 2007 [OASIS Group, 2007]].

Palvelukeskeisessä arkkitehtuurissa oleellista siis on, että siinä löyhästi toisiinsa sidotut ohjelmistokomponentit eli palvelut toimivat keskenään niin, että niitä käyttäen on mahdollista suorittaa palvelukokonaisuuksia. Näiden palveluiden tarjoamat toiminnot on määritetty etukäteen ja palveluille on julkaistu rajapinta, jolla voi toteuttaa prosesseja. Tämän takia palvelut eivät ole riippuvaisia toisistaan vaan ne toimivat itsenäisesti niin, ettei kahden palvelun pitäisi koskaan olla riippuvaisia toistensa tilasta. Oleellinen seikka palvelujen toiminnassa on myös se, että ne ovat dynaamisesti hallittavissa, toisin sanoen niitä voidaan ajonaikaisesti liittää, kutsua ja niiden suoritusta ohjata uudelleen. Näin eri palveluita käyttäen voi varsinainen ohjelmalogiikka, josta käytetään nimitystä prosessimoottori, johtaa eli orkestroida näistä pienemmistä palveluista suurempia palvelukokonaisuuksia. Sen tärkeänä tehtävänä on myös toimia välittäjänä, joka tarjoaa rajapinnan palveluiden toteuttajille ja näin piilottaa palveluiden mahdollisen alustariippuvuuden.

Kuten aikaisemmin todettiin, SOA-arkkitehtuuri ei määrää mitään ohjelmointikieltä tai varsinaista tekniikkaa millä palvelut tulisi implementoida [OASIS Group, 2007]. Yleisesti käytetyin ratkaisu on Web services-tekniikka, mutta myös muita ratkaisuja kuten CORBA on mahdollista käyttää. Web services-ratkaisussa tarjotaan www-tekniikoita käyttäen rajapinta palveluiden välittämiseen. Termit ovat hiukan päällekkäisiä, mutta eivät missään mielessä synonyymeja. Web services on käytettävä teknologia ja palvelukeskeinen arkkitehtuuri tietojärjestelmäarkkitehtuurin toteutusparadigma. Eli vaikka Web services:llä toteutettu palvelu voi olla SOA-arkkitehtuurin mukaista käyttäessään tiedon välitykseen sanomia, voi toteutettava palvelu myös käyttää esimerkiksi suoria funktiokutsuja [World Wide Web Consortium

(W3C), 2004] tai toimia ilman prosessimoottoria [Papazoglou and van den Heuvel, 2007].

Viestipohjainen Web services-teknologia on yleisin käytössä oleva palvelukeskeisen arkkitehtuurin toteutusteknologia ja sille on olemassa laajin tuki järjestelmätoimittajien keskuudessa. Näin myös HL7-standardissa, jossa on määriteltynä laajennos, jossa käytetään Web services:in SOAP-viestinvälitysprotokollaa ja sitä kuvaavaa WSDL-kieltä [HL7 Finland, 2007b]. WSDL-kieli onkin noussut lähes standardiksi palvelukeskeiseen arkkitehtuurin nojaavien ratkaisujen palvelujen kuvauskielenä niin, että myös muille teknologioille kuten CORBA ja J2EE löytyy WSDL-tuki [Papazoglou and van den Heuvel, 2007]. SOAP-viestit siis pitävät sisällään HL7-sanomia.

Oleellinen seikka palvelukeskeisen arkkitehtuurin toteuttamisessa on päättää, miten tietojärjestelmän osapuolet hoitavat toistensa löytämisen. Arkkitehtuurissa puhutaan yleensä palvelun tarjoajasta (server) ja palvelun tilaajasta (client). Osapuolten kommunikoinnin avaus voidaan tehdä kahdella eri periaatteella: palvelun kokoamisella (service aggregator) tai palvelun välittämällä (service broker). Näistä palvelun kokoajaa käytettäessä palvelun tilaajalla on tieto palvelun kokoajasta, joka toimii kahdessa roolissa eli tilaajaan päin palvelun tarjoajana ja varsinaisiin palveluihin päin taas palvelun tilaajana. Näin varsinaiset palvelut ovat tilaajalta piilossa, mutta kuitenkin sen käytettävissä. Huomattavasti suositumpi ratkaisu on kuitenkin palvelun välittäjä-malli, jossa on olemassa tilaajan ja tarjoajan lisäksi kolmas, luotettu osapuoli eli palvelun välittäjä, joka tarjoaa tilaajalle tiedot palvelun tarjoajasta/tarjoajista ja tiedottaa mahdollisista palveluista, joista palvelun käyttäjä voi valita. Olennaista on, että sekä palvelun toteuttaja että sen käyttäjä voivat luottaa välittäjään ja myös palvelutransaktion toiseen osapuoleen, mikäli palvelun välittäjä todentaa osapuolen oikeellisuuden. Näitä palvelun tarjoajien kuvailua varten on Web services-teknologiassa käytössä UDDI-kieli.

3.3. Keskitetty vai hajautettu: tietojärjestelmäarkkitehtuurin taistelukenttä

Tässä luvussa käsitellään kahta vastakkaista tietojärjestelmäarkkitehtuurinäkemystä tietojärjestelmien rakentamiseen. Alkuun niiden nähtiin ”taistelevan” keskenään niin, ettei vastakkaiselle näkemykselle välttämättä nähty tarvetta ollenkaan tai se tuomittiin absoluuttisesti heikommaksi. Tarpeet ja teknologiat ovat vaihdelleet niin, että kummallekin näkemykselle on ollut omat kannattajansa, mutta nykyään palvelukeskeisen arkkitehtuurin myötä näkemys siitä, mikä on hajautettua ja mikä keskitettyä, on laajentanut. Kirjallisuudessa ei enää ajatella kokonaisia tietojärjestelmiä joko puhtaasti hajautettuina tai puhtaasti keskitettyinä. Tässä luvussa kerrotaan lyhyesti suunnista, mihin nämä kumpikin tietojärjestelmäarkkitehtuuri ovat kehittyneet, ja mitä ne tarkoittavat nykyään. Kannattaa huomata, että edellisessä luvussa esitelty palvelukeskeinen tietojärjestelmäarkkitehtuuri lasketaan omaksi kehityssuunnakseen. Palvelukeskeisen arkkitehtuurin kannalta ei ole väliä, onko tietojärjestelmä tai

tietokanta keskitetty vai hajautettu. Tämä seikka kuitenkin huomioidaan luvussa 3.3.3, jossa käsitellään eri tietojärjestelmäarkkitehtuurien mukaista sovellusintegraatiota.

3.3.1. Keskitetty tietojärjestelmäarkkitehtuuri

Keskitetty tietojärjestelmäarkkitehtuuri on alun perin tarkoittanut suurkanone-käytäntöjä, jolloin muunlaisia tietojärjestelmäarkkitehtuureita ei ollutkaan. Tietojärjestelmät toimivat isoissa keskuskoneissa ja käyttäjillä oli niihin yhteys pääteterminaalien kautta. Kun hajautettu tietojärjestelmäarkkitehtuuri alkoi saada suosiota, nähtiin keskitetty tietojärjestelmäarkkitehtuuri vanhanaikaiseksi ja kankeaksi [Linthicum, 2000] [Brown, 2002]. Tietojärjestelmien hajautuksen myötä eteen tuli uusia ongelmia, kuten esimerkiksi tiedon hajaantuneisuus ja sovellusintegraation vaikeus ja toisaalta hajautetun järjestelmän ylläpidon kalleus ja suuritöisyys.

Hajautetun tietojärjestelmäarkkitehtuurin nousu taittui nopeasti keskitetyn järjestelmän uudeksi suosioksi. Internet-huumassa hetken aikaa nähtiin verkkokauppojen suurien asiakasmäärien tarkoittavan käytännössä sitä, että ainoastaan keskitetyillä tehokkailla ratkaisuilla oli sijansa tässä nopeasti muuttuvassa ympäristössä. Toisaalta hajautettu komponenttipohjainen ohjelmistokehitys toi tullessaan uuden lupauksen hajautuksen mahdollisuuksista [Grimson *et al.*, 2000]. Missään vaiheessa suurkanoneet eivät ole kuitenkaan kadonneet konesaleista, mutta keskitetyn tietojärjestelmän rooli on laajentunut. Keskitetyn tietojärjestelmäarkkitehtuurin ei enää nähdä välttämättä tarpeelliseksi olla kiinteästi yhdessä fyysisessä pisteessä, vaan asiaa voi ajatella enemmän liiketoimintaprosessien hallinnollisen mallin kautta [Linthicum, 2003]. Vaikka varsinaisia kiinteästi yhteen palvelimeen sidottuja keskitettyjä perinnejärjestelmäratkaisuja onkin edelleen paljon käytössä [Linthicum, 2000] [Park and Ram, 2004] [Aier and Schönherr, 2006], on teknologian kehittyminen ja sovellusten välisen yhteistoiminnallisuuden tarve saanut niiden roolin muuttumaan. Näin liiketoimintaprosessien mukainen roolitus tuntuukin luonnollisemmalta. Roolien uusjaon jälkeen keskitetyksi tietojärjestelmäksi voi nostaa tai nähdä sellaisena useiden eri tietojärjestelmäkehityksen metodien mukaisia ratkaisuja.

Keskitetyksi ratkaisuksi voidaan laskea vaikkapa data warehousing-ratkaisut, tosin tässä tapauksessa keskitetty tietovarasto on yleensä ajateltu enemmänkin ulkoisena, esimerkiksi raportoinnin tai ennustelaskelmien apuvälineenä eikä kokonaisvaltaisen tietojärjestelmäarkkitehtuurin keskitettynä hallinnoijana. Myös palvelukeskeisen arkkitehtuurin pohjalta rakennetut ratkaisut voidaan nähdä keskitettyinä joissain tapauksissa [Linthicum, 2003] [Davis and Gamble, 2004]. Palvelukeskeisen arkkitehtuurin luvussa esiteltyä palveluväylää käyttäen tietojärjestelmä voi olla keskitettynä tietovarastona tai toisaalta hajautettuna palveluryppäänä. Kysymys onkin uusissa tietojärjestelmäarkkitehtuureissa siis enemmän siitä, mistä näkökulmasta asioita katsoo.

3.3.2. Hajautettu tietojärjestelmäarkkitehtuuri

Hajautettu tietojärjestelmäarkkitehtuuri tuli tietojärjestelmätieteeseen uutena, lupaavana kehityssuuntana henkilökohtaisten tietokoneiden yleistymisen myötä. Jäykät, keskitetyt palvelimet ja niissä pyörivät sovellukset olivat kömpelöitä käyttää verrattuna uusilla graafisilla käyttöjärjestelmillä varustettuihin pc-koneisiin. Lisäksi lehdistö ja yhteisö hurrasivat uusien ratkaisujen tuloa niin, että liiketoimintaa tukevien, pitkäaikaisten tietojärjestelmäpäätösten sijaan usein käyttöön valittiin uutta ja jännittävää teknologiaa [Linthicum, 2000]. Kehityskaari jatkui monikerroksisten arkkitehtuuriratkaisujen kautta komponenttiperustaiseen ohjelmistokehitykseen, missä tiedon ja prosessoinnin hajautus olivat lopulta mahdollisia. Tässäkin tapauksessa keskitetyn ja hajautetun tietojärjestelmäarkkitehtuurin rajat murtuivat. Tieto saattoi olla olemassa keskitetyssä tietovarastossa tai tietokannassa, mutta sen prosessointi oli hajautettua [Cavazos and Jarquín, 2004].

Toisaalta varsinaisen keskitetyn tietojärjestelmän tason takana tapahtui kehitystä. Tietokantatasolla nähtiin, että niitä voitiin hajauttaa niin, että kantoja käyttävillä sovelluksilla tai käyttäjillä ei tarvinnut olla tietoa niiden todellisesta sijainnista tai siitä miten ne muodostuivat. Tästä kehittyi niin sanottuja liitännäistietokantoja (federated database), joissa itsenäiset tietokannat (ja tietojärjestelmät myös) saattoivat toimia yhteen. Liitännäistietokantojen ja tietojärjestelmien käyttäjille kaikki näkyi yhtenä isona kokonaisuutena, johon heillä oli pääsy [Sheth and Larson, 1990] [Stumptner *et al.*, 2004]. Oleellista kuitenkin on, että liitännäistietokantojen osat ovat itsessään itsenäisiä, kokonaisia tietojärjestelmiä, mutta liitännäisyyden kautta niistä on pääsy myös muiden tietoresursseihin.

Hajautettu tietojärjestelmäarkkitehtuuri lisäsi kiinnostavuuttaan komponenttipohjaisten ja hajautettujen ohjelmointistandardien kuten CORBA ja DCOM myötä. Niillä saattoi rakentaa tehokkaita rajapintoja hajautetuille ratkaisuille. Kuitenkin vasta palvelukeskeinen arkkitehtuuri mahdollisti aidon alusta- ja tietojärjestelmäriippumattomuuden. Sitä voidaan kuitenkin pitää omana lähestymistapanaan tietojärjestelmien kehitykseen, mutta on syytä huomata sen mahdollistaneen sekä tiedon että prosessoinnin hajautuksen [Papazoglou and van den Heuvel, 2007].

3.3.3. Hajautetun ja keskitetyn tietojärjestelmäarkkitehtuurin integraatio

Hajautetun ja keskitetyn tietojärjestelmäarkkitehtuurin integroinnissa oleellista on se, että kumpi tietojärjestelmäarkkitehtuuri on lopulta hallitseva ja ylipäättään onko tällaista hallitsevaa osapuolta. Asiaa voidaan ajatella liitännäistietokantojen kautta niin, että jos liitetään keskitetty tietojärjestelmä hajautettuun ympäristöön, tulee tästä keskitetystä tietojärjestelmästä vain yksi hajautetun tietojärjestelmän lähteistä [Sheth and Larson, 1990]. Näin ainakin, jos asiaa katselee hajautetun järjestelmän kautta. Toisaalta, jos

järjestely on tarkoitus tehdä toisin päin eli keskitetyn arkkitehtuurin tietojärjestelmä on hallitseva ja kaiken kokoava, muuttuu integraatio hankalammaksi ja tietojärjestelmien roolit voidaan nähdä muuttuvan prosessissa. Näin siksi, että tällöin hajautettu tietojärjestelmä, mikäli sillä ei integraation jälkeen ole omaa itsenäistä roolia ja toimintoja, on vain välivarasto keskitetylle tietojärjestelmälle. Tämä tilanne on juurikin tämän tapaustutkimuksen aiheena.

Keskitettyjen ja hajautettujen tietojärjestelmien yhdistelmiä voidaan pitää hybridirakenteisina tietojärjestelminä. Niissä on kumpienkin ominaisuuksia ja riippuen siitä, mistä niitä katselee, voi niitä nimittää niin keskitetyiksi kuin hajautetuiksikin [Elmasri and Navathe, 2000].

Palvelukeskeisen arkkitehtuurin voi nähdä toteuttavan sen integraatiotarpeen, mitä näillä kahdella tietojärjestelmäarkkitehtuurilla on keskenään. Se sisältääkin lupauksen siitä, ettei tietojärjestelmän yksittäisten palveluiden tarvitse tietää siitä, missä muut järjestelmän palvelut sijaitsevat, itse asiassa palveluiden ei tarvitse tietää muista palveluista *mitään*. Tiedon ja palveluiden hallinnointi tapahtuu kuitenkin palveluväylän orkestroinnilla [Papazoglou and van den Heuvel, 2007].

4. Tapaustutkimus KANTA-järjestelmän ja aluetietojärjestelmien yhteensovittamisesta

Tässä osiossa esitellään varsinaisen tapaustutkimuksen aihealue. Ensin esitellään KANTA-arkkitehtuurin määrittelyt ja niiden asettamat vaatimukset KANTA:an liittyville terveydenhuollon järjestelmille. Tämän jälkeen kerrotaan muutamista standardeista ja toimintamalleista, jonka pohjalta KANTA-järjestelmää rakennetaan. Tätä seuraa suunnitellun KANTA-tietojärjestelmän kuvaus niin yleiseltä kannalta kuin aluetietojärjestelmien lähtökohdista. Lopuksi kuvaillaan lyhyesti viitetietokantojen roolia hajautettuna tietojärjestelmänä ja toisaalta kansallisen sähköisen arkiston roolia keskitettynä tietojärjestelmänä.

4.1. KANTA-potilastietojen arkistopalvelun määrittelyt

KANTA-määrittelyt ovat jaettu tässä projektissa perinteisiin ohjelmistoteknologiassa lähes vakiintuneisiin osioihin eli vaatimusmäärittelyyn (mitä palvelun pitää tehdä) ja arkkitehtuurimäärittelyyn (miten se mitä palvelu tulee tekemään yleisellä tasolla toteutetaan). Huomioitavaa on, että vaatimusmäärittelyt on kirjoitettu kertaluonteisesti samoihin aikoihin arkkitehtuurimäärittelysten kanssa. Sen lisäksi se, että määrittelytyö on tehty konsultointityönä johtaa tiettyihin piirteisiin: Määrittelyt eivät ole käyneet montaakaan iterointikierrosta läpi. Julkisia lausuntokierroksia on pidetty vain yksi. Sen vuoksi KANTA-määrittelyt eivät ole samalla tavalla muutosalttiita kuin mitä esim. Haikala [2006] näkee tärkeäksi ohjelmistoprojektissa tehtävän. Tosin KANTA-määrittelyissä on useat yksityiskohdat jätetty avoimeksi, epäilemättä siksi ettei toimintaympäristön muutoksia ja resurssointia vielä voinut määrittelysten laadintavaiheessa nähdä. Näin esimerkiksi KANTA-kokonaisuuden testauksesta ja laadun varmistuksesta mainitaan vain yleisiä ohjeita kuten, että järjestelmä tulee olla käytettävissä 99.9 % ajasta.

Pelkät konsultointina toteutetut määrittelydokumentit ovat varsin vajavaisia, kun ajattelee kuvatun palvelun liittämistä terveydenhuollon toimintaprosesseihin. Esimerkiksi asiakirjan luonti hoitajaksoista jätetään auki ja varsinaiset use case-skenaariot ovat yliyksinkertaisia. Tästä syystä palveluiden ja kokonaisarkkitehtuurin vaatimus- ja arkkitehtuurimäärittelysten lisäksi KANTA-järjestelmään luotu Kanta jatkomäärittely eli Ydindokumentin uusi versio [STM, 2007e] on erittäin tarpeellinen. Sen tehtävänä on tarkentaa nimenomaan niitä monia asioita, jotka KANTA-määrittelyissä ovat jääneet huomioimatta joko tarkoituksella tai tahattomasti. Osittain huomioidut ovat samoja kuin tässä tutkimuksessa, mutta ydindokumentin lähestymistapa avoimiin kysymyksiin on vähemmän tekninen ja keskittyy enemmän terveydenhuollon prosessien ja KANTA-toteutuksen integrointiin [STM, 2007e]. Samalla siinä

huomioidaan kuitenkin jo olemassa olevan lain sosiaali- ja terveydenhuollon asiakastietojen sähköisestä käsittelystä tuomat rajoitukset.

KANTA-arkkitehtuurin kuvaus ei ole tämän tutkimuksen pääaluetta, vaan tämä tutkimus kohdistuu mahdollisuuksiin liittää hajautetut alueelliset viitetietokantajärjestelmät KANTA-palveluun. Näin ollen voidaan yleisesti todeta, että terveydenhuollon kansallisen tietojärjestelmäarkkitehtuurin määritykset ovat teknisessä mielessä tarpeeksi kattavat ilman, että niissä sitoudutaan liikaa mihinkään paradigmaan tai ollaan turhan spesifisiä. Jotkin toisaalta tekniset ja toisaalta toiminnalliset seikat vaativat kuitenkin lisää määrittelyä. Esimerkiksi palvelukeskeisen arkkitehtuurin esittelyn yhteydessä käsitellyn prosessimoottorin orkestrointi eri toimintatapauksissa on KANTA-määrityksissä varsin suurpiirteisesti kuvattu.

4.1.1. Vaatimusmäärittelyistä

Kansallisen terveydenhuollon sähköisen arkiston vaatimusmäärittelyissä on kiinteästi olemassa vaatimuksia siitä, mitä minkäkin komponentin tulee tehdä. Tarkemmin vaatimukset on jaoteltu seuraaviin ryhmiin: yleiset, juridiset, toiminnalliset, arkkitehtuuriin liittyvät, käytettävyyteen liittyvät, suorituskykyyn liittyvät, tietoturva koskevat ja laatuun liittyvät vaatimukset. Lisäksi vaatimuksille on olemassa priorisointi riippuen siitä, onko vaatimus pakollinen, tarpeellinen vai toivottava. Vaatimusten luettelointi ja jaottelu on varsin suositeltua vaatimusmäärittelyissä. Kun vaatimukset ovat kiinteästi kirjattu, saadaan näin asetettua mittareita, joilla voidaan arvioida koko ohjelmiston toimivuutta ja oikeellisuutta. Toisaalta, koska vaatimusmäärittelyt usein kuitenkin tehdään aikaisessa vaiheessa ohjelmistoprojektia, tulevat ne (tai niiden voi jopa toivoa tulevan) aina elämään jonkun verran [Haikala, 2006]. KANTA-projektin tapauksessa kuitenkin monet vaatimukset juontavat juurensa olemassa oleviin lakeihin ja suosituksiin, jolloin näillä vaatimuksilla on olemassa varsin tukeva tausta. Tosin, jos todelliset tarpeet ovat ristiriidassa lakitekstin sisällön kanssa tai lakiteksti itsessään muodostaa ristiriidan, on tällöin myös lakitekstin jälkikäteinen muuttaminen työlästä.

Kokonaisarkkitehtuurin vaatimusmäärittelyssä todetaan, että se on luotu käyttäen avuksi monien aikaisempien työryhmien tutkimusten tuloksia [STM, 2007b]. Vaikka työryhmillä onkin kokonaisnäkymä terveydenhuollon toimintamalleihin ja lakisäädöksiin, olisi oleellista varmistaa, että suunnittelussa huomioidaan myös jokapäiväisen käytön helppous. Tämä on toki kiinni vähintään yhtä paljon perusjärjestelmien muuttamisesta sujuvasti käyttämään KANTA-palvelua kuin itse KANTA-järjestelmän kehityksestä. Helppokäyttöisyys voi toisaalta olla pahimmillaan ristiriitainen vaatimus tietoturvan ja suostumusten hallinnan kanssa, mutta varsinaisena ongelmana on se, että käyttöön ja sen sujuvuuteen liittyviä vaatimuksia on vaikea kirjoittaa tarkasti mitattavilla parametreilla kun lopullisen järjestelmän kokonaisuudesta ei ole vielä kuin ylimalkaisia suuntaviivoja.

Vaatimusten lisäksi KANTA-järjestelmän vaatimusmäärittelyihin on kirjattu ajateltuja käyttötapauksia. Haikalan [2006] mukaan tämän voi ajatella liittävän järjestelmän määrittely paremmin reaali maailman käyttöympäristöön. Tässä tapauksessa siis liitetään esimerkiksi sairaanhoidon asiakaskäynti osaksi KANTA-järjestelmän vaatimusmäärittelyä.

4.1.2. Arkkitehtuurimäärittelyistä

Ensimmäisiin arkkitehtuurimäärittelyihin tehtiin muutoksia, joissa poistettiin riippuvuuksia tiettyihin ohjelmistoteknologioihin ja muutettiin järjestelmä toteutettavaksi käyttäen palvelukeskeistä arkkitehtuuria [STM, 2007f]. Tämä lähestymistapa on varsin järkeenkäypä, jos arkkitehtuurimäärittelyissä ei sitouduta erikseen mihinkään teknologiaan [Haikala, 2006]. Teknisen määrittelyn tarkoituksena on kuvata varsinainen toteutus niin aukottomasti kuin mahdollista. Koska toteutusprojekti on vuoden 2008 alussa vasta suunnitteluvaiheessa, ei teknistä määrittelyä vielä ole ainakaan julkaistuna. Toisaalta tekniset määrittelyt voidaan ajatella jo lähes liikesalaisuuteen rinnastettavaksi tiedoksi, joten niiden julkaisua ei välttämättä tule tapahtumaan koskaan. Kokonaisarkkitehtuurin ja palveluiden määrittelyissä silmäänpistävää on se, kuinka paljon määrittelyiden tarkkuus vaihtelee. Tämä on ymmärrettävää, kun valituista teknologioista ei ole vielä varmuutta, mutta toisaalta tämä tekee määrittelyistä hiukan hajanaisen kokonaisuuden. Arkkitehtuurimäärittelyjä ei ollut ainakaan helmikuun 2008 alkuun mennessä julkaistu kuin vasta kokonaisarkkitehtuurista, viestinvälitys-palvelusta ja tunnistuksen ja sähköisen allekirjoituksen palvelusta.

4.1.3. Määrittelyiden asettamista vaatimuksista

Tutkittaessa sitä, mitä asioita Sosiaali- ja terveysministeriön laatimat KANTA-määrittelyt (tässä huomioidaan myös KANTA-jatkomäärittely eli ydindokumentti) asettavat ulkoisille, siihen liittyville terveydenhuollon tietojärjestelmille, saatiin niistä kerättyä varsin kattava valikoima erilaisia vaatimuksia. Löydetyt vaatimukset on jaoteltu viiteen eri kategoriaan:

- Yleisiin vaatimuksiin
- Juridisiin vaatimuksiin
- Toiminnallisiin vaatimuksiin
- Yleisiin arkkitehtuuri- ja teknisiin vaatimuksiin
- Tietoturvan vaatimuksiin.

Jaottelu on hiukan eri kuin mitä on KANTA-vaatimusmäärittelyissä, joissa edellämainittujen vaatimusten lisäksi ovat erottuna käytettävyyksivaatimukset, suorituskykyvaatimukset ja laatuvaatimukset. Koska vaatimukset ulkoisille järjestelmille olivat varsin yleisiä, ei näiden mukaista jaottelua nähdä tarpeelliseksi. Tosin Kela on laatimassa ulkoisille järjestelmille omaa sertifiointimenettelyä, jossa

varmasti käsitellään juuri esim. suorituskyky- ja laatuvaatimuksia. On syytä myös huomata, että tietojärjestelmille toki asetetaan monia sääntöjä juridisessa mielessä, mutta joita ei kuitenkaan mainita erikseen KANTA-järjestelmän määräyksissä. Tällaisia tapauksia on esimerkiksi laissa oleva vaatimus hoitosuhteen olemassaolosta, jotta terveydenhuollon työntekijällä on oikeus asiakkaan tietojen käyttöön. Asetetut vaatimukset ovat kokonaisuudessaan listattuna liitteessä 1.

Löydettyjen vaatimusten analysoinnissa keskitytään lähinnä niihin, jotka koskettavat erityisesti viitetietokantoihin perustuvia aluetietojärjestelmiä. Perustavaa laatua olevaan kysymykseen viitetietokantojen roolista KANTA-järjestelmän käyttöönoton jälkeen liittyy toiminnallinen vaatimus, jossa todetaan, että aluejärjestelmä voi olla liittyneenä KANTAan, mikäli se pystyy toimimaan kansalliseen järjestelmään päin potilastietojärjestelmän kaltaisesti ja toisaalta, että se pystyy toimimaan paikallisiin järjestelmiin päin vastaavalla tavalla palvelut välittäen kuin kansallinen sähköinen potilastietoarkisto. Tämä vaatimus siis määrittää, että mikäli viitetietokanta toimii perusjärjestelmän ja KANTA:n välissä, ei sillä saa olla mitään sellaista toiminnallisuutta, jota sitä ilmankin näiden välissä voisi olla.

Edellä mainittu vaatimus antaa ymmärtää, että viitetietokanta olisi jollakin tavalla potilastietojärjestelmän ja KANTA-palvelun välissä toimien eräänlaisena mediaattorina. Tästä lähestymistavasta kannattaa kuitenkin tehdä muutamia huomioita: Nykyiset viitetietokannat eivät ole suoraan integroituneet perusjärjestelmiin, vaan viitetietokantoja varten on perusjärjestelmiin luotu omat adapterit, joiden avulla ne luovat talletettavat viitteet asiakirjoista ja toisaalta, joilla ne luovuttavat omasta tietokannasta noudetut potilastiedot viitetietokannan kautta toisen organisaation käyttäjän nähtäville [Nykänen ja Karimaa, 2002a]. Viitetietokantaan liittyminen ja noudettujen asiakirjojen tutkiminen taas tapahtuvat erillisellä selainliittymällä, eli viitetietokannalla ja potilastietojärjestelmillä ei ole suoraa linkkiä toisiinsa muuten kuin tiedon siirron kannalta. Kansallista sähköistä arkistoa sitä vastoin on heti alusta alkaen lähdetty rakentamaan niin, ettei sillä tule olemaan omaa käyttöliittymää, vaan että sen käyttäjät käyttävät sitä suoraa omista perusjärjestelmistään. KANTA-palvelusta tehdyt haut ja asiakirjojen luovutukset pitää olla käyttäjälle mahdollisimman läpinäkyviä. Tähän taas viitetietokannoilla ei ole pyritty, ne kun ovat toteutettu mahdollisimman pienellä integrointitarpeella.

Määrittelyistä on löydettävissä ainoastaan toinen suora vaatimus joka suoraan liittyy viitetietojärjestelmiin. Yleisissä toiminnallisissa vaatimuksissa todetaan, että ”Potilasjärjestelmä huolehtii viitetietojärjestelmiin talletettujen tietojen siirtämisestä arkistopalveluun ja siirron edellyttämien suostumusasiakirjojen kirjaamisesta”. Tämän vaatimuksen voi nähdä hiukan ristiriitaisena sen vaatimuksen kanssa, jossa viitetietojärjestelmän tuli toimia nimenomaan välittäjänä KANTA-järjestelmän ja perusjärjestelmien välillä. On syytä miettiä, kuinka viitetietojärjestelmä voi toimia

näiden kahden tietojärjestelmän välissä, jos kuitenkin potilastietojärjestelmät ovat niitä, jotka siirtävät viitetietokantoihin tallennetut mahdolliset tiedot KANTA-järjestelmään? Toisaalta mitä nämä viitetietokantojen tiedot ylipäättään ovat? Viitetietokannoissa on talletettuna ainoastaan viitteitä asiakirjoista, jotka sijaitsevat eri puolilla sairaanhoitopiiriä eikä KANTA tee näillä tiedoilla mitään, koska sille itselle on talletettuna näiden asiakirjojen alkuperäiset sähköiset kopiot. Suostumusasiakirjat ovat hiukan eri asia, ne voidaan hyvinkin tallettaa sekä KANTA:aan että viitetietokantoihin, toki riippuen siitä, kuinka organisaation ulkopuolisten asiakirjojen katselu tullaan toteuttamaan.

Näiden vaatimusten lisäksi koodistopalvelun vaatimusmäärittelyssä [STM, 2007j] on maininta mahdollisuudesta, jossa perusjärjestelmän ja kansallisen sähköisen arkiston välissä voisi toimia alueellinen tai paikallinen koodistopalvelu. Tällainen koodistopalvelu voisi hyvinkin olla osana viitetietokannoista muodostuvaa aluetietojärjestelmää, onhan jo nykyisissä viitetietokannoissa olemassa omia koodistoja.

Lopuksi voitaneen muistuttaa, että vaikka edellä kerrottiin erilaisista vaatimuksista viitetietojärjestelmille, ovat nämä vaatimukset varsin suurpiirteisiä. Jopa määrityksissä todetaan, että viitetietokantojen rooli voi muuttua, mutta ne eivät tarkalleen sano miten. Kokonaisarkkitehtuurin vaatimusmäärittely sanoo seuraavaa: ”Aluetietojärjestelmien käyttöä ja roolia jatkossa KANTA-palveluiden suhteen voidaan linjata siten myös sairaanhoitopiireissä ja muissa alueellisissa organisaatioissa.”

4.2. KANTA-arkkitehtuurin käyttämiä standardeja ja teknologioita

Seuraavassa kerrotaan lyhyesti muutamista oleellisista standardeista ja teknologioista, joita KANTA-järjestelmän kehityksessä on käytetty.

4.2.1. HL7-standardi

HL7 on lyhenne, jolla tässä viitataan Amerikkalaisen, ei-kaupallisen terveydenhuollon standardeja kehittävän Health Level Seven-organisaation kehittämään standardien ryhmään. Varsinaisesti viitetietokantojen ja KANTA-järjestelmän viestinvälityksessä käytetään Suomen oman HL7-sisarorganisaation laajennoksia HL7-standardista. Tietojen vaihto KANTA-järjestelmän ja perusjärjestelmien välillä tapahtuu sanomaviesteinä käyttäen HL7 CDA R2-standardin (Clinical Document Architecture Release2) mukaista XML-pohjaista asiakirjarakennetta. Toisaalta sanomaviestintä viitetietokantojen ja perusjärjestelmien välillä on tähän saakka noudattanut vanhempaa CDA R1-standardia. Siirtyminen tuoreempaan versioon on kuitenkin tekeillä. R1-versiosta R2-versio on kehittynyt niin, että nyt sovellusten on käytettävä toistensa tuntemia koodistoja ja niiden tulee kertoa mitä koodistoa ne kulloinkin viestejä välittäessään käyttävät. Näin myös sovellukset ”ymmärtävät” toistensa viestejä eikä tulkinta perustu vain vastaanottavan pään ihmisten tulkintaan [HL7 Finland, 2007a]. Riippumatta CDA-asiakirjan versiosta on siinä aina kaksi osaa: CDA Header ja CDA

Body. Näistä CDA Header käsittää asiakirjan tunnistetiedot ja muut kuvailutiedot, joita ovat mm. hoitotapahtuma tai -jakso, potilas ja hoitava organisaatio sekä ammattilainen. CDA Body taas sisältää varsinaisen kliinisen sisällön [HL7 Finland, 2007a].

CDA Header-tiedot eli yleisemmin asiakirjaan liittyvät metatiedot ovat erittäin tärkeitä, koska ainoastaan niillä varmistetaan, että haluttu asiakirja löytyy arkistosta sitä myöhemmin haettaessa. CDA R2-rakenne perustuu siis XML-rakenteeseen, mutta sillä on kiinteästi sovittuna kentät, mitkä sen pitää sisältää ja toisaalta, mitä se voi sisältää [HL7 Finland, 2007a].

HL7-standardi sisältää erilaisia suosituksia ja tietomalleja. Suomessa yleisesti käytetään lähinnä HL7 CDA-standardia, joka perustuu HL7-standardin yleiseen tietomalliin RIM:iin (Reference Information Model) [HL7 Finland, 2007a]. Kirjallisuudessa HL7 CDA-järjestelmää on vertailtu muihin vastaaviin sähköisten asiakirjojen rakenteisiin. Sekä [Ferranti *et al.*, 2006] että [Eichelberg *et al.*, 2005] löysivät HL7-standardista paljon hyviä puolia, muun muassa tehokkaan tuen sovellusten yhteistoiminnallisuudelle ja hyvän muunneltavuuden. Sen vuoksi Suomen HL7-organisaatiokin on voinut luoda omia paikallisia laajennoksia standardista.

4.2.2. Muista standardeista

HL7-asiakirjojen lisäksi KANTA-arkistoon tullaan tallentamaan DICOM-standardin mukaisia kuvia. DICOM-standardi sisältää niin oman kuvaformaatin kuin myös viestinvälitysformaatin [DICOM, 2007]. Vaikka KANTA-järjestelmä perustuikin pitkälti HL7-asiakirjojen käyttöön, on vaatimusmäärittelyssä nähty tarpeelliseksi myös DICOM-kuvien suora tallettaminen ilman tätä HL7-kehystä [STM, 2007g]. Kannattaa huomioida, että vaikka kuvat talletettaisiinkin suoraan KANTA-palveluun, on niihin tarpeen sitoa HL7-asiakirjaa vastaavat metatiedot [STM, 2007e].

Asiakirjojen yksilöinti tapahtuu yksikäsitteisillä, ISO-standardin mukaisilla OID-tunnuksilla. Tätä standardoitua tunnusta voidaan käyttää yksilöimään niin CDA R2-dokumentit kuin DICOM-kuvatkin [Ensio ja Ruotsalainen, 2003].

KANTA-arkkitehtuurin tulevaisuuden näkymissä on alkuperäisten DICOM- ja HL7-dokumenttien lisäksi myös monen muun datan tallennus. Kysymykseen voisi tulla erilaiset biosignaalit. Se, missä formaatissa kaikki eri data talletetaan, on vielä julkistamatta. Ei ole mitään esteitä sille, että myös näissä käytettäisiin HL7-rakennetta. Tällöin ainakin varmistuttaisiin siitä, että raakadataan saataisiin mukaan myös oleellinen metatieto. Data itsessään ilman siihen liittyvien sisältöjen merkitystähän ei olisi vielä informaatiota. Ongelmana näissä uusien formaattien asiakirjoissa, samaten kuin DICOM-kuvissakin, on niiden mahdollinen suuri koko, joka voi johtaa mahdollisiin pullonkauloihin niin verkkoyhteyksissä kuin koko järjestelmän suorituskyvyssäkin [STM, 2007e].

4.2.3. Palvelukeskeinen arkkitehtuuri KANTA-arkkitehtuurissa

Kuten aikaisemmin todettiin, KANTA-arkkitehtuuri perustuu palvelukeskeiseen arkkitehtuuriin. Myös tässä tapauksessa sisäiset palvelut ovat itsenäisiä ja löyhästi toisiinsa sidottuja. KANTA-arkkitehtuurin viestinvälitykseen on kuvattu palveluväylä, joka piilottaa palvelut toisiltaan ja jossa sijaitseva prosessimoottori orkestroi näistä sisäisistä palveluista ulkoisia, perusjärjestelmille ja muille toimijoille näkyviä kokonaisuuksia. Esimerkiksi asiakirjojen haku ja luovutus kummatkin käyttävät sekä suostumustenhallinta- että loki-valvontapalvelua, vaikka ovatkin erillisiä ulkoisia palveluita eli ydinprosesseja. Lisäksi kaikissa operaatioissa tarvitaan käyttäjän tunnistamista ja tässä käytetään tunnistuspalvelua. KANTA-arkkitehtuurin määrityksissä kuvataan koodistopalvelu paikkana, josta perusjärjestelmien on mahdollista lukea arkistopalveluun liittymiseen tarvittavat tiedot kuten palveluosoitteet. Näin ollen KANTA-arkkitehtuuri noudattaa luvussa 3.1 esiteltyä palvelun välittäjän (service broker) mallia, jossa koodistopalvelu toimii kummankin osapuolen, perusjärjestelmien (client) ja KANTA-järjestelmän (server) luottamana toimijana.

Yhtenä erityispiirteenä on se, että ainakaan alkuvaiheessa KANTA-järjestelmässä ei ole suunniteltuna montaakaan erillistä, palvelukeskeisen arkkitehtuurin mukaista palvelua. Koska palvelukeskeinen arkkitehtuuri on kuitenkin modulaarinen ja palvelut on suojattu toisiltaan, voidaan uusia palveluita tarvittaessa lisätä hyvin pienillä muutoksilla kokonaisjärjestelmän rajapintoihin. Palvelut siis voivat toimia dynaamisesti keskenään niin, ettei palvelun tilaajan (client) tarvitse tietää palvelun tarjoajasta (server) muuta kuin palveluväylän tarjoamat tiedot. Tämä ratkaisu vastaa lähestymistapaa, jonka myös Papazogolou ja van den Heuvel [2007] esittelivät tutkimuksessaan. Toinen huomio KANTA-arkkitehtuurin SOA-arkkitehtuurin implementaatiosta tulee siitä, että palveluita voisi olla helppo replikoida ja hajauttaa niin, että osa käyttäjistä käyttäisi jotain toista palveluryppään identtistä palveluista ja joku toinen toista. Toki KANTA-arkkitehtuurin keskitetty rakenne asettaa omat rajoitteensa ainakin arkiston palveluiden hajautukselle. Kysymykseen voisivatkin tulla lähinnä jotkut tukitoiminnot.

Kuten aiemmin todettiin, palvelukeskeinen arkkitehtuuri ei määrää mitään ohjelmointikieltä. Näin myös KANTA-arkkitehtuurin alustavia määrityksiä muutettiin niin, että niistä poistettiin riippuvuuksia Web services-ratkaisuihin. Luultavasti Web services on kuitenkin toteutusratkaisu, onhan esimerkiksi Suomessa käytössä olevassa HL7-standardissa jo olemassa tuki SOAP-sanomakielelle ja koodistopalvelulla on olemassa palveluosoitteet Web services-ratkaisun UDDI-muodossa.

4.3. KANTA-järjestelmän palvelut

Seuraavassa kerrotaan KANTA-järjestelmän SOA-arkkitehtuurin mukaisista sisäisistä palveluista. Lähinnä keskitytään arkistopalveluun ja toisaalta viestinvälitykseen ja

siihen oleellisesti liittyvään koodistopalveluun. Muut palvelut esitellään vain pääpiirteissään.

4.3.1. Arkistopalvelu

Kokonaisarkkitehtuurimäärittely näkee arkistopalvelun kahtena eri osana: varsinaisena arkistona ja sen operatiivisena käyttönä ja toisaalta arkistonhallintana, joka käsittää asiakirjojen ja itse arkiston ylläpidolliset toimet. Jaottelu on varsin järkevä, koska jo tuore terveydenhuollon sähköisen asioinnin laki [L159, 2007] sanoo, että kansallisessa arkistointijärjestelmässä noudatetaan arkistointilakia. Tämä taas määrää arkiston rakenteen ja käytön niin, että määrittelyssä esitelty jaottelu tuntuu luontevalta [L831, 1994].

Arkistopalvelun vaatimukset perustuvat siis tarkoin lakiin, jossa määrätään, miten arkiston tulee toimia ei-teknisessä mielessä. Peruseriaate on se, että (arkistolaki koskee niin sähköisiä kuin paperiarkistoja [L831, 1994]) jokainen terveydenhuollon toimintayksikkö on aina arkistonmuodostaja ja jokaisella arkistonmuodostajalla on oma (ainakin yksi) arkistonhoitaja eli vastuuhenkilö, joka huolehtii varsinaisesti arkiston aineistosta, sen käytöstä, luovutuksesta ja mahdollisesta hävityksestä. Arkistonrakenteesta ja arkistonmuodostajan velvollisuuksista kerrotaan tarkemmin arkistonmuodostussuunnitelmassa, joka toimii oppaana arkiston rakentamisen yhteydessä. Arkistonmuodostussuunnitelman ohjeistuksen ylläpidosta huolehtii kansallisarkisto. Arkistointipalvelun vaatimusmäärittelyssä peruseriaate on, että vaikka KANTA-arkiston tieto onkin yhdessä paikassa, on se näennäisesti jaettu satoihin erillisiin arkistoihin niin, että jokainen arkistonmuodostaja eli rekisterinpitäjä saa oman arkistonäkymänsä omine oikeuksineen ja velvollisuuksineen.

Koska arkiston tiedon tuottajat ja käyttäjät ja tiedon ylläpitäjät ovat eri henkilöitä (tai toimivat ainakin eri rooleissa) ja tekevät täysin eri tehtäviä, on arkkitehtuurimäärittelyssä todettu jako varsin luonteva. Seuraavassa kuvataan lyhyesti arkiston operatiivista käyttöä kuten sitä kuvataan kokonaisarkkitehtuurimäärittelyssä ja arkistokomponentin vaatimusmäärittelyssä.

Arkistopalvelun määrittelyt ovat varsin suurpiirteisiä sen suhteen, koska ja miten asiakirjoja tulisi lisätä arkistoon. Sen vuoksi ydindokumentin kuvaamat käytännöt ovat tarpeen, jotta voidaan päätellä, koska asiakirjoja on tarkoitus arkistoida ja koska ei [STM, 2007e]. Peruseriaate kuitenkin on, että mikäli asiakirja on KANTA:aan lisätty, se säilyy siellä tarvittaessa haettavissa niin kauan kuin sen tulee arkistointilain mukaan säilyä. Mikäli asiakirjan julkaisussa on tapahtunut virhe (se esimerkiksi tulee kirjatuksi väärälle potilaille), tulee siitä silloin tehdä uusi asiakirja, joka korvaa vanhan kuitenkin niin, että tämä muutos on järjestelmästä jäljitettävissä.

Kun terveydenhuollon palveluyksiköt ovat integroineet itsensä KANTA-järjestelmään ja käyttävät jokainen omaa arkistokomponentin näkymäänsä, tämä vaatii, että kaikissa organisaatioissa on arkistonmuodostamissuunnitelma toteutettu täysin

samalla tavalla. Käytännössä arkistokomponentin vaatimusmäärittelyssä tämän todetaan tarkoittavan sitä, että jokaiselle asiakirjalle tulee annetuksi oikeat ja periaatteiltaan yhtenevät metatiedot, joilla asiakirjaa voi myöhemmin hakea arkistosta [STM, 2007g].

4.3.2. Hakemisto- ja rekisteröintipalvelu

Hakemisto- ja rekisteröintipalvelu toimii osana KANTA-palvelua tarjoten mahdollisuuden rekisteröidä järjestelmään talletettavat asiakirjat ja toisaalta hakea jo rekisteröityjä asiakirjoja. Asiakirjan rekisteröinnillä tarkoitetaan, että asiakirjasta talletetaan erikseen sen hakutiedot ja muut kuvailutiedot. Näillä hakutiedoilla voidaan sitten etsiä potilaan mahdollisia asiakirjoja arkistointipalvelusta. Huomioitavaa on, että muilla kuin varsinaisen asiakirjan luoneella palvelujen antajalla (rekisterinpitäjällä) on mahdollisuus hakea potilasasiakirjoja ainoastaan laissa määrätyillä hakutiedoilla. Asiakirjan luonut palveluntuottaja voi kuitenkin hakea tiedon KANTA-palvelusta myös laajennetuilla hakutiedoilla.

Asiakirjojen rekisteröinnin ja hakemisen lisäksi palvelu mahdollistaa jo rekisteröityjen asiakirjojen rekisteröintitietojen poiston järjestelmästä, mikä tapahtuu itse asiakirjan poiston yhteydessä.

Määrittelyissä hakemisto- ja rekisteröintipalvelu on kuvattu itsenäisenä palveluna niin, ettei hakutietojen rekisteröinnillä ole suoraa yhteyttä itse arkistoidun asiakirjan kanssa. Toisaalta voidaan ajatella, että tämän palvelun toiminnallisuus voitaisiin tarvittaessa liittää suoraan myös asiakirjojen arkistointipalveluun. Kuitenkin KANTA-palvelun yhtenä kantavana ideana on, että asiakkaan suostumus tarvitaan erikseen sekä tietojen hakuun että varsinaiseen asiakirjan luovutukseen.

4.3.3. Viestinvälitys

KANTA-järjestelmä ja perusjärjestelmät keskustelevat keskenään käyttäen viestinvälityspalvelun rajapintaa. Lisäksi KANTA-järjestelmän sisäiset palvelut käyttävät tätä samaa rajapintaa, tosin ne eivät kommunikoi keskenään suoraan, vaan jokainen neuvottelee viestinvälityksen osasen, palveluväylän kanssa. Sen tehtävänä on siis tarjota rajapinnat tiedon tuottajille, käyttäjille ja KANTA-järjestelmän osapalveluille. Sen toinen tehtävä on orkestroida palveluarkkitehtuurin mukaisesti KANTA:n osapalveluista (arkistopalvelu, haku- ja rekisteröintipalvelu jne) kokonaisuuksia ulkoisten perusjärjestelmien ja kansalaisportaalin käytettäväksi.

Varsinainen palveluiden orkestrointi tapahtuu palveluarkkitehtuurin mukaisesti niin sanotun prosessimoottorin avulla joka on siis osa tätä palveluväylää. Kaikki informaation käsittely ja kokonaisvaltainen järjestelmän ”äly” on tässä prosessimoottorissa. Karkeana analogiana ihmisruumiiseen palveluväylä on kuin hermosto, jota pitkin prosessimoottori (aivot) ohjaa elimiä (palvelut).

Viestinvälityksen tehtävänä on myös tarkastaa ja tarvittaessa monitoroida viestiliikennettä. Se siis kutsuu tunnistusta ja varmennepalvelua kun HL7-viesti

(mahdollisesti myös viestityypit, esim. DICOM) saapuu järjestelmään ip-verkon kautta. Toisaalta se myös huolehtii tarvittavasta siirtokehyksien purkamisesta ja lisäämisestä tiedon tullessa palveluväylälle ja lähtiessä sieltä.

Viestinvälityksen arkkitehtuurimäärittelyssä on kerrottu tavallisen toiminnallisuuden lisäksi mahdollisesta infrastruktuurista, johon palveluväylä sijoitetaan, ja on esitelty mahdollisia palomuuriratkaisuja ja tiedon suodatuksia. Viestinvälityksen arkkitehtuurimäärittelyissä on orkestrointi kuvattu selkeästi. Huomio kiinnittyy siihen, että vaikka dokumentissa KANTA-järjestelmän määritysten lausuntokierroksen tuloksista [STM, 2007f] annetaankin ymmärtää, että kaikista määrittelyistä pyritään poistamaan suorat sitoutumiset web service-teknologiaan, on viestinvälityksen arkkitehtuurimäärittelyssä suoraan sanottu toteutuksessa käytettävän web services-teknologian WS-security-standardia [STM, 2007d].

4.3.4. Tunnistus, varmennus ja sähköinen allekirjoitus

Tunnistus, tiedon salaus ja ylipäättään tietoturva ovat tärkeä osa KANTA-arkkitehtuuria, koko järjestelmään perustuu tiedon ja sen tuottajien ja käyttäjien oikeellisuuteen ja seurattavuuteen. Kuitenkin itse arkistoinnin ja sen käytön kannalta varmenteet, tunnistukset ja tiedon allekirjoitus ovat tukitoimintoja, joilla mahdollistetaan oikean tiedon talletus, käyttö ja luovutus. Sen takia tämän palvelukokonaisuuden sisältö kerrotaan tässä vain ylimalkaisesti.

Tunnistaminen ja allekirjoitus tapahtuvat henkilökohtaisilla toimikorteilla ja PKI-tunnistusta käyttäen. Valmiissa palvelukokonaisuudessa, missä KANTA-järjestelmä toimii perusjärjestelmän kanssa, on tunnistamisen ja sähköisen allekirjoittamisen logiikkaa kummassakin päässä. Käyttäjakohtainen tunnistaminen tapahtuu perusjärjestelmässä toimikortilla, joka on yhdistetty omaan salaiseen tunnukseensa. Näin jokaisen käyttäjän tekemä asiakirjamuutos tai haku saadaan tarvittaessa selvitettyä jälkikäteen. Tästä perusjärjestelmän tunnistamisvastuusta johtuen kansallisella järjestelmällä ei ole omaa käyttäjakohtaista tunnistusta tai käyttäjien oikeuksien rajausta.

PKI-toiminnallisuuteen perustuu, että kahden järjestelmään liittyneen osapuolen (tässä siis perusjärjestelmä ja KANTA) lisäksi on olemassa kolmas osapuoli, joka huolehtii varmenteista ja jota käyttäen kumpikin käyttävä osapuoli tietää vastapuolen olevan, kuka se väittääkin olevansa. Tänä kolmantena osapuolena eli varsinaisena varmennepalveluna (jolta perusjärjestelmä kysyy tunnistuksista ja käyttäjien varmenteista) toimii terveydenhuollon oikeusturvakeskuksen (TEO) ylläpitämä rekisteri.

Kansalaisen katseluyhteyden tunnistus on vielä auki, mutta ainakin sen pitäisi onnistua Väestörekisterikeskuksen tarjoamalla kansalaisvarmenteella. Tämän lisäksi tupas-tunnistamisella eli pankkiyhteyden mukaisella tunnistamisella voisi olla mahdollista katsoa omia tietoja. Määrittelyissä kuitenkin tunnustetaan, että se ei ole lain

mukainen vahva tunnistaminen. Ongelmana on se, että mahdolliset väärinkäytökset olisi erittäin vaikeaa todentaa [STM, 2007h].

4.3.5. Koodistopalvelu

Kansallinen koodistopalvelu on erillinen, itsenäinen palvelu KANTA-arkkitehtuurissa. Sen tarjoajana toimii suunnitellusti aluksi Stakes, mutta myöhemmin on tarkoitus, että Kela tarjoaa palvelun ja Stakes tuottaa vaan käsiteltävän tiedon.

Koodistopalvelu ylläpitää keskitettyä tietovarastoa sosiaali- ja terveydenhuollon digitaalisissa toimintajärjestelmissä hyödynnettävistä termistöistä, nimikkeistöstä ja muista luokitelluista koodistoista [STM, 2007j]. Toisaalta palvelu tarjoaa rajapinnat koodiston noutamiseen ja kyselyyn, mutta se myös mahdollistaa tietojen selailun ja ylläpidon.

Koodistopalvelu on oleellinen osa kokonaisarkkitehtuuria, koska sieltä löytyy myös KANTA-palveluiden ja perusjärjestelmien kommunikoinnin mahdollistavat palveluosoitteistot, ts. osoitteisto jolla mahdollistetaan osapuolten välisen viestinvälityksen reititys. Näin ollen koodistopalvelu tarjoaa SOA-arkkitehtuurissa puhutun ”tietoisuuden” palvelun tarjoajan (KANTA) ja käyttäjän (perusjärjestelmä) välillä [OASIS Group, 2007]. Vaikka määrittelyistä on pyritty riisumaan web services-teknologiaan tehdyt viittaavuudet, löytyy koodistopalvelun rajapinnan vaatimusmäärittelyistä palveluyksiköiden tiedoista kertova lista, jossa jokaiselle osapuolelle on annettu oma WSDL-määrittysten mukainen IP-osoite.

4.3.6. Suostumuksenhallintapalvelu

Suostumuksenhallintapalvelun tehtävänä on valvoa, että potilastietoja luovutetaan ainoastaan sellaisille palvelujenantajille, joilla on siihen potilaan antama suostumus. Järjestelmässä voi olla tallessa kansalaisella suostumus- ja/tai kieltoasiakirjoja. Erityistapauksista, joissa luovutuksia saa kiellosta huolimatta tehdä, on kirjattu tarkennukset lakiin sosiaali- ja terveydenhuollon asiakastietojen sähköisestä käsittelystä 13 §:n 3 momenttiin. Kysymyksessä ovat lähinnä tilanteet, joissa potilas on itse estynyt (esim. tajuton) antamaan suostumustaan. Suostumukset ja kiellot voivat koskea joko yksittäistä palvelun tuottajaa, asiakirjaa tai palvelukokonaisuutta.

4.3.7. Loki- ja valvontapalvelu

Loki- ja valvontapalvelun tehtäviin kuuluu seurata ja pitää lokia KANTA-järjestelmän tapahtumista. Palvelussa ei ole kyse vain tavallisista vikalokeista, vaan sillä on monia tehtäviä, joilla seurata KANTA-järjestelmän käyttöä. Yhtenä oleellisimmista tehtävistä on pitää kirjaa arkistoon talletettujen asiakirjojen luonneista, niiden muutoksista (jotka ovat siis itsessään uusia asiakirjoja) ja luovutuksista. Näiden lisäksi lokipalvelu pitää kirjaa myös tietojen hauista, kansalaisella kun on oikeus tietää, mitä tietoja hänestä on olemassa, mitä niistä on haettu ja luovutettu. Lisäksi terveydenhuollon työntekijöiden

oikeusturvan kannalta on oleellista, että pystytään selvittämään tarkasti, mitä tietoja oli käytettäessä hoitoa annettaessa tai tehtäessä hoitoon liittyviä päätöksiä.

Loki- ja valvontapalvelussa tehdään tilastointia data- ja käyttäjämääristä ja tarvittaessa luovutetaan näitä tietoja mahdollisille muille sidosryhmille. Lisäksi siinä toimii myös hälytys- ja raportointipalvelu, jolla pitäisi saada koko KANTA-järjestelmän vikatilanteet nopeasti informoiduksi eteenpäin [STM, 2007i].

4.3.8. Kansalaisen katseluyhteys

Kansalaisen katseluyhteyden avulla on tarkoitus mahdollistaa asiakkaille heidän omien potilastietojensa katselu tiettyynajaan asti suoraan esim. selainliittymällä. Tämän toteuttamiseksi tarvittaisiin vahvaa tunnistusta käyttävä tietoturvaratkaisu ja selain, jolla on pääsy internetiin. Katseluyhteyden on tarkoitus tulla käyttöön ensimmäisten KANTA-palveluiden joukossa, mutta selvittettäviä asioita riittää vielä useita, esim. mihin tietoihin asiakkaalla tulisi olemaan pääsy ja kuka tekisi hänen potilastietojensa suodatuksen. Tarkoitus kun ei ole koskaan näyttää asiakkaalle kaikkia hänen tietojaan vaan noudattaa tässäkin tapauksessa lakia asiakastietojen sähköisestä käsittelystä [L159, 2007]. Kansalaisen katseluyhteys on varsin ulkopuolinen osio KANTA-kokonaisuudessa, mutta se haluttiin lisätä ensimmäisten toteutettavien palveluiden joukkoon epäilemättä herättämään kansalaisten kiinnostus asiaan ja toisaalta osoittamaan konkreettisesti, että hallinnon sähköistyvät palvelut ovat tulossa myös terveydenhuollon järjestelmiin. Tämä taas on oleellinen osa eHealth-konseptia.

4.4. Viitetietokannat hajautettuna tietojärjestelmänä

Alueelliset viitetietokannat voidaan nähdä kokonaisuutena, johon kuuluu useita itsenäisiä tietojärjestelmiä, joilla on myös yhteistoiminnallisuutta niiden välillä. Siinä mielessä viitetietokannoista puhuttaessa termi ”hajautettu tietojärjestelmä” on harhaanjohtava, että tässä tapauksessa tiedon hajautus ylittää niin maantieteelliset, organisaatio- kuin sovellustasot. Se on siis jo yhden integraatioprosessin tuotos.

Viitetietokantajärjestelmää voi pitää hajautuksen näkökulmasta juuri heterogeenisyyden ja yhteistoiminnallisuuden ansiosta liitettyinä tietokantana (federated database system). Tosin termi voidaan tiukasti rajattuna ajatella koskevan vain systeemeitä, missä kaikilla on suoraan sovelluksesta läpinäkyvä pääsy moniin jaettuihin, yhteiskäyttöisiin tietokantoihin yhtä aikaisesti [Sheth and Larson, 1990]. Viitetietokantojen takana on kuitenkin ajatus, että suostumuksen saatuaan voidaan potilaalle hakea ensin viitekannasta löytyvät viitteet ja tämän jälkeen tehdä kysely jokaiseen kantaan erikseen. Toisaalta liitännäisten tietokantojen käsite on saanut ajan myötä uusia muotoja eikä näin jyrkkiä rajauksia välttämättä enää tehdä [Stumptner *et al.*, 2004]. Oleellinen seikka, jolla voidaan päätellä viitetietokantojen kuuluvan nimenomaan liitettyihin tietokantoihin on, että jokainen perusjärjestelmä on itsenäinen tietojärjestelmä tietokantoinen, mutta kuitenkin viitetietojärjestelmän avulla

ulkopuolisilla käyttäjillä on mahdollisuus hakea sieltä tietoa. Viitetietokantajärjestelmä on siis, luvussa 3.3.3 kuvatuunlainen hybridirakenne, keskitettyjen tietojärjestelmien (perusjärjestelmät) hajautettu yhteenliittymä [Elmasri and Navathe, 2000].

Varsinainen viitetietokanta sisältää viittauksia ja niiden hakuja ohjaa ja valvoo viitetietojärjestelmä [Nykänen ja Karimaa, 2002a] joka voidaan nähdä vastaavan kirjallisuudessa esiintyvää liitännätietojärjestelmää (federated information system) [Stumptner *et al.*, 2004] [Wyss and Van Gucht, 2001]. Viitetietojärjestelmän kanssa neuvottelevat perusjärjestelmäadapterit voidaan nähdä Wyssin ja Van Guchtin [2001] mainitsemaksi osaksi hajautettua liitännätietojärjestelmää. Näin siksi, että perusjärjestelmäadaptereilla on varsin paljon valtaa ja toisaalta tehtäviä viitetietokantajärjestelmässä: Ne toisaalta kokoavat viitteet palvelutapahtumista ja siirtävät ne viitetietojärjestelmälle ja toisaalta hakevat pyydetyn tiedon viitteen avulla perusjärjestelmästä viitetietojärjestelmälle [Nykänen ja Karimaa, 2002a]. Erityisesti tiedon haku viitteen perusteella on oleellinen toiminto, koska perusjärjestelmäadapteri myös muuntaa asiakirjan HL7 CDA-muotoon, jollaisena se on nähtävissä sitten viitetietojärjestelmän tarjoamassa selainkäyttöliittymässä. Näin vältetään kirjallisuudessa usein korostettu tietojärjestelmäintegraatiossa piilevä riski semanttisista konflikteista [Sheth and Larson, 1990] [Stumptner *et al.*, 2004] [Beyer *et al.*, 2004].

Kuten luvussa aikaisemmin todettiin, viitetietokannat eivät ole läpikäyneet täyttä sovellusintegraatiota. Niille ei ole tähän päivään mennessä tehty suoraa liityntää perusjärjestelmistä, vaikka tällainenkaan ratkaisu ei varmasti olisi mahdoton. Tämä suora integraatio on kuitenkin se, mitä KANTA-arkkitehtuuri vaatii perusjärjestelmiltä.

Perusjärjestelmien adapterien ja viitetietojärjestelmän viestinvälitys perustuu HL7-standardin Suomen laajennoksen mukaiseen Avoimet rajapinnat-määritykseen. Sanomat ovat web services-ohjelmistojärjestelmän suojattuja SOAP-sanomia, joita välitetään suojatun http-yhteyden yli [STM, 2007d]. Sanomaliikenne on kuitenkin aina etukäteen määriteltyä kahdensivelistä liikennettä. Sen ei voida siis sanoa noudattavan palvelukeskeistä arkkitehtuuria, vaikka se sisältääkin usein palvelukeskeisessä arkkitehtuurissa käytettäviä elementtejä.

4.5. KANTA-arkkitehtuuri keskitettynä tietojärjestelmänä

Keskitetty kansallinen potilastietojen arkisto on muutamassa käytäntöä koskettavassa mielessä ylivoimainen hajautettuihin ratkaisuihin verrattuna. Ensinnäkin vastuukysymykset ovat keskitetyssä KANTA-palvelussa selkeitä; Kela vastaa KANTA-järjestelmästä ja sen tiedoista ja toisaalta siihen liittyneet yksiköt omista järjestelmistään (tietojärjestelmän toimittajan kanssa) ja tiedoista. Näin jokaisen perusjärjestelmän rakentaminen yhteensopivaksi KANTA-järjestelmän kanssa ja liittäminen siihen ei jää yhden toimittajan vastuulle, vaan jokaisen järjestelmätoimittajan täytyy itse muokata järjestelmänsä. KANTA vain tarjoaa rajapinnan siihen, miten sen kanssa toimitaan. Etuna tämä on suuri, koska erilaisia

terveydenhuollon tietojärjestelmiä on useita ja kaikki ne ovat rakennettu omien periaatteidensa mukaisesti. Sosiaali- ja terveystieteiden tutkimuskeskuksen selvityksessä [STM, 2006] on todettu, että hajautettu, yhteistoiminnallinen kansallinen kokonaisuus olisi myös kalliimpi ratkaisu kuin keskitetty tietojärjestelmä. Integraatiotarve hajautetussa, runsaan 500 erillisen terveydenhuollon toimiyksikön kokonaisuudessa olisi suunnaton. Keskitettynä ratkaisuna KANTA mahdollistaa kustannustehokkaan pitkäaikaistalutuksen sähköisesti allekirjoitetuille potilastiedoille.

KANTA noudattaa hyvin tarkkaan data warehousing-mallia eli tietovarastomallia, jonka Grimson *et al.* [2000] on esitellyt integroinnin läpikäyneelle keskitetylle terveydenhuollon tietojärjestelmäkokonaisuudelle ja joka esiteltiin jo luvussa 3.1.4. Grimson *et al.* [2000] kuvailee tietovarastomallissaan keskitetyn tietovaraston, johon yksittäiset järjestelmät ovat integroituneet ja jonka sisältö on vahvasti homogenisoitu (KANTA-arkkitehtuurissa on HL7-asiakirjoja). Kun asiakirja on syötetty tietovarastoon, säilyy se edelleen myös liittyneen järjestelmän tietokannassa. Aivan kuten KANTA-arkkitehtuurin mukainen sähköinen arkisto todellisuudessa tulee olemaan, näkee myös Grimson *et al.* [2000] tietovarastomallin yksisuuntaisena eli liitäntäjärjestelmistä tietovarastoon syötetty data on myöhemmin lähinnä vain luettavissa sieltä. KANTA-arkkitehtuurissa tämä tosin on hyvin oleellinen seikka, kuten jo aikaisemmin kerrottiin.

On syytä huomata, ettei KANTA kuitenkaan ole sanan varsinaisessa merkityksessä suora keskitetty tietojärjestelmä. Arkkitehtuurissa on kaksi seikkaa, jotka tekevät siitä enemmänkin hybridi-mallisen tietokantaratkaisun:

- KANTA-arkkitehtuurin riippuvuus perusjärjestelmistä
- KANTA-arkkitehtuurin käyttämä palvelukeskeinen arkkitehtuuri.

Voidaan ajatella, että vaikka KANTA onkin itsenäinen tietojärjestelmä, ei se ole kokonaisvaltainen sellainen ilman siihen liittyneitä perusjärjestelmiä. Tämä linjaus voidaan tehdä, koska KANTA-tietojärjestelmällä (arkistolla) ei ole olemassa suoraa käyttöliittymää, jolla syöttää siihen tietoa (Kansalaisen katseluyhteys, kuten nimestäkin voi päätellä, ei mahdollista tietojen syöttöä). Eli kokonaisvaltaiseen tietojärjestelmäkokonaisuuteen kuuluu myös perusjärjestelmät, jotka taas ovat itsenäisiä tietojärjestelmiä. Tämän vuoksi KANTA on keskitetty vain sovellusintegraation näkökulmasta. Aier ja Schönherr [2006] kutsuvat tällä tavalla integroitua järjestämää keskitetyksi hub&spoke-malliksi eli pyörän napa ja pinna-malliksi.

Toinen seikka, mikä tekee KANTA-arkkitehtuurista enemmän hybridin tietomalliltaan, on sen käyttämä palvelukeskeinen arkkitehtuuri. KANTA-järjestelmän tapauksessa palvelukeskeinen arkkitehtuuri on enemmän helpottamassa perusjärjestelmien integraatioprosessia ja mahdollista KANTA-järjestelmän palveluiden laajennettavuutta kuin itse tietoprosessien ja -virtojen hajautusta. Tämä on tosin

ymmärrettävää juuri keskitetyn arkkitehtuurin näkökulmasta toimivassa, Kelan rakentamassa ja ylläpitämässä KANTA-järjestelmässä.

Aier ja Schönherr [2006] pitävät normaalina ellei jopa todennäköisenä, että todelliset sovellusintegraatiota läpikäyvät tietojärjestelmät voivat olla arkkitehtuuriltaan hybridejä, mutta heidän mukaan tästä huolimatta tutkimuksissa voidaan käyttää karkeampia jaotteluita. He näkevät että heidän tutkimuksessaan esiteltyjä jaotteluilta voidaan käyttää apuna itse integraatioprosessia suoritettaessa tai sitä analysoitaessa, aivan kuten tässä tutkimuksessa tehdään.

Viimeisenä sivuhuomiona voidaan miettiä termiä hybridi tietojärjestelmäarkkitehtuuri; Yleensä sillä tarkoitetaan keskitetyn ja hajautetun tietojärjestelmäarkkitehtuurin yhdistämistä, mutta KANTA-arkkitehtuurissa onkin piirteitä sekä SOA- eli palvelukeskeisestä että keskitetystä arkkitehtuurista. SOA-arkkitehtuuri ei vaadi tietojärjestelmäarkkitehtuurilta hajautusta tai keskitystä, koska prosessimoottorin toteutus ratkaisee sen, mistä mahdollisesti dynaamisesti valittavat palvelut löytyvät.

5. Tulokset

Tässä luvussa käsitellään tutkimuksen tuloksia tietojärjestelmäkehityksen kannalta. Ensin kuvaillaan hiukan periaatteita, kuinka hajautetut ja keskitetyt tietojärjestelmät voitaisiin mahdollisesti liittää yhteen ja toisaalta kuinka palveluarkkitehtuuri voi tukea näiden kahden tietojärjestelmäparadigman mukaisen järjestelmän integraatiota. Lopuksi annetaan karkeita, esimerkinomaisia käyttötapoja siitä, kuinka viitetietokantoihin perustuvat tietojärjestelmät voisivat toimia yhteen KANTA-arkkitehtuurin kanssa.

5.1. Viitetietokantojen ja KANTA-arkkitehtuurin yhdistämisestä

Kun mietitään viitetietokantojen ja KANTA-arkkitehtuurin mahdollista integrointia keskenään, voidaan lähtökohdaksi ottaa aikaisemmin esitelty KANTA-määrittelyistä löydetty vaatimukset viitetietojärjestelmille. Löydettyistä vaatimuksista on mahdollista tulkita kaksi erilaista lähestymistapaa sovellusintegraatiolle.

Viitetietokannan muuttaminen toimimaan KANTA-palvelun ja perusjärjestelmien välissä ei luultavasti ole mahdotonta, mutta tutkimuksen tuloksissa analysoidaankin lähinnä sen mielekkyyttä. Toinen vaihtoehtoinen lähestymistapa voisi olla näkökulma, jossa viitetietokanta toimisi edelleen erillisessä roolissaan niin, että perusjärjestelmät ottavat yhteyttä sekä valtakunnalliseen KANTA-palveluun että sairaanhoitopiirin viitetietokantaan. Tätäkin mallia analysoidaan tutkimuksen tuloksissa tarkemmin.

Huomioitavaa on, että toisaalta viitetietokannat toteuttavat toimintoja, joita KANTA:an ei joko voida toteuttaa (esimerkiksi resurssointitoiminnot) tai joiden toteuttaminen tapahtuu vasta määrittelemättömässä tulevaisuudessa (ajanvarausten teko ja tietojen tallennus). Huomioitavaa on myös, että viitetietokannat toimivat tavallisesti omissa suljetuissa verkoissaan.

Koko sovellusintegraatiota mietittäessä on syytä sopia tarkasti vastuukysymykset. Sekä KANTA-järjestelmässä että viitetietokantojen kokonaisuuksissa on monia eri osapuolia, on niin perusjärjestelmien käyttäjäorganisaatioita, ohjelmistojen toimittajia ja viitetietokantojen ja KANTA-arkiston ylläpitäjiä. Näiden kaikkien pitää tietää roolinsa ja vastuunsa, mikäli KANTA ja viitetietokannat yritetään yhdistää. Todennäköisesti vaihtoehdoksi tulee tilanne, jossa viitetietokannat ovat kuten mikä tahansa perusjärjestelmä, joka liittyy KANTA:aan. Tällöin siis KANTA-järjestelmän toiminnallisuutta ei mitenkään erikseen muuteta toimimaan viitetietokantojen kanssa yhteen. Tässä vaihtoehdossa vastuukysymykset yksinkertaistuvat, varsinkin verrattuna tilanteeseen, jossa palveluita ulkoistettaisiin viitetietokantajärjestelmiin. Usean toimijan kokonaisuudesta seuraa vastuultaan sekava tilanne, joka voi olla vaarallista potilasturvan kannalta.

5.1.1. KANTA-arkkitehtuurin ja viitetietokantojen yhdistäminen sovellusintegraation näkökulmasta

Tässä luvussa käsitellään KANTA-arkkitehtuurin ja viitetietokantojen sovellusintegraatiota siten, että ensin analysoidaan Bayer *et al.*:in [2004] tunnistamia yhteensopivuuden ongelmatilanteita ja sen jälkeen sovelletaan niiden ilmenemistä tässä integraatiotapauksessa. Lisäksi käydään läpi Linthicumin [2000] 12-kohtaista listaa sovellusintegraation prosessista. Bayer *et al.* [2004] tutkimus valittiin sen takia, että siinä esitetyt yhteensopivuusseikat ovat linjassa terveydenhuollon tietojärjestelmien yleisten integraatiopulmien kanssa. Toisaalta Linthicumin [2000] lista taas on niin konkreettinen, että voidaan miettiä sen soveltuvuutta tähän tapaukseen. Sovellusintegraation mahdollisuuksia ja käytäntöjä käydään yksityiskohtaisemmin läpi myöhemmin tulosten esittelyssä.

Beyer *et al.* [2004] esitteli sovellusintegraatiolle terveydenhuollon tietojärjestelmien kolme tärkeää yhteensopivuutta: datan rakenteisuuden (syntaksi), tyyppitason semanttisuuden (ontologiataso) ja instanssitason semanttisuuden (terminologia). Näistä kaksi ensimmäistä yhteensopivuutta on varsin yksikertaista saavuttaa, koska sekä viitetietokannat että KANTA-arkkitehtuuri perustuvat HL7-sanomille ja CDA-asiakirjoille. Kaksi asiaa, joihin täytyy kiinnittää huomiota puhuttaessa tyyppitason semanttisuudesta ovat sisäiset tietomallit ja CDA-asiakirjan versiot.

Sisäisillä tietomalleilla tarkoitetaan tässä tapauksessa sitä, miten asiakirjojen sisältö on sisäisesti käsitteellisesti mallinnettu, mikä on niiden tiedon ontologia. Mahdollisia tietomalleja ei voida vertailla, koska KANTA-kokonaisarkkitehtuurin arkkitehtuurimäärittelyssä [2007b] tietomalli on esitelty luonnosmaisesti ja toisaalta ainakaan Nykäsen ja Karimaan [2002b] mukaan Satakunnan Fiale-kannan määrittelyissä ei ole esitelty käsitteellistä tietomallia. Kuitenkin, kuten todettiin, sisäisten tietomallien erot eivät ole suuri ongelma, jos viestinvälitys tapahtuu standardoiduilla sanomilla.

Isompi ongelma kuin sisäisten tietomallien erot on se, että viitetietokannat käyttävät HL7 CDA R1-asiakirjoja kun KANTA käyttää asiakirjan R2-versiota. Versioero johtaa kuitenkin tilanteeseen, jossa KANTA ei tue R1-sanomaa, koska siitä puuttuu rakenteisuus body-osalle. Toisaalta, koska potilasjärjestelmien pitää alkaa tukea CDA R2-versiota KANTA-kommunikoinnissaan, voidaan olettaa, että myös viitetietokannat siirtyvät käyttämään tätä versiota.

Instanssitason yhteensopivuus tarkoittaa yhteisien koodistojen (esimerkiksi ICD-koodisto) käyttöä tietojärjestelmästä riippumatta. Tämän ei pitäisi tuottaa vaikeuksia, koska tarkoitus on, että koodistot pidetään ajantasalla.

Linthicum [2000] esittelee 12 kohtaisen listan toimista, joita pitää käydä läpi sovellusintegraationprosessia läpikäydessä. Vaikka lista onkin enemmän organisaation sisäisten tietojärjestelmien integrointiprosessia koskettava, voidaan sitä käyttää tässä

apuna. Jos KANTA-arkkitehtuurin ja viitetietokantojen mahdollista sovellusintegraatioprosessia mietitään tämän listan kautta, voidaan tehdä muutamia havaintoja. Kuten todettiin, kuvaa Linthicumin lista enemmän kertaluonteista sovellusintegraatioprosessia kuin jatkuvaa tai kierroksittain suoritettavaa sovellusintegraatiota. Lisää ongelmia voidaan olettaa aiheutuvan siitä, että integraatioprosessi saattaa olla enemmänkin yksipuolista eli viitetietokantojen muokkaamista KANTA-arkkitehtuuria käyttämään eikä niinkään kaksisuuntaista yhteistyötä. Myöskään toinen KANTA-viitetietokanta-integraation erityispiirre eli tilanne, jossa on kaksi osittain päällekkäistä tietojärjestelmää, ei ole mitenkään mukana Linthicumin [2000] listassa. Seuraavaksi käydään kuitenkin läpi kohta kohdalta listaa ja sen vastaavuutta KANTA-arkkitehtuurin ja alueellisten viitetietokantajärjestelmien sovellustason integraatiossa.

1. Johtuen organisaatioiden eroista, on listan ensimmäinen kohta eli kohdealueen liiketoiminnan ja integraatio-ongelman tunnistus vain hankalaa, kun integraatioprosessi ylittää organisaatioiden rajat.
2. Myös kohta kaksi eli tietojen ja niiden tietovirtojen tunnistaminen on hankalampaa kuin tavallisen, sisäisen integraatioprosessin tapauksessa. Eri organisaatioissa voi hyvin olla erilainen näkemys ja käsitteistö (organisaatiotason ontologia) kohdealueestaan ja sen ydintiedoista. Tämä pätee erityisesti ajateltaessa organisaation henkilökuntaa tietojärjestelmien käyttäjinä.
3. Kolmas kohta liiketoimintaprosessien analysointi ja ymmärtäminen on myös haastava tehtävä. Ongelmaa ei helpota se, että viitetietokannat hajautettuna systeeminä on hankala hahmotettava, sen yleisiä toiminnallisia prosesseja on hankala seurata, kun esimerkiksi osa operaatioista on selainpohjaista tietojen etsintää ja noutamista, osa tietoturvaan liittyvää toimintaa ja osa perusjärjestelmiin liitettyjen adapterien toimintaa.
4. Neljäs kohta Linthicumin [2000] listassa on rajapintojen tunnistaminen. Tämä on harvoja lähes selkeitä tapauksia, koska kumpikin integraatioprosessin osapuoli käyttää jo yhteistä viestinvälitysrajapintaa eli HL7-asiakirjoja. Vaikka tietomalleissa voisikin olla eroja, on tämä yhdistävä standardi kuitenkin luotu juuri yhteistoiminnallisuuden tarpeisiin ja sopii sen vuoksi hyvin tähän integraatioprosessiin. Sovellusintegraatio voisi helpottaa vielä huomattavasti, jos KANTA-arkkitehtuurin palvelukeskeinen arkkitehtuuri toteutetaan nimenomaan Web services-palveluna, koska tällöin voidaan käyttää Suomen HL7-organisaation jo aikaisemmin tekemiä laajennoksia HL7-standardiin.
5. Viides kohta eli liiketoimintatapahtumien tunnistaminen ja tiedon tallennuksen periaatteet on haastava tehtäväkokonaisuus ja tämä voidaan

nähdä listan ensimmäisen kohdan jälkeen tehtyjen vaiheiden avulla kerättyjen tietojen kootuksi tuotokseksi.

6. ja 7. Kohdat kuusi ja seitsemän helpottuvat huomattavasti siitä syystä, että kummallakin osapuolella on käytössä HL7-standardi. Kuitenkin esimerkiksi KANTA-arkkitehtuurin koodistopalvelun ja viitetietokantojen omien koodistojen kanssa voi olla mahdollisia eroja.
8. Kohdassa kahdeksan eli sovellusintegraation teknologioiden valinta ja varsinainen integraation tekeminen ovat vaikeita, koska viitetietokannoillekin on useampia toimittajia, jotka eroavat toki hiukan toisistaan. Tähän seikkaan vaikuttaa se, ettei KANTA-arkkitehtuuria luotaessa yhteensopivuutta viitetietokantojen kanssa ajateltu kovin tarkasti.
9. Kohta yhdeksän eli sovellusintegraation testaus on tärkeä prosessi kokonaisuuden kannalta. Testausurakkaa helpottaa hiukan se, että asiakkaiden sairaskertomukset eivät muuta sisältöään, vaan uusia tietoja lisätään vanhan perään. Näin tarvittavien vaihtoehtojen testaus vähenee huomattavasti.
10. Suorituskyvyn arviointi on tärkeää, ovathan tehokkuusvaatimukset korkeat jo itse KANTA-järjestelmälle, jossa käyttäjiä ja operaatioita on paljon. Toisaalta vaikka viitetietokannat voivat tuoda uutta, yleistä tiedon käyttöä ja raskautta lisäävää toimintaa, on tässä tutkimuksessa luvussa 5.2 tutkittu myös mahdollisuuksia, joilla viitetietokannat voisivat lisätä asiakastietojen etsinnän ja katselun kokonaisvaltaista operatiivista tehokkuutta.
11. Sovellusintegraatioprosessin jälkeisen hyödyn arviointi on mahdollista tietenkin vasta sitten kun itse prosessi on käyty läpi. Mikäli sovellusintegraatiota aiotaan suorittaa, voi se tapahtua vaiheittain yhdistellen eri toimintoja, joita KANTA-järjestelmästä ja viitetietokannoista löytyy. Sen takia sovellusintegraation hyötynäkökulmat voivat muuttua sen mukaan, mitä KANTA-järjestelmään on implementoitu ja mitä hyötyä viitetietokantojen vastaavan ominaisuuden käytöstä saa.
12. Kohta 12 käsittelee sovellusintegraation läpikäyneen kokonaisuuden ylläpitoa ja vastuukysymyksiä. Tämä on erittäin oleellinen kysymys mieltä jo huomattavasti aikaisemmin kuin vasta prosessin lopuksi. Varsinkin tällaisessa tapauksessa, jossa sovellusintegraation tulokset muodostuvat usean eri osapuolen yhteisen ponnistelun tuloksena.

Linthicumin [2000] esittelemästä listasta voi olla apua mietittäessä asioita, joita pitää sovellusintegraatioprosessissa huomioida. Kuten aikaisemmin todettiin, on se luonteeltaan enemmän organisaation sisäiseen sovellusintegraatioon sopiva kuin monen osapuolen järjestelmän liitoksiin.

5.1.2. Tietojärjestelmien integraation mahdollisuudet palvelukeskeisen arkkitehtuurin näkökulmasta

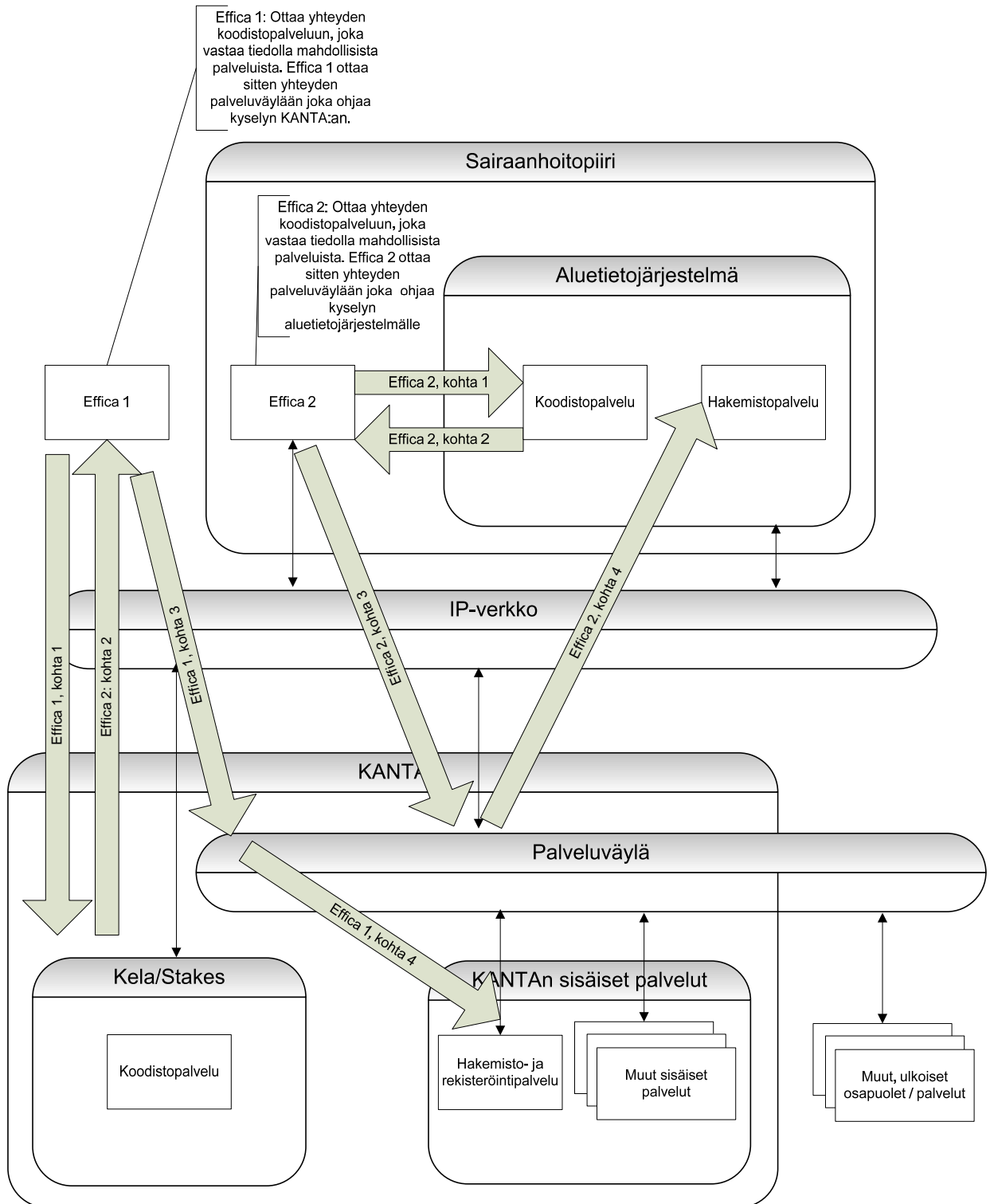
Aikaisemmin kerrottiin, että palvelukeskeinen arkkitehtuuri mahdollistaa toiminnallisuuden hajautuksen pienemmiksi osiksi, palveluiksi. Tämä taas periaatteessa mahdollistaa useita erilaisia ratkaisuja, joilla viitetietokannat ja KANTA-tietojärjestelmä saataisiin toimimaan yhteen. Ongelmana useissa ratkaisuisa on se, että KANTA-määritykset ja laki eivät mahdollista sellaisia ratkaisuja, joissa KANTA- ja viitetietokannat ovat tasavertaisina tiedon säilöjinä.

Yksi vaihtoehto, jolla palvelukeskeisestä arkkitehtuurista voisi olla hyötyä sovellusintegraatiossa, olisi sairaanhoitopiirin alueelta tulevien palveluiden kutsujen erottaminen toimimaan saman alueen viitetietokannoissa KANTA-palvelun sijaan. Tässä tapauksessa KANTA:n palveluväylä orkestroisi viitetietokannan tarjoamia palveluita.

Tässä tapauksessa pitäisi palvelun tarjoajan tietää jotain käyttäjästä tai käyttäjän tietää palvelun tarjoajasta, jotta kutsut voitaisiin ohjata vaihtoehtoisesti joko KANTA-palvelulle tai jollekin viitetietokannalle, riippuen esimerkiksi maantieteellisestä sijainnista, josta palvelukutsu tulee.

Esimerkinomaisesti kuvassa 3 on esitelty tilanne, jossa kaksi terveydenhuollon perusjärjestelmää, Effica1 ja Effica2, ovat kiinni KANTA-järjestelmässä niin, että toinen on siinä itsenäisesti suoraan kiinni ja toinen on kiinni sekä KANTA-järjestelmässä että sairaanhoitopiiriin viitetietojärjestelmässä. Kun suoraan kiinni oleva palvelu, Effica1, ottaa yhteyden KANTA:an, se hakee mahdollisesti tietoa palveluista (tai palveluosoitteistosta) koodistopalvelulta ja ottaa tämän jälkeen yhteyden KANTA-järjestelmän palveluväylään, joka ohjaa asiakirjahaun hakemisto- ja rekisteröintipalvelulle. Viitetietokannoissa kiinni oleva perusjärjestelmä, Effica2, taas voisi mahdollisesti ottaa yhteyden oman alueensa koodistopalveluun, joka ohjaisi palvelukutsun KANTA-järjestelmän palveluväylälle. Palveluväylä taas tunnistaisi dynaamisesti asiakirjahaun tulevan aluetietojärjestelmään liittyneeltä perusjärjestelmältä ja voisi ohjata tämän kyselyn alueelliseen viitetietokantaan, joka suorittaisi samantasoisien palvelun kuin KANTA itse. Etuina voisi olla esimerkiksi nopeus KANTA-järjestelmään verrattuna ja KANTA-arkkitehtuurin kuormituksen vähentäminen. Tästä ei kuitenkaan ole mitään takeita, mutta esitys on tässä vain esimerkkinä.

Palvelukeskeinen arkkitehtuuri ei rajaa erityisemmin mitään vaihtoehtoa pois sovellusintegraation suorittamisessa. Päinvastoin, pienistä palveluista koostuva suurempi palvelukokonaisuus voi olla sekä tehokas että monikäyttöinen tietojärjestelmäratkaisu [Aier and Schönherr, 2006]. KANTA-arkkitehtuurin tapauksessa rajoitukset tulevatkin enemmän muualta, kuten lakitekstistä, kuin teknologiasta tai sovellusarkkitehtuurista.



Kuva 3: Hakemistopalvelun käyttö KANTA-arkistosta ja toisaalta viitetietokannasta.

5.2. Mahdollisia käyttötapoja

Tässä kokonaisuudessa on esiteltyä muutamia esimerkin omaisia käyttötapoja, joilla viitetietokantoihin perustuvia aluetietojärjestelmiä voisi mahdollisesti käyttää terveydenhuollon kansallisen sähköisen arkiston, KANTA, kanssa. Jotkut esimerkkien

toiminnallisuuksista ovat jo valmiina nykyisissä viitetietokannoissa, joten näille arvioidaan vain niiden mahdollista tarvetta KANTA-arkkitehtuurin kanssa toimittaessa ja toisaalta mahdollisia muutostöitä, joita niille yhteensopivuuden toteuttamiseksi pitäisi tehdä. Edellisessä luvussa kuvailtiin keinoja, joilla palvelukeskeinen tietojärjestelmäarkkitehtuuri voisi auttaa sovellusintegraatiossa KANTA-järjestelmän ja viitetietokantojen välillä. Siinä esitellyt keinot pätevät myös erilaisiin käyttötapoihin, jotka löytyvät sekä KANTA:sta että viitetietokannoista.

Vaihtoehdot käyttötavoissa voidaan jakaa usealla eri tavalla. Teknisessä mielessä valittavissa on kolme erilaista vaihtoehtoa:

- Ratkaisu, jossa viitetietokannat ovat erillään KANTA-arkkitehtuurista niin, että niistä voi olla rakennettuna yhteys KANTA:an, mutta niillä ei kuitenkaan ole suoraa toiminnallisuutta, vaan viitetietokantaan liittyneet perusjärjestelmät käyttävät KANTA:a suoraan omalla liittymällään.
- Ratkaisu, jossa alueelliset viitetietokannat ja KANTA toimivat yhdessä niin, että viitetietokannat voisivat ottaa osan KANTA-arkiston tehtävistä esimerkiksi vähentääkseen sen kuormitusta.
- Ratkaisu, jossa viitetietokannat ovat perusjärjestelmien ja KANTA-arkiston välissä toimien KANTA:an päin kuin perusjärjestelmä ja perusjärjestelmiin päin kuin KANTA.

Kuva 4 kuvaa tilannetta, jossa sairaanhoitopiirin viitetietokannan integrointi tehtäisiin KANTA-tietojärjestelmän ja perusjärjestelmien väliin. Kuvassa 5 on esitetty integraatioprosessin vaihtoehto, jossa viitetietokannat jätetään erilleen perusjärjestelmien ja KANTA-palvelun integraatiosta ja viitetietokannoille on rakennettu oma liittymä KANTA:an. Eri integraatiotilanteiden kuvien kaaviot ovat yksinkertaistettuja versioita kuvasta 2 paitsi viitetietokannan ja sen sijainnin osalta.

Vaikka jotkut esimerkkien toiminnallisuuksista ovat varsin suuritöisiä tai peräti utopistisia, on ne kuitenkin kirjattu tähän. Toisaalta monet esimerkeistä ovat sellaisia, etteivät ne ole ristiriidassa keskenään. Toisin sanoen, monet toiminnallisuudet voitaisiin sisällyttää viitetietokantaan. Jokaisessa näistä vaihtoehdoista on oletuksena, että viitetietokannasta on jotain todellista hyötyä verrattuna KANTA:an, kuten että koko järjestelmän suorituskyky on parempi. Kannattaa kuitenkin huomata, että esimerkiksi resurssointi ja lähete-palaute-ratkaisut ovat täysin viitetietokantakohtaisia kun KANTA-palvelussa ei ole ajanvarausasiakirjoja tai mitään resurssointitietoja. Samaten esimerkiksi suunniteltu hoito- ja palveluketjujen ylläpito puuttuu myös. Kuka nyt näkee tai hoitaa asiakkaan palvelukokonaisuuden kulkua KANTA-tietojärjestelmässä?

Mikäli viitekannoissa on jo palvelukokonaisuuksien hoitoa ja ylläpitoa mahdollistavia seikkoja, ovat ne varmasti vielä tarpeellisia. Viitetietokantojen lokipalvelu voi menettää merkitystään, mikäli viitetietokantojen kautta kulkevan datan määrää muutellaan. Samaten esim. Makropilotissa mukana olleet mahdolliset data

warehouse-ratkaisut menettävät merkitystään. Toisaalta mikäli tilastoja tai johtamisjärjestelmiä halutaan, voisi näihin harkita jotain agentti- tai eräajopalvelua, joka hoitaisi tietomassojen keräyksen (olettaen että viitetietojen kerääminen lopetetaan).

Kun viitetietokantoja käytetään tällä hetkellä selaimen avulla, on se erillinen prosessointi verrattuna siihen, että KANTA-palvelua on tarkoitus käyttää suoraan terveydenhuollon perustietojärjestelmistä. Nämä kaksi erilaista käyttöliittymää (voisi sanoa lähestymistapaa) eroavat jokapäiväisessä käytössä toisistaan huomattavasti, kun selainpohjainen viitetietokantajärjestelmän ratkaisu on aina erillinen ohjelmisto perusjärjestelmien rinnalla.

Seuraavissa luvuissa esitellään erilaisia ratkaisuja ja toiminnallisuuksia, joihin alueellisia viitetietokantoja voisi käyttää silloinkin kun KANTA-arkkitehtuuri on jo käytössä. Näistä esitellyistä ratkaisuista ainoastaan kohdissa 5.2.8 ja 5.2.9 lähdetään siitä, että viitetietokantoihin tehtäisiin suurempia muutoksia niin, että niiden rooli ja käyttötapa todella muuttuisi radikaalisti. Tällöin voidaan kysyä, onko tarvittavat muutokset ylipäättään sellaisia, että niistä saadut hyödyt ovat suuremmat kuin niihin laitettu panos.

Osa ehdotetuista muutoksista on sellaisia, että ne voivat vaatia muutoksia lakiin. Tämä analyysi jätetään kuitenkin tämän tutkimuksen ulkopuolelle, koska tarkoituksena mahdollisten käyttötapojen analyysissä on ollut tehdä vain kartoitusta ja ideointia.

5.2.1. Jatketaan kuten ennenkin

Tässä käyttötapauksessa jatketaan kuten ennenkin, eli kaikki potilastiedot ovat tallessa perusjärjestelmissä ja viitteet luodaan viitetietokantoihin. KANTA-integraation myötä potilastietojärjestelmässä luotu asiakirja siirtyy kansalliseen palveluun, mutta samalla perusjärjestelmäadapteri tallentaa viitteen viitetietokantaan. Tässä vaihtoehdossa mietitään nimenomaan sitä, että viitetietokanta säilyy peruseräajopalvelultaan muuttumattomana. Tämä käyttötapa pohjaa toimintansa viitetietokantojen ja KANTA-arkkitehtuurin erottamiselle.

Tässä vaihtoehdossa tehdään oletus, että nykyinen selainkäyttöinen asiakirjojen haku olisi parempi kuin mitä KANTA-integraatio pystyy tarjoamaan potilasjärjestelmiin. Miksi muuten käyttäjät haluaisivat käyttää KANTA:aa asiakirjojen katseluun. Toisaalta voidaan kysyä, voiko viitetietokantaan luottaa, kun siellä ei kuitenkaan olisi sellaisia potilaan asiakirjoja, jotka ovat tehty sairaanhoitopiirin ulkopuolella. Toisena pulmana on, että selaimella katseltavat asiakirjat ovat aina vain sähköisiä kopioita virallisesta asiakirjasta joka löytyy KANTA-tietokannasta. Todellisuudessa tämä ei ole ongelma, koska perusjärjestelmissä ei pitäisi olla mahdollista poistaa tai muokata tietoja. Kuitenkin tässä vaihtoehdossa säilötään kaikki tiedot kahteen kertaan pysyvästi. Nykyinen KANTA-määrittely toteaa, että asiakirjat saa säilyttää perusjärjestelmissä niin kauan kuin ne ovat KANTA-palvelussakin. Toisaalta esteitä asiakirjojen poistolle heti KANTA:an siirron jälkeen ei ole.

5.2.2. Väliaikainen tallennus

Tämä käyttötapa perustuu edellisen luvun ratkaisuun. Tässä jatkettaisiin kuten ennenkin eli säilöttäisiin keskeneräiset asiakirjat kuten aina, mutta valmiit asiakirjat poistettaisiin perusjärjestelmästä jonkun ajanjakson jälkeen (esim. 3 kk). Tähän tapaan pätee samat periaatteet kuin edelliseenkin, lisäksi kannattaa huomioida että vaikka KANTA:an ei talleteta keskeneräisiä asiakirjoja, ei niillä juurikaan ole merkitystä, koska asiakirjoista pitäisi tulla valmiita varsin nopeasti KANTA-jatkokehitys-dokumentissa olleen esityksen perusteella. Ainoastaan joitain palvelukokonaisuuksia talletettaessa voisi tällä olla merkitystä. Määräaikaisesta säilönnästä voisi kuitenkin olla hyötyä, koska usein palvelukokonaisuudet ovat lyhytkestoisia tai hoitoon liittyvät tarkastuskäynnit tapahtuvat pian hoidon jälkeen ja voivat tapahtua eri organisaatiossa kuin mitä tutkimus on tehty. Näin kummassakin tapauksessa tiedot olisi löydettävissä viitetietokannan avulla (Sillä oletuksella, että ne tehdään saman sairaanhoitopiirin sisällä).

5.2.3. Hakutietojen talletus

Tässä käyttötavassa asiakirjojen hakutiedot lisätään viitetietokantaan niin, että asiakirjan selailuhaku voidaan tehdä viitetiedoista ja varsinainen asiakirjahaku KANTA:sta. Tällöin säästyttäisiin hakemasta asiakirjan metatietoja KANTA-palvelusta, jonka tässä oletetaan olevan jotenkin hitaampaa (huomioitavaa että viitetietokantoja käytetään kuitenkin selaimella tai kuitenkin erillisellä ohjelmalla kuin perusjärjestelmää).

Tämä ominaisuus voitaisiin rakentaa viitetietokantaan riippumatta siitä, onko viitetietokanta sijoitettu erilleen kuten kuvassa 5 vai KANTA:n ja perusjärjestelmän väliin kuten kuvassa 4. Hakutietojen keräily voisi kuitenkin olla luonnollisempaa, jos viitetietokanta olisi sijoitettu välittäväksi. Koska asiakirjan metatiedot talletetaan joko asiakirjan luonnin yhteydessä tai viimeistään sähköistä allekirjoitusta tehtäessä, tulisi haku- eli metatiedot siirtää erikseen vielä viitetietokantoihin, mikäli ne eivät olisi asiakirjan tallettavan perusjärjestelmän ja KANTA-palvelun välissä.

Suurena ongelmana on, että hakutiedoista ei edelleenkään löytyisi tietoja, jotka on tallennettu sairaanhoitopiirin (viitetietokannan käyttöalueen) ulkopuolella. Tähän ratkaisu voisi olla jonkunlainen älykäs agentti, joka tutkisi samalla, onko muualla tälle potilaalle olemassa käyntejä. Toki kysely vaatisi käyttäjän antaman suostumuksen. Ongelmana on se, että viitetietokannassa toimivalla palvelujen antajalla tai sovelluksella ei ole minkäänlaista hoitosuhdetta näihin muilla alueilla tehtyihin tutkimuksiin, joten arkistolaki saattaisi estää tämän toiminnon.

5.2.4. Kuvantamispalvelut jätetään

Kuvantamispalvelut eli kuvien otto, katselu, lausuntojen antaminen ja jakelu ovat käytetyimpiä palveluita viitetietokantajärjestelmissä [Harno, 2007]. Ne ovat juuri niitä palveluita, joista saadaan tällä hetkellä eniten hyötyä, kun jaettava tieto siirtyy eri

organisaatioiden välillä [STM, 2006] [Harno, 2007]. Kuvien haku ja esittäminen on niin sujuvaa, että KANTA-tietojärjestelmältä vaaditaan paljon, mikäli siihen aiotaan saada yhtä tehokas työkalu kuin mitä se on viitetietokannoissa.

Myöskään tämä ominaisuus ei ota kantaa siihen, onko viitetietokanta sijoitettava erilleen (kuva 5) vai KANTA:n ja perusjärjestelmän väliin (kuva 4).

Miten pitäisi sitten muuttaa mahdollisesti järjestelmiä? Ainakin oletusarvoisesti pitäisi asiakirjojen haussa olla kuvien haku pois käytöstä. Kuvasiirtojen vaatimat kaistanleveydet ja arkistoitavien kuvien suuri määrä ovat nostaneet esiin tarpeen juuri lähiarkistolle [STM, 2006].

5.2.5. Koodistopalvelu

Koodistopalvelu on siinä mielessä selvä palvelu, että sellainen löytyy jo kummastakin tietojärjestelmästä, niin viitetietokannoista kuin KANTA-tietojärjestelmästä. KANTA-arkkitehtuurissa koodistopalvelu on erillään muista palveluista ja sen sisällöstä huolehtii Stakes ja ylläpidosta Kela. Palvelun vaatimusmäärittelyissä [STM, 2007j] todetaan koodistojen suoraliittymän olevan tarkoitettuna vain KANTA-arkkitehtuurin sisäisille palveluille. Mikäli suoraliittymiä tarjottaisiin myös aluetietojärjestelmien käyttöön, voisivat viitetietokannat joko jaella tai tarkastaa koodistoja perusjärjestelmiin. Tällöin viitetietokantaan liittyneillä järjestelmillä olisi aina ajan tasalla olevat koodistot (olettaen, että koodistojen vaihto onnistuu perusjärjestelmän päässä). Tämän palvelun hyöty voi kylläkin olla näennäinen, jos tätä varten pitää rakentaa erikseen liittymä. Tämän toiminnon kanssa ei tarvitse ottaa kantaa viitetietokannan sijoittumisesta KANTA- perusjärjestelmäintegraatiossa.

5.2.6. Resursoinnin apuvälineet sairaanhoitopiirin sisällä

Viitetietokantojen resursointikalenteri ja lähete-palaute-ratkaisut kannattaa säilyttää ilman muuta, koska suurimmat tehokkuushyödyt saadaan juuri niistä [Harno, 2007]. Toisaalta resursoinnin sijainti erillisessä palvelussa/selaimessa ei myöskään ole ongelma, koska se ei kuitenkaan liity suoraan tietyn asiakkaan hoidosta kirjattavaan asiakirjaan. Johtuen ratkaisun erillisyydestä, myöskään tämä toiminto ei ota kantaa siihen, miten viitetietokannat sijoittuvat KANTA- ja perusjärjestelmäintegraatiossa. Kansallinen ajanvarauspalvelu on kyllä suunnitteilla, mutta luultavaa on, ettei se valmistu 2008 alussa vielä moneen vuoteen [Harno, 2007]. Sen vuoksi viitetietokannoissa olevia resursoinnin apuvälineitä ei kannata tai ylipäätään voida poistaa.

5.2.7. Alueellinen toiminnanohjaus ja data warehouse-ratkaisut

Viitetietokantajärjestelmien oheen on asennettu erinäisiä tietovarasto- eli data warehouse-ratkaisuja, joita käytetään alueellisen toiminnanohjauksen apuna. Tällaisia ratkaisuja on käytössä niin Navitas- kuin Fiale-järjestelmissä [Harno, 2007] [Nykänen

ja Karimaa, 2002a]. Käytetyin ratkaisu on radiologian RIS-ratkaisu [STM, 2007g], mutta muita ratkaisuja on olemassa. Harno [2007] pohti asiantuntijoiden konsultointi- tai keskustelufoorumien käyttökelpoisuutta aluetietojärjestelmätasolla. Nämä konsultointi- ja asiantuntijapalvelut voisivat olla toiminnanohjauksen tukena. Joka tapauksessa alueellisella tasolla toimiva toiminnanohjaus on hyödyllistä, kun esimerkiksi palveluiden resurssivaatimukset voidaan arvioida tarkemmin nykyisen tarpeen mukaan.

5.2.8. Aito integrointi noudattaen KANTA-vaatimuksia

Tämä malli lähtee oletuksesta, jossa viitetietokannat muutetaan toimimaan niin, että potilasjärjestelmä näkee sen kuin se olisi KANTA ja toisaalta KANTA näkee sen kuin se olisi potilastietojärjestelmä. Vaikka tämä tapa oli kuvattuna kokonaisarkkitehtuurin vaatimusmäärittelyssä, on se samalla hiukan hämmentävä tietojärjestelmien integroinnin kannalta. Ongelmia tulee lähinnä siitä, että mikä olisi viitetietokannan rooli ja oikeudet toimia palvelukeskeisestä arkkitehtuuria käyttävässä kokonaisuudessa?

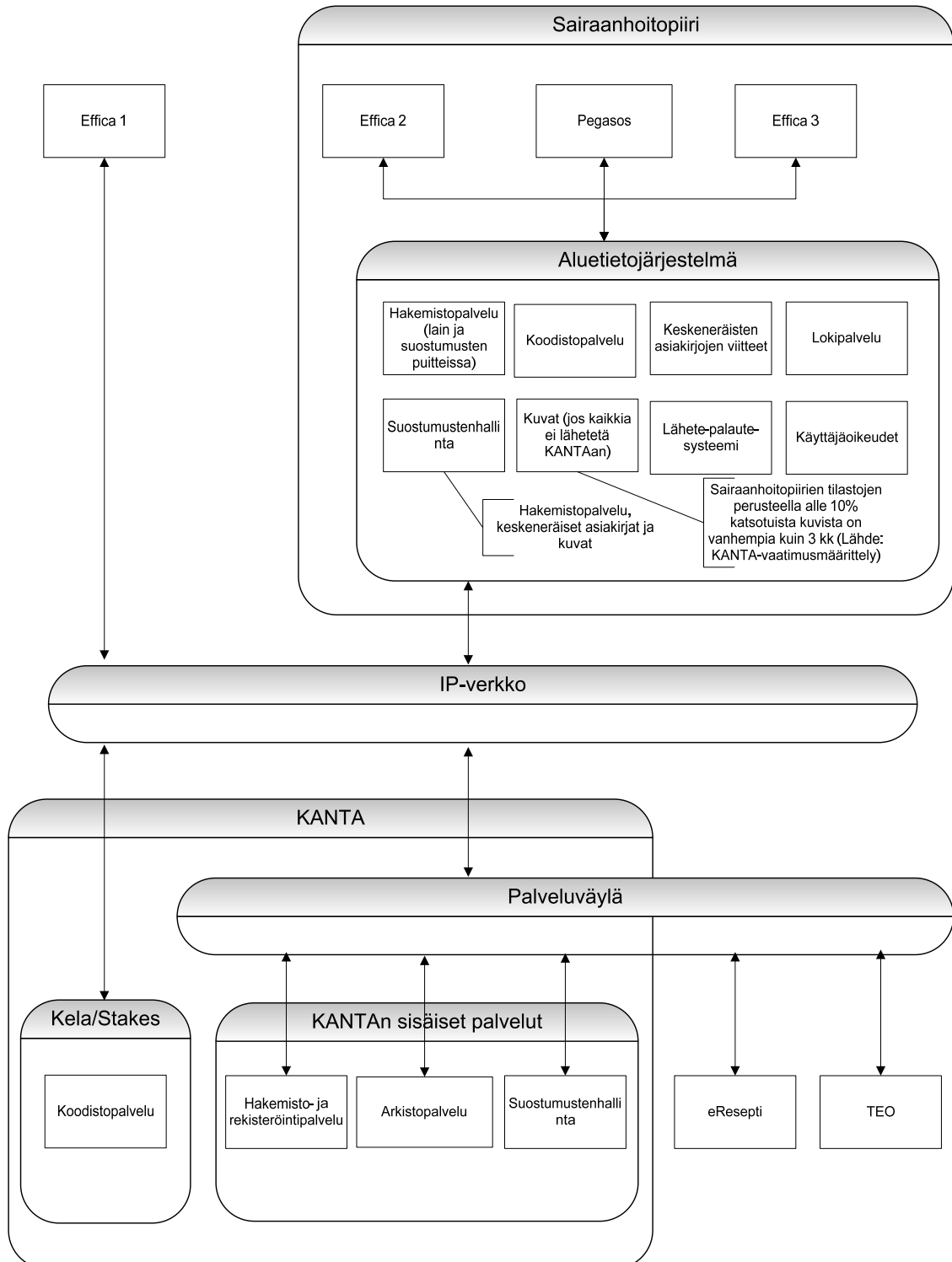
Voidaan kysyä, mitä hyötyä perusjärjestelmälle, viitetietokannalle tai KANTA-palvelulle olisi siitä, että HL7-sanomat välitettäisiin viitetietokannan kautta? Ainoa järjellinen tapa tuntuisi olevan se, että varsinaiset viestit vain reititettäisiin viitetietokannan kautta. Siis että viitetietokanta itse ei koskisi tai tulkitisi sen läpi kulkevia viestejä.

KANTA-integraation jälkeen perusjärjestelmä ottaa sanomaliikennettä avatessaan yhteyttä kansallisen arkkitehtuurin palveluväylään eli siihen palveluosoitteeseen jonka palvelunvälitin (service broker) eli koodistopalvelu on sille tarjonnut. Jotta viitetietokanta voisi jotenkin osallistua KANTA-järjestelmän ja perusjärjestelmän väliseen tiedonvälitykseen, pitäisi perusjärjestelmä jotenkin ”hämätä” (luultavasti ei onnistuisi) käyttämään viitetietokannan palvelinväylää (joka pitäisi rakentaa), jotta sillä olisi pääsy sanomiin. Kysymys kuuluu, mitä se niillä tekisi? Mitä hyötyä tällaisesta ratkaisusta olisi perusjärjestelmien kannalta? Viestit saavat kyllä kulkea viitetietokantojen läpi, mutta itse kahdenkeskiseen tietoliikenteeseen perusjärjestelmän ja KANTA:n välillä viitetietokantajärjestelmillä ei ole tarvetta tai mahdollisuutta. Mielekkäämpi vaihtoehto on tilanne, joka kuvataan seuraavassa alakohdassa

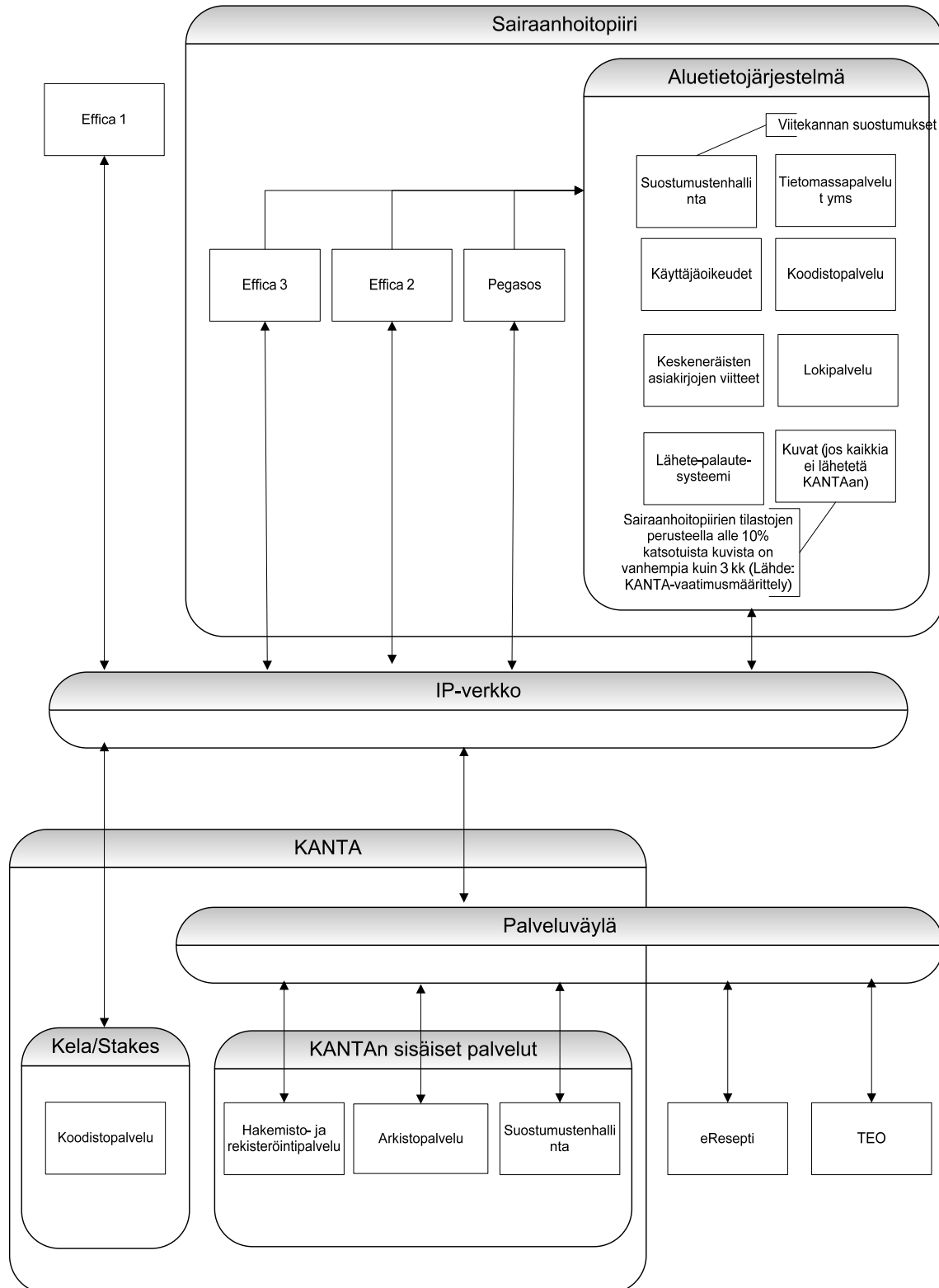
5.2.9. Viitetietokanta aktiivisena toimijana

Tämä käytötapa tarkoittaa tilannetta, joka esiteltiin jo karkealla tasolla luvussa 5.1.2. Idea on, että KANTA:n palveluväylä voisi todeta viitetietokannan pystyvän tekemään jonkun palvelun KANTA-tietojärjestelmän puolesta. Kysymykseen tulisi esimerkiksi hakutoimintojen suorittaminen viitetietokannassa KANTA:n sijaan. Avoinna tässä olisi kuitenkin se, kuinka tietyn sairaanhoitopiirin viitetietokanta ilmoittaisi palveluväylälle voivansa tehdä jonkun palvelun sellaisille käyttäjille, jotka ovat kyseenomaisella sairaanhoitopiirin alueella.

Yleisenä periaatteena tässä on, että viitetietokanta toimii aktiivisena osapuolena KANTA-tietojärjestelmän ja perusjärjestelmien kanssa tehden oma-aloitteisesti jotain sellaisia toimintoja, joita voidaan ajatella tarvittavan käyttäjien käyttäessä KANTA:a. Ajatuksena on, että viitetietokanta voisi toimia esimerkiksi välivarastona, josta noutaa tai hajauttaa tietokantahakuja ja dataa. Syytä on huomata, että KANTA-arkkitehtuurin tiukasti keskitetty rakenne rajoittaa edellä mainittuja vaihtoehtoja. Toisaalta myöskään tämän lähestymistavan hyödyistä pitkällä aikavälillä ei ole takeita.



Kuva 4: Viitetietokanta välittävänä palveluna KANTA-järjestelmän ja perusjärjestelmien välissä (Yksinkertaistettu kuva).



Kuva 5: Viitetietokanta erillisenä palveluna perusjärjestelmien rinnalla (Yksinkertaistettu kuva).

5.3. Viitetietokantojen tarpeellisuus- ja hyödyllisyyskartoitus

Tässä tutkimuksessa on analysoitu keskitetyn KANTA-arkkitehtuurin ja viitetietokantoihin perustuvien aluetietojärjestelmien yhteenliittämisen mahdollisuuksia. Vaikka tutkimuksessa löydettiin viitetietokantajärjestelmille mahdollisia rooleja, joissa ne voivat toimia myös kansallisen tietojärjestelmän käyttöönoton jälkeen, on asiaa syytä miettiä myös viitetietokantaa käyttävän ja toisaalta sitä ylläpitävän organisaation näkökulmasta. Viitetietokantaintegraatiota hoitavalla tietojärjestelmäpäälliköllä on vaikea tilanne tehdä päätöksiä siitä, mikä on tarpeen, mikä ei ja kuinka tämä kaikki vaikuttaa kustannuksiin tai peruskäyttäjien operatiiviseen tehokkuuteen. Viitetietokannan säilyttäminen (tai järjestelmän vaihto) ei saisi olla mikään itseisarvo, vaan vaa'ssa pitäisi painaa lähinnä vain edellä mainitut tarpeet, kustannukset ja tehokkuusnäkökulma. Olkoonkin, että esimerkiksi hyväksi todetun tietojärjestelmän vaihto voi tuoda tullessaan käyttöönotto-ongelmia, muutosvastarintaa, ylimääräistä (mutta pakollista) koulutusta ja muita ongelmia. Mietittäessä viitetietokantojen roolia ja tarpeellisuutta voidaan tehdä muutamia hahmotelmia, joista voi mahdollisesti olla suuntaa-antavaa apua päätöksenteossa. Tässä osioissa siis pyritään tekemään sitä mitä Linthicum [2000] esitteli 12 kohtaisessa sovellusintegraation metodilistassa kohtana 11 eli määritellä sovellusintegraation hyötyä. Toki Linthicumin [2000] mallissa kyse oli integraatioprosessin läpikäyneen tietojärjestelmän analysoinnista eikä etukäteisarvailuista kuten tässä, mutta varmasti periaattellisesti kyse on samasta asiasta. Tosin tässä kartoituksessa mietitään tarpeellisuutta ja korvattavuutta etukäteen eikä jälkikäteisesti. KANTA-arkkitehtuurin ja viitetietojärjestelmien sovellusintegraatiossa on se ominaispiirre, että tavallaan useat ominaisuudet ovat rinnakkaisia näiden kahden järjestelmäkokonaisuuden välillä. Tässä kuitenkin oletetaan, että viitetietokannoilla ja KANTA-arkkitehtuurilla voisi olla jotain molemminpuoleista yhteistoiminnallisuutta, jossa viitetietokannoillakin olisi toiminnallinen rooli. Toisin sanoen niin, että joitain viitetietokantojen ominaisuuksia, jotka löytyvät myös KANTA-järjestelmästä voitaisiin edelleen käyttää viitetietokannoista niin, että tiedot siirtyisivät niistä sitten KANTA-palveluun. Tällaisten ratkaisujen esteenä voivat olla toisaalta laki (tieto pitää saada siirrettyä suoraan perusjärjestelmästä KANTA-järjestelmään), toisaalta mahdollisten erillisten liittymien rakentamisen työläys, ylläpito ja vastuukysymykset (vaikka suoraa estettä viitetietokantojen käytölle KANTA-järjestelmän sijaan ei ole, on vastaava tieto kuitenkin saatava jotenkin hoidettua KANTA:an). Erillisten liittymien mahdollista rakentamista tulee miettiä tarkkaan, koska tällöin on olemassa vaara, että KANTA-arkkitehtuurin modulaarisuus rikkoutuu. Palvelukeskeinen arkkitehtuuri kyllä toisaalta tukee vaihtoehtoisia ratkaisuja eikä KANTA-määrityksien rajapinta-arkkitehtuuri ota kantaa siihen, mistä tai miten jokin tieto KANTA:aan tulee, kunhan sen sisältö on oikeaa ja noudattaa viestinvälityksen ja KANTA-arkkitehtuurin sääntöjä.

Kuvan 5 vuokaaviossa on esitetty karkealla tasolla viitetietokannan tarpeellisuus- ja hyötykartoituksen prosessi. Kuvatun prosessin avulla voidaan arvioida toimintokohtaisesti viitetietokantaa ja arvioida toimintoa ja sen välttämättömyyttä operatiiviselle toiminnalle, sen korvattavuutta, siitä aiheutuvia hyötyjä (materiaaliset ja immateriaaliset) ja toisaalta haittoja (lähinnä kuluina). Kaavio voidaan ajatella pragmaattisena ellei jopa tylynä työkaluna. Oletusarvoisesti kaaviossa lähdetään siitä, ettei kahden päällekkäisen järjestelmän ylläpito ole järkevää. Viitetietokannasta täytyy saada selviä hyötyjä tai siinä täytyy olla ominaisuuksia, joita KANTA-arkkitehtuurissa ei ole ja joita ei voi korvata helposti. KANTA-arkkitehtuurin käyttö on kuitenkin laissa määrätty, joten siitä ei voida oletusarvoisesti luopua.

Vuokaavion tarpeellisuus- ja hyötykartoitus tehdään iteroiden jokaiselle toiminnolle. Lopuksi tehdään arvio siitä, oliko mahdolliset kulut (negatiiviset pisteet) suurempia kuin saadut hyödyt (positiiviset pisteet). Kannattaa huomioida erikoistapaus, jossa jotain toimintoa ei löydy KANTA-arkkitehtuurista, se todetaan välttämättömäksi eikä sitä voi korvata toisella tietojärjestelmällä tai sovelluksella *järkevään hintaan*. Tällöin todetaan, että viitetietokanta täytyy ehdottomasti säilyttää, eikä tarpeellisuuskartoitusta tarvitse jatkaa.

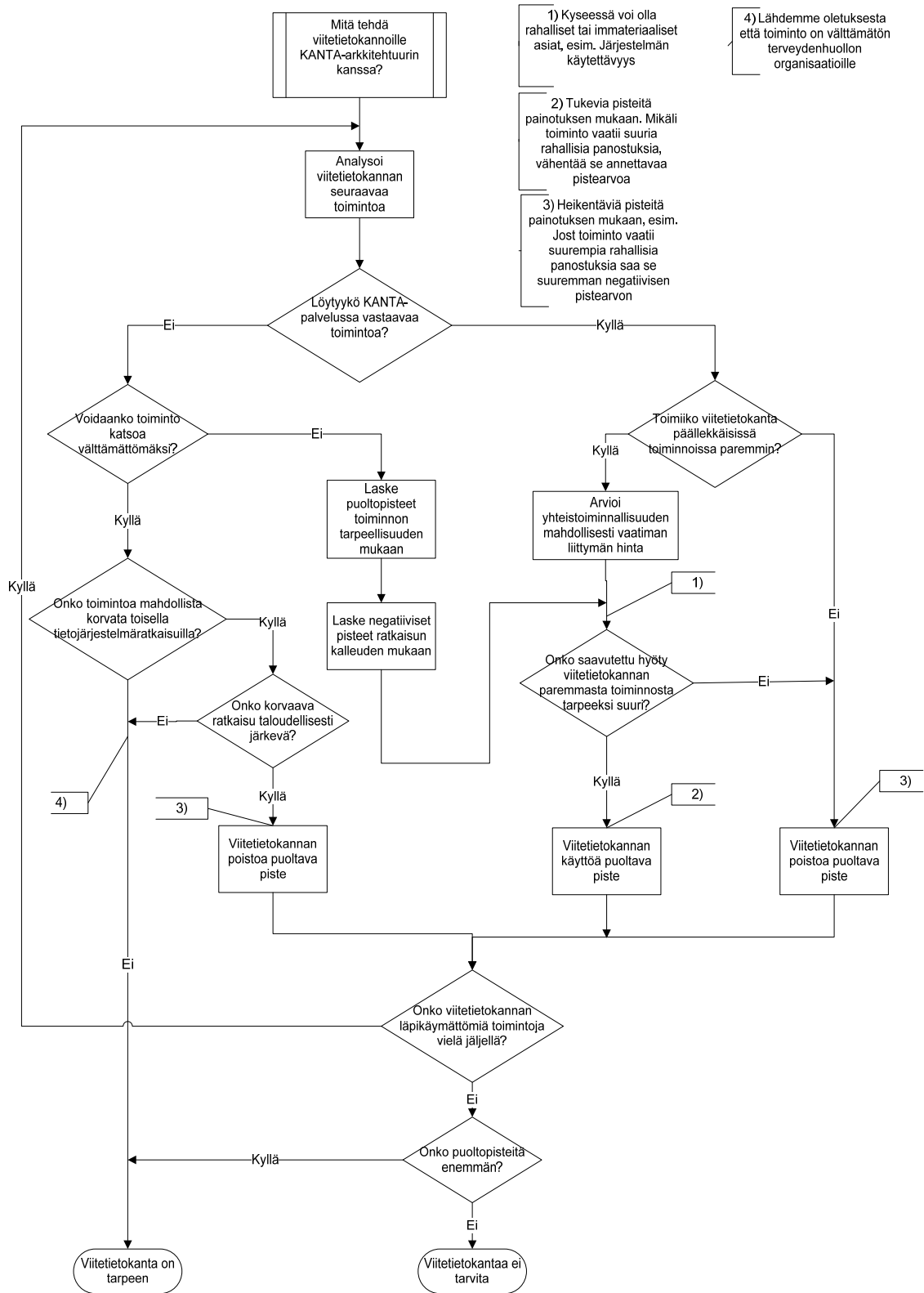
Seuraavassa käydään esimerkinomaisesti muutamia toimintojen hyötykartoituksia läpi. Kaaviossa käydään viitetietokannan toiminnot kohta kohdalta läpi arvioiden ensin sitä, löytyykö KANTA-arkkitehtuurista vastaavaa toimintoa. Jos vastaavaa toimintoa ei löydy, voidaan edellisessä kappaleessa kuvattujen ratkaisujen lisäksi myös vaihtoehtoisesti arvioida, tarvitaanko toimintoa ylipäätään ja laskea sen ylläpidolle mahdollisesti hintaa ja hyötyä.

Jos KANTA-järjestelmästä löytyy viitetietokannan toiminnolle vastine, tulee arvioida, kumpi näistä ratkaisuista on operatiivisesti parempi. Mikäli KANTA on hyödyllisempi/parempi, ei viitetietokannan tätä toimintoa tarvita ollenkaan. Mikäli taas viitetietokanta on hyödyllisempi/parempi, pitää arvioida sitä, onko se niin paljon parempi, että se oikeuttaa kahden, päällekkäisen järjestelmän ylläpidon tälle toiminnolle. Mutta kuten jo aikaisemmin tässä luvussa todettiin, eteen saattaa tulla tilanne, jossa laki kuitenkin määrää, että tietyt tiedot on lähetettävä KANTA:an. Näin ollen voi olla, että mikäli viitetietokantoja halutaan edelleen käyttää johonkin operaatioon, tulee niistä rakentaa jonkunlainen liittymä KANTA-järjestelmään tiedon siirtoa ja vaihtoa varten.

Esitelty tarpeellisuus- ja hyötykartoitus on tehty vain esimerkiksi viitetietokantojen tulevaisuuden roolin mietintään. Siinä on selviä puutteita monissa asioissa, mutta voidaan todeta, ettei se ole koskaan tarkoitettukaan sisällöltään kattavaksi. Ongelmina mallissa on muun muassa se, että kaaviossa käydään läpi hyötyjä toimintokohtaisesti, mutta samalla pisteet lasketaan kuitenkin koko järjestelmälle. Ongelma ei ole suuri, koska jos toiminto todetaan tarpeettomaksi, voidaan ajatella että sen vaatimat resurssit

(esimerkiksi palvelinkapasiteetti) ovat turhaa ylimenevää kapasiteettia. Mahdollisten lisenssimaksujen ja muiden juoksevien kulujen maksaminen tällaiselle ei ole järkevää. Toinen ongelma kaaviossa on, että siinä pyritään asettamaan samaan mittaristoon vaikeasti pääteltävät ja toisistaan riippumattomat tekijät kuten hyöty ja kulut. Kulujen laskeminen on helppoa, mutta kuinka pitäisi laskea jonkun tietyn toiminnon hyöty? Kuinka pitkälle hyödyn laskennan ulottaa?

Tarpeellisuus- ja hyötykartoituksen mallin esiteltiin edellä lyhyesti. Malli voi mahdollistaa sellaisten seikkojen huomioimisen, mitkä muuten olisivat jääneet tunnistamatta. Jatkotutkimuksessa asiaan voisi tutustua paremmin, tai perustaa tämän analyysin kirjallisuudesta löytyviin tutkimuksiin. Integraatioprosessin onnistumista ovat arvioineet omissa tutkimuksissaan muun muassa Vasconcelous *et al.* [2007] ja Linthicum [2003]. Kuitenkin sovellusintegraation tarpeet ja mahdollisuudet ovat aina tapauskohtaisia, eikä asioita voi koskaan oppia täydellisesti pelkästään kirjoista tai ennakkoon tunnistaa kaikkia ongelmatilanteita.



Kuva 6: Vuokaavio viitetietokannan tarpeellisuus- ja hyötykartoituksen prosessista.

6. Yhteenveto ja pohdintaa

Tässä tutkimuksessa tutustuttiin terveydenhuollon sähköisten tietojärjestelmien yhteistoiminnallisuuden tilaan Suomessa kansallisen sähköisen potilastietojärjestelmäarkiston toteutuksen kynnyksellä. Tutkimuksessa arvioitiin nykyistä tilannetta ja käytiin läpi sitä, miten sovellusintegraatioprosessia voitaisiin suorittaa kahden erilaista tietojärjestelmäarkkitehtuuria noudattavan osapuolen välillä. Tutkimuksessa käytiin läpi, mitä sovellusintegraatio tarkoittaa kahden erilaisen tietojärjestelmän välillä niin teoreettisista lähtökohdista kuin käytännönläheisesti ajatellen. Lisäksi käsiteltiin erilaisia malleja tietojärjestelmäarkkitehtuureille ja niiden sovellusintegraatitavoitteille. Tutkimuksessa tutustuttiin palvelukeskeiseen arkkitehtuuriin, malleihin hajautetuista ja keskitetyistä tietojärjestelmistä. Lisäksi analysoitiin sovellusintegraatiotutkimuksen näkökulmasta, mitä näiden kahden erilaisen tietojärjestelmäparadigman integrointi voisi tarkoittaa.

Kirjallisuuskartoituksen jälkeen tutustuttiin tutkimuksen varsinaiseen kohdealueeseen eli KANTA-arkkitehtuuriin ja toisaalta alueellisiin viitetietokantoihin. Aikaisempia tutkimuksia käytettiin analysoidessa sitä, minkälaisia tapaustutkimuksessa olleet tietojärjestelmät ovat. Huomattiin, että aiemmissa tutkimuksissa esitetyt jyrkät luokittelut eivät olleet sovellettavissa täysin KANTA-arkkitehtuuriin tapauksessa.

Tutkimuksen tuloksissa pyrittiin yhdistämään aiempien tutkimusten tiedot ja tapaustutkimuksen kohdealueen tietämys. Viitetietokantojen ja KANTA-arkkitehtuuriin yhdistämistä mietittiin niin yleisen sovellusintegraation kuin palvelukeskeisen arkkitehtuuriin lähtökohdista. Huomattiin, että sovellusintegraatioon on useita näkökohtia riippuen siitä, kuinka syväksi sovellusintegraation ja yhteistoiminnallisuuden halutaan näiden kahden järjestelmän välille saattaa. Tuloksissa tultiin siihen johtopäätökseen, että palvelukeskeinen arkkitehtuuri voi olla ratkaisu moniin ongelmiin, mikäli sen mahdollisuuksia halutaan hyödyntää sovellusintegraatiossa. Tutkimuksessa mietittiin eri näkökulmien pohjalta mahdollisia konkreettisia toimintoja, jotka voisivat toimia yhteen KANTA-arkkitehtuuriin ja viitetietokantojen kanssa. Tuloksissa tultiin siihen johtopäätökseen, että enemmän kuin teknisistä mahdollisuuksista, kyse on tässä sovellusintegraatioprosessissa siitä, mitä halutaan ja lain mukaan sallitaan tehtävän.

Kansallisen sähköisen arkiston saapuessa tulee viitetietokantojen rooli muuttumaan. Miten ne muuttuvat, on vielä selvittämättä. Tässä tutkimuksessa pyritään tuomaan esiin mahdollisia toimintoja, joita viitetietokannoille voisi antaa.

Luultavasti KANTA-järjestelmän arkistosta ei haluta tehdä kovinkaan useita liittymäratkaisuja erilaisille siihen liittyville tietojärjestelmille. Viitetietokantojen rooli voi olla siten vähäpätisempi kuin mitä se mahdollisesti voisi olla, ainakin ajateltaessa palvelukeskeisen arkkitehtuuriin mahdollistamia moninaisia vaihtoehtoja.

Todennäköisesti KANTA-tietojärjestelmään ei tule viitetietokantojen toimintoja vielä moneen vuoteen tietojärjestelmän käyttöönotosta. Sen vuoksi alueellisille viitetietokannoille on olemassa tilaus vielä KANTA-arkiston käyttöönoton jälkeenkin. Monia asioita voidaan ajatella hoidettavan helpommin sairaanhoitopiiritasolla ja niiden omilla järjestelmillä kuin kansallisiin ratkaisuihin tukeutuen.

Viiteluettelo

- [Abels et al, 2005] Sven Abels, Liane Haak, Axel Hahn. Identification of common methods used for ontology integration tasks. In. *Proceedings of the first international workshop on Interoperability of heterogeneous information systems, Interoperability Of Heterogeneous Information Systems*. (2005), 75 – 78.
- [Aier and Schönherr, 2006] Stephan Aier, Marten Schönherr, Evaluating Integration Architectures – A Scenario-Based Evaluation of Integration Technologies. In. *Trends in Enterprise Application Architecture* **3888** (2006), Springer, 2-14.
- [Avison and Young, 2007] David Avison, Terry Young, Time to rethink health care and ICT. *Comm. ACM*, **50**, 6 (June 2007).
- [Beyer et al., 2004] Mario Beyer, Klaus A. Kuhn, Christian Meiler, Stefan Jablonski, Friedrich Alexander, Richard Lenz, Towards a flexible, process-oriented IT architecture for an integrated healthcare network. In. *Proc. of the 2004 ACM symposium on Applied computing*, 2064- 271.
- [Bhattacharjee et al., 2007] Anol Bhattacharjee, Neset Hikmet, Nir Menachemi, Varol O. Kayhan, Robert G. Brooks. The Differential Performance Effects of Healthcare Information Technology Adoption. *Information Systems Management*, **24**, 1, (January 2007) 5 – 14.
- [Brown, 2002] Wayne Brown, Centralizing information technology in a distributed system (again?), In. *Proceedings of the 30th annual ACM SIGUCCS conference on User services*. (2002) 222 – 225.
- [Cavazos and Jarquín, 2004] Francisco Álvarez Cavazos, Juan Carlos Lavariega Jarquín,
A 3-tiered client-server distributed database system component-based. In. *Proceedings of the winter international symposium on Information and communication technologies, ACM International Conference Proceeding Series*, **58**, 2004, 1 – 6.
- [Commission of the European Communities, 2007] Commission of the European Communities, Draft Recommendation of the Commission on eHealth interoperability, 16.07.2007, saatavilla http://ec.europa.eu/information_society/newsroom/cf/document.cfm?action=display&doc_id=369 , 27.04.2008.
- [Davis and Gamble, 2004] L. Davis, R. Gamble, Understanding Services for Integration Management. *Lecture Notes in Computer Science*, **2959**, Springer. 2004, 84 – 93.
- [DICOM, 2007] Digital Imaging and Communication in Medicine (DICOM), Strategic Document, nov. 2007. <http://medical.nema.org/dicom/geninfo/Strategy.pdf>, 4.2.2008.

- [Eichelberg *et al.*, 2005] Marco Eichelberg, Thomas Aden, Jörg Riesmeier, Asuman Dogac, Gokce B. Laleci, A survey and analysis of Electronic Healthcare Record standards. *ACM Computing Surveys (CSUR)*, **37**, 4, (December 2005) 277 – 315.
- [Elmasri and Navathe, 2000] Ramez Elmasri, Shamkant B. Navathe, *Fundamentals of Database Systems. Third edition*. Addison-Wesley, 2000.
- [Ensio ja Ruotsalainen, 2003] Ensio, Antero ja Ruotsalainen, Pekka, Sähköisen asiakas- ja potilasasiakirjojen säilytyksen ja kiistämättömyyden hyvä käytäntö. Stakes, 2003, saatavilla http://sty.stakes.fi/NR/rdonlyres/00BDBB83-AEE0-467B-949C-EB73DFE12704/1313/osve_2_2003.pdf, 4.2.2008.
- [Ferranti *et al.*, 2006] Jeffrey M. Ferranti, R. Clayton Musser, Kensaku Kawamoto and W. Ed Hammond, The Clinical Document Architecture and the Continuity of Care Record: A Critical Analysis. *J. Am. Med. Inform. Assoc.*, **13**, 2, (March 2007) 235 - 238.
- [Grimson *et al.*, 2000] Jane Grimson, William Grimson, Wilhelm Hasselbring, The SI Challenge in Health Care. *Comm. of the ACM*, **43**, 6, (June 2000) 48 – 55.
- [Haikala, 2006] Haikala, Ilkka, *Ohjelmistotuotanto*. Talentum Media Oy, 2006.
- [Hallitus, 2006] Hallituksen esitys Eduskunnalle sosiaali- ja terveydenhuollon asiakastietojen sähköistä käsittelyä koskevaksi lainsäädännöksi. <http://www.finlex.fi/fi/esitykset/he/2006/20060253.pdf>, 4.2.3008.
- [Harno *et al.*, 2008] Kari Harno, Pekka Ruotsalainen, Pirkko Nykänen, Kyösti Kopra. Migration from Regional to a National eHealth Network. In: *Second International Conference on the Digital Society (ICDS 2008)*, 107 – 110.
- [HL7 Finland, 2007a] HL7 Finland, Open CDA2007 Johdanto, 18.1.2008. <http://virtual.vtt.fi/virtual/hl7/cda/06-open-cda2007/opencda2007-johdanto-v3.0.doc>, 4.2.2008.
- [HL7 Finland, 2007b] HL7 Finland, Open CDA2007 HL7 Finlandin suositus V3-viestien käyttöönottoon, Alustava versio, 12.9.2007. <http://virtual.vtt.fi/virtual/hl7/cda/open-cda2006-tyodokumentit/hl7v3-messaging-implemointi-1.x2.zip>, 4.2.2008.
- [Hristidis *et al.*, 2006] Vagelis Hristidis, Peter J. Clarke, Nagarajan Prabakar, Yi Deng, Jeffrey A. White, Redmond P. Burke. A flexible approach for electronic medical records exchange. In: *Proc. of the international workshop on Healthcare information and knowledge management, Healthcare data integration and exchange*, 33 – 40.
- [Itälä, 2000] T. Itälä, Perusjärjestelmaintegraatio ja aluetietojärjestelmä. Toiminnallinen kuvaus ja vaatimukset toteutukselle, Versio 1.0, 3.11.2000.
- [Järvinen ja Järvinen, 2004] Pertti Järvinen ja Annikki Järvinen, Tutkimustyön metodeista. Opinpajan kirja, 2004.

- [Kelan tiedote, 2007] Kansaneläkelaitoksen tiedote. <http://www.kela.fi/in/internet/suomi.nsf/NET/300507115115CS?openDocument> , 4.2.2008.
- [L159, 2007] Laki sosiaali- ja terveydenhuollon asiakastietojen sähköisestä käsittelystä, (159/2007). <http://www.finlex.fi/fi/laki/ajantasa/2007/20070159> , 4.2.2008.
- [L61, 2007] Laki sähköisestä lääkemääräyksestä, (61/2007). <http://www.finlex.fi/fi/laki/ajantasa/2007/20070061> , 4.2.2008.
- [L831, 1994] Arkistolaki, (831/1994). <http://www.finlex.fi/fi/laki/ajantasa/1994/19940831> , 4.2.2008.
- [Linthicum, 2000] David S. Linthicum. *Enterprise Application Integration*. Addison-Wesley, 2000.
- [Linthicum, 2003] David S. Linthicum. *Next Generation Application Integration: From Simple Information to Web Services*. Addison-Wesley, 2003.
- [Matsunaga *et al.*, 2007] Andréa Matsunaga, Maurício Tsugawa, José A. B. Fortes, Integration of text-based applications into service-oriented architectures for transnational digital government. In. *Proceedings of the 8th annual international conference on Digital government research: bridging disciplines & domain*, ACM International Conference Proceeding Series; **228** (2007), 112 – 121.
- [Mendoza *et al*, 2006] L. E. Mendoza, M. Perez, A. Griman. Critical Success Factors for Managing Systems Integration. *Information Systems Management*, **23**, 2, (2006) 56-75.
- [Nykänen ja Karimaa, 2002a] Pirkko Nykänen, Erkki Karimaa, Osa 1. Teoksessa *Satakunnan Makropilotin tietoteknologian arviointi. Tietoteknologian arviointihankkeen loppuraportti*. Pirkko Nykänen (toim.), Syyskuu 2002. University of Tampere, Dept. of Computer Science, Report **B-2002-3**, 2002. <http://www.cs.uta.fi/reports/bsarja/B-2002-3.pdf> , 12.04.2008.
- [Nykänen, 2007] Pirkko Nykänen, Fiale-aluejärjestelmän vaikuttavuusarviointi Pirkanmaan sairaanhoitopiirissä. University of Tampere, Dept. of Computer Science, Report **B-2007-1**, 2007. <http://www.cs.uta.fi/reports/bsarja/B-2007-1.pdf>, 12.04.2008.
- [Nykänen ja Karimaa, 2002b] Pirkko Nykänen, Erkki Karimaa, Satakunnan Makropilotin ratkaisujen mallit ja tietotekniset suunnitelmat. Teoksessa *Satakunnan Makropilotti: tulosten arviointi FinOHTAn raportti 21/2002*, Jukka Ohtonen (toim.), Stakes, 2002. <http://finohta.stakes.fi/NR/rdonlyres/E81C4727-11B1-437D-AFD6-79D8A7BDC9DF/0/r021f.pdf> , 12.04.2008.
- [OASIS Group, 2007] OASIS, Reference Model for Service Oriented Architecture. <http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf>, 4.2.2008.

- [Ohtonen, 2002] Jukka Ohtonen, Taustaa Satakunnan Makropilotti-hankkeelle. Teoksessa *Satakunnan Makropilotti: tulosten arviointi FinOHTAn raportti 21/2002*, Jukka Ohtonen (toim.), Stakes, 2002. <http://finohta.stakes.fi/NR/ronlyres/E81C4727-11B1-437D-AFD6-79D8A7BDC9DF/0/r021f.pdf>, 12.04.2008.
- [Papazoglou and van den Heuvel, 2007] Mike P. Papazoglou, Willem-Jan van den Heuvel, Service oriented architectures: approaches, technologies and research issues, *The VLDB Journal — The International Journal on Very Large Data Bases*, **16**, 3, (July 2007) 389 – 415.
- [Park and Ram, 2004] Jinsoo Park, Sudha Ram, Information systems interoperability: What lies beneath?. *ACM Trans. on Information Systems*, **22**, 4, (October 2004) 595 – 632.
- [Sheth and Larson, 1990] Amit P. Sheth, James A Larson, Federated Database Systems for Managing Distributed, Heterogenous, and Autonomous Databases. *ACM Computing Surveys*, **22**, 3, 183-236.
- [Siltanen, 2004] Juha Siltanen, Tietoarkkitehtuuriin perustuva sovellusintegraatiometodi – tapaus Tieliikelaitos, Pro Gradu-tutkielma, Tampereen yliopisto.
- [Smith, 2007] Elizabeth Smith, Fiona M. Ross, Service user involvement and integrated care pathways. *Int J Health Care Qual Assur Inc Leadersh Health Serv*, **20**, 3, (2007) 2-3.
- [Sousa *et al.*, 2006] Paulo Sousa, Matthew Bazeley, Sören Johansson, Helle Wijk, The use of national registries data in three European countries in order to improve health care quality. *Int. J of Health Care Qualy Assur*, **19**, 7, (2006) 551-560(10).
- [STM, 2006] Sosiaali- ja terveystieteiden ministeriö, Terveystieteiden valtakunnallisen tietojärjestelmäarkkitehtuurin periaatteet. Alueellisista ratkaisuksista kansalliseen kokonaisuuteen, Sosiaali- ja Terveystieteiden ministeriön selvityksiä 2006:8, 2006. saatavilla http://www.stm.fi/Resource.phx/publishing/store/2006/02/pr1138954125517/passt_hru.pdf, 12.4.2008.
- [STM, 2007a] Sosiaali- ja terveystieteiden ministeriö, Terveystieteiden kansallisen tietojärjestelmäarkkitehtuurin määrittelyprojekti, KANTA – kokonaisuusarkkitehtuuri, vaatimusmäärittely, 28.2.2007. saatavilla <http://www.stm.fi/Resource.phx/vastt/tietoh/maar.htx.i1625.doc>, 4.2.2008.
- [STM, 2007b] Sosiaali- ja terveystieteiden ministeriö, Terveystieteiden kansallisen tietojärjestelmäarkkitehtuurin määrittelyprojekti, KANTA – kokonaisuusarkkitehtuuri, arkkitehtuurimäärittely, 28.2.2007. saatavilla <http://www.stm.fi/Resource.phx/vastt/tietoh/maar.htx.i1626.doc>, 4.2.2008.

- [STM, 2007c] Sosiaali- ja terveysministeriö, Terveydenhuollon kansallisen tietojärjestelmäarkkitehtuurin määrittelyprojekti, KANTA – viestinvälitys, vaatimusmäärittely, 28.2.2007. saatavilla <http://www.stm.fi/Resource.phx/vastt/tietoh/maar.htx.i1636.doc> , 4.2.2008.
- [STM, 2007d] Sosiaali- ja terveysministeriö, Terveydenhuollon kansallisen tietojärjestelmäarkkitehtuurin määrittelyprojekti, KANTA – viestinvälitys, arkkitehtuurimäärittely, 28.2.2007. saatavilla <http://www.stm.fi/Resource.phx/vastt/tietoh/maar.htx.i1637.doc> , 4.2.2008.
- [STM, 2007e] Sosiaali- ja terveysministeriö, Kanta jatkomäärittely; Ydindokumentti lausuntojen perusteella muokattuna, 28.2.2007. saatavilla <http://www.stm.fi/Resource.phx/vastt/tietoh/jatkomaar.htx.i479.pdf> , 4.2.2008.
- [STM, 2007f] Sosiaali- ja terveysministeriö, lausuntoyhteenveto, STM/18/2006, <http://www.stm.fi/Resource.phx/vastt/tietoh/maar.htx.i1643.pdf> , 4.2.2008.
- [STM, 2007g] Sosiaali- ja terveysministeriö, Terveydenhuollon kansallisen tietojärjestelmäarkkitehtuurin määrittelyprojekti, KANTA – Arkistopalvelu, vaatimusmäärittely, 28.2.2007. saatavilla <http://www.stm.fi/Resource.phx/vastt/tietoh/maar.htx.i1640.doc> , 4.2.2008.
- [STM, 2007h] Sosiaali- ja terveysministeriö, Terveydenhuollon kansallisen tietojärjestelmäarkkitehtuurin määrittelyprojekti, KANTA – Tunnistaminen ja sähköinen allekirjoitus, vaatimusmäärittely, 28.2.2007. saatavilla <http://www.stm.fi/Resource.phx/vastt/tietoh/maar.htx.i1629.doc> , 4.2.2008.
- [STM, 2007i] Sosiaali- ja terveysministeriö, Terveydenhuollon kansallisen tietojärjestelmäarkkitehtuurin määrittelyprojekti, KANTA – Loki- ja valvontapalvelu, vaatimusmäärittely, 28.2.2007. saatavilla <http://www.stm.fi/Resource.phx/vastt/tietoh/maar.htx.i1635.doc> , 4.2.2008.
- [STM, 2007j] Sosiaali- ja terveysministeriö, Terveydenhuollon kansallisen tietojärjestelmäarkkitehtuurin määrittelyprojekti, KANTA – koodistopalvelu, vaatimusmäärittely, 28.2.2007. saatavilla <http://www.stm.fi/Resource.phx/vastt/tietoh/maar.htx.i1642.doc> , 4.2.2008.
- [STM, 2007k] Sosiaali- ja terveysministeriö, Opas Ydintietojen, otsikoiden ja näkymien toteuttaminen sähköisessä potilaskertomuksessa, Versio 2.2, 31.1.2007. saatavilla <http://www.stm.fi/Resource.phx/vastt/tietoh/ydintiedot.htx.i304.pdf> , 5.5.2008.
- [Stumptner *et al.*, 2004] Markus Stumptner, Michael Schrefl, Georg Grossmann, On the Road to Behavior-Based Integration. *Proc. of the first Asian-Pacific conference on Conceptual modelling, Conferences in Research and Practice in Information Technology Series*, **59**, (2004) 15-22.
- [Vasconcelos *et al.*, 2007] André Ferreira Ferrão Couto e Vasconcelos, Pedro Manuel Moreira Vaz Antunes de Sousa and José Manuel Nunes Salvador Tribolet,

Information System Architecture Metrics: An Enterprise Engineering Evaluation Approach. *The Electronic Journal Information Systems Evaluation*, **10**, 1, (Jun. 2007) 91-122.

[Virtanen, 2000] A. Virtanen, Avoimen arkkitehtuurin ja tietoteknisten palvelujen kehittäminen. Projektisuunnitelma, versio 0.2, 11.9.2000.

[World Wide Web Consortium (W3C), 2004] World Wide Web Consortium, Web Services Architecture. <http://www.w3.org/TR/ws-arch/>, 4.2.2008.

[Wyss and Van Gucht, 2001] Catharine Wyss, Dirk Van Gucht, A Relational Algebra for Data/Metadata Integration in a Federated Database System. *Proc. of the tenth international conference on Information and knowledge management*. 65-72.

Kansallisen keskitetyn sähköisen arkiston arkkitehtuurin määrittelyiden asettamat yleiset vaatimukset jaoteltuna

Seuraavassa on esiteltyä KANTA-arkkitehtuurin määrittelyistä poimitut vaatimukset ulkoisille, siihen liittyville terveydenhuollon tietojärjestelmille. Vaatimuksen jälkeen suluissa lukee määrittely mistä se on löydetty ja määrittelyksen kohta otsikkotasolla. Mikäli vaatimus koskee erityisesti aluetietojärjestelmää, on se merkattu erikseen laittamalla ko. kohtaan neljä tähteä (****).

Yleiset vaatimukset:

1. KANTAn ja perusjärjestelmän yhteiskäytön tulee jo alkuvaiheessa tehostaa operatiivista toimintaa (kokonaisarkkitehtuuri: luku 5.1).
2. Terveydenhuollon asiakkaille tulee taata hyvä tietosuojatila. Suostumuskäytäntöjen tulee olla kuitenkin yksinkertaisia ja selkeitä jotta niitä on helppo käyttää (KANTA-jatkokehitys luku 1.3).
3. Palvelun antajien tietoprosesseista tulee poistaa sellaisia käsittelytapoja jotka periytyvät paperiarkiston ajoilta (KANTA-jatkokehitys luku 1.7.6, ei sinällään koske perusjärjestelmää).
4. Avohoitokäynneistä (määrämuotoisista asiakirjoista) pitää tallettaa merkinnät vuorokauden kuluessa potilaan poistumisesta vastaanotolta (KANTA-jatkokehitys: luvut 2.5.2, 2.5.3).
5. Potilastietojärjestelmän tulee täyttää kelan sertifiointimääritykset (sitten kun sellaiset on olemassa) (KANTA-jatkokehitys: luku 9).
6. Potilastietojärjestelmien toimittajan tulee peilata omaa järjestelmäänsä KANTA-jatkokehityksen tavoitetilan kuvaukseen ja toteuttaa kehityssuunnitelma KANTA-yhteensopivuuden vaatimista muutoksista ja niiden vaiheistuksesta (KANTA-jatkokehitys: luku 1.7.4).
7. Mikäli potilastietojärjestelmää ei ole mahdollista liittää suoraan käyttämään koodistopalvelua metatietojen hakua ja käsittelyä varten, saa metatietojen haku tapahtua eräajotyypillisesti ja käsittelysäännöt sijaita potilastietojärjestelmässä, mikäli järjestely mahdollistaa koodistojen mahdollisten muutosten päivityksen potilastietojärjestelmään. Näin järjestelyn tulee läpäistä Kelan tuleva sertifiointi. (KANTA-jatko luku 2.6 teksti on mielestäni ristiriidassa koodistopalvelun määrittelyjen kanssa koska siinä puhutaan, ettei koodiston suorakäyttöä tekisi kuin KANTA itse: luvut 1.2, 2.1, 6.3, 6.3)

Juridiset vaatimukset:

1. Sosiaali- ja terveydenhuollon palvelujen antajan tulee pitää rekisteriä omien asiakastietojärjestelmiensä ja asiakasrekisteriensä käyttäjistä sekä näiden käyttöoikeuksista. Tämän rekisterin tulee kattaa sekä ajantasaiset tiedot että aikaisemmat käyttäjätiedot. Käyttäjärekisterin tiedoista tulisi käydä ilmi kenelle käyttöoikeudet on myönnetty, mihin potilasrekisteriin tai sen osaan oikeudet on myönnetty, käyttöoikeuden laajuus (katselu, ylläpito) sekä käyttöoikeuden alkamis- ja päättämisaikajankohta (kokonaisarkkitehtuurin vaatimismäärittely, luku 8.3, kohta KA-JR004).

Toiminnalliset vaatimukset:

1. Terveydenhuollon ammattihenkilöstö käyttää KANTA-palveluja potilastietojärjestelmän kautta ilman suoraa yhteyttä KANTA-järjestelmään (kokonaisarkkitehtuuri: luvut 2.7, 6.3,).
2. Potilastietojärjestelmän tulee olla integroitu KANTA-viestinvälityksen rajapintaan, jonka avulla se osaa arkistoida asiakirjoja (kokonaisarkkitehtuuri: luku 9.6), noutaa arkistosta asiakirjoja (käyttö- ja luovutuspyynnöt) (kokonaisarkkitehtuuri: luku 9.8), näyttää käyttäjälle KANTAsta noudettujen asiakirjojen sisältämät tiedot (kokonaisarkkitehtuuri: luku 9.9) ja käsitellä näiden tapausten mahdolliset virhetilanteet.
3. Aluejärjestelmä voi olla liittyneenä KANTAan, mikäli se pystyy toimimaan kansalliseen järjestelmään päin potilastietojärjestelmän kaltaisesti ja toisaalta, että se pystyy toimimaan paikallisiin järjestelmiin päin vastaavalla tavalla palvelut välittäen kuin kansallinen sähköinen potilastietoarkisto (Kokonaisarkkitehtuuri: luku 3.2.3). ****
4. Potilasasiakirjat tulee muodostaa eri toimittajien järjestelmissä saman standardin sisällön ja rakenteen mukaisesti käyttäen OID-tunnistusta asiakirjojen yksilöinnissä (kokonaisarkkitehtuuri: luvut 5.3.1, 9.4). Erityisesti ydintiedot pitäisi olla yhdenmukaisesti määriteltyjä: ”Ydintiedoilla tarkoitetaan terveyden- ja potilaskertomuksen yhdenmukaisesti määriteltyjä rakenteisia tietoja” (kokonaisarkkitehtuuri: luku 7.5.1) Omaa tulkintaa: Koska tämä ei ole varmastikaan mahdollista muuttaa suoraan tietojärjestemätasolla ilman suurempia muutoksia, täytyy perusjärjestelmiin ainakin rakentaa adapteri tiedon muokkaamiseen tarkoittaen että ainakin asiakirjat ja niiden palvelukokonaisuudet/-tapahtumat yksilöidään OID-tunnisteiden avulla (kokonaisarkkitehtuuri: luku 7.5.1). Huom! Sovellusintergraation taso pitää määrittää!
5. Asiakirjoja voidaan hakea potilastietojärjestelmän kautta lähettämällä hakusanoma hakemisto- ja rekisteröintipalveluun (kokonaisarkkitehtuuri: luvut 5.4.1, 9.7). Palvelun pitää ymmärtää myös mahdolliset virhetilanteet. Oma tulkintaa: Käyttäjällä pitää olla siihen suostumus (kokonaisarkkitehtuuri: luku 5.4.1).
6. Potilastietojärjestelmän tulee pystyä luomaan ja käsittelemään palvelutapahtumia ja -kokonaisuuksia, jotka on yksilöity OID-tunnuksilla. Palvelukokonaisuuksissa pitää olla myös mahdollista olla olemassa tietoja, joita potilastietojärjestelmä ei itse näe tai mahdollista lisätä palvelutapahtumia, jotka

- liittyvät palvelukokonaisuuksiin, mitä järjestelmä ei itse omista (luku 9.5, katso myös kohta 3 tässä listassa). Jokaisella merkinnällä pitää olla potilastietojärjestelmässä ainakin palvelutapahtuma (KANTA-jatkokehitys: luku 3.2) ja palvelutapahtumia tulee pystyä myös hakemaan OID-tunnuksella (KANTA-jatkokehitys: luku 3.3). HUOM! Tämä voi olla työläs toteutettava.
7. Potilastietojärjestelmän tulee osata hallita palvelukokonaisuustunnuksen välittäminen eri palvelun antajien välillä (KANTA-jatkokehitys: luku 3.3). Yksittäiset palvelutapahtumat pitää pystyä yhdistämään palvelukokonaisuuteen (mikäli sellainen on).
 8. Potilasjärjestelmän tulee kirjata käyttölokiinsa kaikki arkistohaut KANTA-palvelusta. Käyttölokeista tulee olla mahdollista selvittää potilasta koskevien tietojen käyttö ja ne tulee säilyttää säädösten mukaan (kokonaisarkkitehtuurin vaatimismäärittely).
 9. Potilasjärjestelmässä tulisi säilyttää jatkuva potilaskertomus, mutta antaa samalla mahdollisuus kytkeä saman vaivan hoitoon liittyvät tiedot toisiinsa. Tehdyt merkinnät tulee olla mahdollista tehdä rakenteisessa muodossa jotta niiden haku ja tulkinta on helpompaa (KANTA-jatkokehitys luvut 1.3, 1.6). Tehdyt merkinnät tulee olla myös mahdollista näyttää rakenteisesti (KANTA-jatkokehitys: luku 2.3).
 10. Potilastietojärjestelmän käsitteet tulee muuntaa yhteisesti käytettyyn ja ymmärrettyyn muotoon. Huom! Koska Kela tekee tämän käsitelistan? (KANTA-jatkokehitys luvut 1.6, 5.4, 1.7.2).
 11. Arkistohallinnolliset kuvailutiedot (metatiedot) tulee tuottaa potilasjärjestelmässä automaattisesti ja heti silloin kun itse merkintäkin tehdään. (KANTA-jatkokehitys: luku 2.6)
 12. Potilastietojärjestelmän tulee liittää merkintää kuvailevaksi tiedoksi ainakin seuraavat tiedot: Näkymä, hoitoprosessin vaihe ja otsikko (KANTA-jatkokehitys: luku 2.3).
 13. Potilastietojärjestelmän tulee pystyä pitämään kirjaa siitä, mitkä merkinnät on jo siirretty arkistoon osana jotain asiakirjaa ja mitkä eivät, jotta kaikki merkinnät varmasti siirtyvät KANTAan (KANTA-jatkokehitys: luku 2.5.1).
 14. Mikäli järjestelmässä on hyväksymättömiä asiakirjoja, tulee niitä varten rakentaa hälytystoiminto tai mikäli allekirjoittamiseen riittää järjestelmän allekirjoitus, tulee näistä asiakirjoista tehdä hyväksyntä automaattisesti (KANTA-jatkokehitys: luku 2.5.1).
 15. Potilastietojärjestelmässä voi olla mahdollista sähköisesti allekirjoittaa monta asiakirjaa kerralla kuitenkin niin, että kerralla voi allekirjoittaa vaan useita yhden ja saman potilaan asiakirjoja (KANTA-jatkokehitys: luku 2.7.1).
 16. Potilastietojärjestelmän tulee allekirjoittaa myös arkistohallinnon asiakirjat, jotka eivät ole varsinaisia potilasasiakirjoja (Esim, tietojen hakusanomat) (KANTA-jatkokehitys: luku 2.7.2).
 17. Potilaan käyntiin liittyvät suostumusasiakirjat tulee olla lähetettynä kansalliseen arkistoon (Miten lähetetään, pitääkö perusjärjestelmän huolehtia tästä? Miten hoidettu aluejärjestelmissä?) (KANTA-jatkokehitys: luku 5.2.5). ****
 18. Potilastietojärjestelmän tulee tarkastaa, että tiedon luovutuksen yhteydessä luovutuksen pyytjä on eri henkilö kuin potilashallinnon varmistuksen perusteena

- olevan kirjauksen tekijä (KANTA-jatkokehitys: luku 5.3.1, Suostumuksenhallinta: luku 1.3.4).
19. Mikäli Potilastietojärjestelmässä on mahdollista tehdä ostopalveluita tai olla sellaisen tarjoajana, pitää järjestelmässä olla mahdollista luoda varmistusasiakirja (Onko tällaista oikeasti määritelty?, KANTA-jatkokehitys: luku 5.3.2).
 20. Potilastietojärjestelmässä ei saa muuttaa arkistosta saatuja asiakirjoja (niiden kopioita), mutta niiden tietoja voidaan hyödyntää. Tällainen kopioitu tieto sisältää kuitenkin omat tunnisteensa ja metatietonsa (KANTA-jatkokehitys: luvut 5.4, 6.1).
 21. Potilaan hoitoon liittyviä tietoja voidaan säilyttää potilastietojärjestelmissä niin kauan kuin se potilaan hoidon kannalta on tarpeen - kuitenkin korkeintaan se sama aika, kuin kyseisiä tietoja säilytetään KANTA-palvelussa (KANTA-jatkokehitys: luku 6.1).
 22. Potilastietojärjestelmien pitää tunnistaa mikäli haettava tai käsiteltävä tieto on erityissuojattava ja antaa käyttäjälle ylimääräinen varoitus siitä, jotta käyttäjät osaa välttää tämän tiedon katselu (mikäli katseluun ei ole aiheutta) (KANTA-jatkokehitys: luku 11).
 23. Potilastietojärjestelmän tulee muodostaa luovutuspyyntöasiakirja, suostumusasiakirja sekä potilashallinnon varmenneasiakirja oikean muotoisina ja sisällöltään oikeina ja lähettää ne kansalliselle arkistopalvelulle (Suostumuksenhallinta: luku 1.3.4).
 24. Kaikista kirjallisista suostumuksista luodaan suostumusasiakirja, josta tulostetaan jäljennös potilaalle (Suostumuksenhallinta: luku 1.3.4).
 25. Potilastietojärjestelmä varmistaa, että järjestelmien käyttäjät eivät pääse ohittamaan ohjelmistoihin kuvattuja kontrolleja (Suostumuksenhallinta: luku 1.3.4).
 26. Potilastietojärjestelmä tarkistaa potilaan sähköisen allekirjoituksen, mikäli potilaalle annetaan mahdollisuus allekirjoittaa hakemistohaku ja korvata tällä allekirjoituksella potilashallinnon varmenne (Suostumuksenhallinta: luku 1.3.4).
 27. Potilastietojärjestelmä huolehtii luovutuspyynnöllä saatujen kopioiden poistosta viimeistään silloin, kun niiden luovutuksen perusteena oleva hoitajakso tai hoitokokonaisuus kyseisessä yksikössä on päättynyt (Suostumuksenhallinta: luku 1.3.4).
 28. Mikäli terveydenhuollon palvelun antaja järjestää potilaille suostumusten tai kieltojen kirjaamiseksi käyttöliittymiä, joissa voi nähdä potilasasiakirjoja, näiden käyttöliittymien tietoturvasa ja tietosuojassa tulee noudattaa samoja periaatteita kuin Kansaneläkelaitoksen järjestämässä katseluyhteydessä (Suostumuksenhallinta: luku 1.3.4).
 29. Luovutukset muulle kuin toiselle terveydenhuollon palvelun antajalle sekä oma-aloitteinen tietojen luovutus (lähete-palaute) toiselle terveydenhuollon palvelujen antajalle toteutetaan palvelun antajan järjestelmästä. Palvelunantajan potilastietojärjestelmän tulee varmistaa, että potilastietoja luovutetaan vain potilaan suostumuksella tai luovutuksen oikeuttavan lain nojalla. Potilastietojärjestelmä tallentaa perusteen luovutukseen (kirjallinen suostumusasiakirja HL7 CDA R2-asiakirjana, tieto suullisesta suostumuksesta

tai luovutuksen perustuminen lakiin), potilashallinnon varmenneasiakirjan sekä luovutuspyyntöasiakirjan kansalliseen terveydenhuollon arkistopalveluun (Suostumuksenhallinta: luku 1.3.4).

30. Potilasjärjestelmä huolehtii potilaan edustajan valtuutuksen varmistamisesta silloin kuin edustaja antaa potilaan puolesta suostumuksia tai kieltoja (Suostumuksenhallinta: luku 1.3.4).
31. Potilasjärjestelmä huolehtii viitetietojärjestelmiin talletettujen tietojen siirtämisestä arkistopalveluun ja siirron edellyttämien suostumusasiakirjojen kirjaamisesta (SH028) (Suostumuksenhallinta: luku 1.3.4). ****
32. Potilastietojärjestelmän ollessa yhteydessä KANTAan tarvitaan aina varmenne (Kokonaisarkkitehtuuri: luku 7.2). Tarkemmin suostumukset ja asiakirja kokonaisuus menee taulukon mukaan:

Toiminto/tieto	Potilasasiakirja	Suostumus	Varmenne
Rekisteröinti	Kuvailutiedot tarvitaan	Ei tarvita	Tarvitaan
Arkistointi	Tarvitaan	Ei tarvita	Tarvitaan
Haku	Haku palauttaa potilasasiakirjan hakutiedot	Tarvitaan suullinen tai kirjallinen	Tarvitaan, tieto hoitosuhteesta riittää
Käyttö	Palauttaa potilasasiakirjan	Ei tarvita	Tarvitaan, tieto hoitosuhteesta riittää
Luovutus	Palauttaa potilasasiakirjan	Tarvitaan kirjallinen	Tarvitaan, tieto hoitosuhteesta riittää
Hallinta	Ainoastaan kuvailutietoja voidaan korjata	Ei tarvita	Tarvitaan

Yleiset arkkitehtuuri- ja tekniset vaatimukset:

- Potilasjärjestelmän tulee osata tuottaa ja käsitellä CDA R2-viestejä ja muita arkiston edellyttämiä asiakirjastandardeja (kokonaisarkkitehtuuri: luku 3.2.1). Asiakirjoina on potilasasiakirjan lisäksi suostumus-, kielto- ja luovutusasiakirjat (kokonaisarkkitehtuuri: luku 9.3).
- Tietojen yksilöintiin tulee käyttää OID-tunnistusta (kokonaisarkkitehtuuri: luvut 5.3.1, 9.4).
- Asiakirja tekstiin ei saa tehdä tietojärjestelmäkohtaisia tehostemerkintöjä tai

muotoilukoodeja. Ainoastaan W3C:n sallimat korostusmerkinnät (*italic*, *emphasis*, *ym*) sallitaan (KANTA-jatkokehitys: luku 2.3).

4. Liitännäisjärjestelmät keskustelevat viestinvälitysrajapinnan kanssa omilla Web Services (SOAP-Client) komponenteilla (Viestinvälitys: luvut 4.1, 5.1).
5. Viestien vastaanottoa KANTA-palvelulta (kuittaukset) varten liitännäisjärjestelmiin täytyy rakentaa oma SOAP Server-komponentti (Viestinvälitys: luvut 4.1, 5.1).
6. Potilastietojärjestelmien tulee käyttää X.509-varmenteita sanomaliikenteen osapuolten tunnistamisessa (Viestinvälitys: luku 4.7.3).
7. Perusjärjestelmien tulee ladata koodistot ja toimipaikkatiedot kertaluonteisesti omaa suorakäyttöä varten (Koodistopalvelu: luvut 2.1, 6.2). Suoraa yhteyttä koodistopalveluun niillä ei tule olemaan (Koodistopalvelu: luku 6.3).
8. Perusjärjestelmien ja kansallisen palvelun välissä voi olla paikallinen/alueellinen koodistopalvelu (Aluejärjestelmä?, Koodistopalvelu: luku 2.1). ****
9. Tulevaisuudessa voi olla mahdollista rakentaa perusjärjestelmiin liittymä, jolla otetaan yhteys koodistopalveluun ja päivitetään tehty muunnos sinne (Koodistopalvelu: luku 4.2.4).

Tietoturva-vaatimukset:

1. Käyttäjien tunnistaminen ja käyttöoikeuksien tarkistus tulee tapahtua potilastietojärjestelmässä riittävällä (Lain määräämällä) tietoturvasolla (TEOn ammattivarmenteet) (kokonaisarkkitehtuuri: luvut 3.2.1, 6.3, kerrottu tarkemmin kohdassa 9.2).
2. Potilastietojärjestelmän tietoliikenneyhteys arkistopalveluun tulee olla riittävän suojattu (luottosuhde) (kokonaisarkkitehtuuri: luku 3.2.1).
3. Potilastietojärjestelmällä on voitava suorittaa ohjelmallinen tai terveydenhuollon ammattihenkilön toimikorttiin perustuva sähköinen asiakirjan allekirjoitus TEOn ohjeistamalla menettelyllä. Ensivaiheessa ei tarvita tukea potilaan sähköiselle allekirjoitukselle (kokonaisarkkitehtuuri: luku 9.11).