

Päätöksenteon tuki

Rajapintamäärittäminen

SerAPI-projekti	
Yhteyshenkilö	Marko Suhonen (Marko.Suhonen@uku.fi)
Dokumentin versio	1.0
Päiväys	30.8.2007

Sisällysluettelo

1	Johdanto ja määrittelyn tavoitteet	6
2	Käsitteet ja tavoitteet	7
2.1	Määrittelyn tausta, käsitteet ja toteutuslähtökohdat	7
2.1.1	Päätöksentuen ei-toiminnalliset vaatimukset	8
2.2	Toteutuslähtökohtien arviointi	8
2.2.1	Suurin mahdollinen joustavuus/avoimuus.....	9
2.2.2	Helppo liitettävyys potilastietojärjestelmien kannalta	10
2.2.3	Helppo siirtyminen nykypilotista.....	11
2.2.4	Tehokkuuden maksimointi käyttäjän kannalta	11
2.2.5	Päätöksentuen käynnistystilanteet	12
2.3	Taustaa: CDA-lomakkeiden käyttö päätöksentuen rajapinnassa	13
2.3.1	HL7 Finland: Päätöksentuen CDA R2	13
2.3.2	Potilaskertomuksen CDA-dokumenttien käyttö	14
2.3.3	CDA:n käyttämättä jättäminen	15
2.4	Taustaa: Kansallisen arkiston näkökulmat.....	15
3	Rajapinnan ja tietosisällön taustaa	17
3.1	Kutsun tuottamisen pohjana käytetty objektimalli.....	17
3.2	Päätöksentuen palaute.....	19
3.3	E-Utilities: tietämyksen haku ja katselu linkkien kautta	20
3.4	Esimerkki tietämyksen suorittamisesta: päätöksentukiskriptit	21
3.4.1	Käypä hoito -suosituksiin ja EBMG-artikkeleihin liittyvät palautteessa hyödynnettävät tiedot	22
4	Rajapinnan käytön edellytyksiä ja vaatimuksia.....	23
4.1	Oikeuksiin, tietoihin ja toimintatapoihin liittyvät vaatimukset.....	23
4.2	Tekniikkaan liittyvät vaatimukset	23
5	Rajapintaratkaisun kuvaus	25
5.1	Sovellusosapuolet ja sovellusarkkitehtuuri.....	25
5.2	Tekninen arkkitehtuuri.....	25
6	Toiminnallisuus.....	27
6.1	Roolit ja vastuut.....	27
6.1.1	Kutsujan vastuut	27
6.1.2	Palveluntarjoajan vastuut	28
6.2	Palvelun käyttötavat	28
6.2.1	Kertakutsu.....	28
6.2.2	Uusien ja muuttuneiden tietojen toimitus.....	29
6.3	Päätöksentukityypit	29
7	Rajapinta ja sen operaatiot.....	31
7.1	Vakiotietotyypit ja attribuutit	31
7.2	Kutsu (DSSRequest).....	32
7.2.1	Characteristics.....	33
7.2.2	MedicationList	34
7.2.3	ProblemList	35
7.2.4	RiskList	37
7.2.5	HealthFactors.....	40
7.2.6	PreventionList.....	41
7.2.7	TestList.....	42

7.2.8	PhysioMeasureList.....	43
7.2.9	ProcedureList.....	43
7.2.10	Jatkohoito - TreatmentPlan.....	44
7.2.11	Kutsun laajennus.....	44
7.3	Vastaus (DSSResponse)	45
7.4	Virhetilanteet.....	46
7.5	Esimerkki SOAP-viestistä	47
8	Toteutuskohtaiset asiat	49
8.1	Rajapinnan toteutuksessa huomioitavat asiat.....	49
8.2	Määrityksen mukaisuus	49
9	Määrityksen jatkokehitys ja toteutukset	51

Liite: WSDL-kuvaus

Versiohistoria

Versio:	Pvm:	Laatijat:	Selitys:
Versio 1.0	30.8.2007	Marko Suhonen Juha Mykkänen Esa Paakkanen Heli Luostarinen Assi Pöyhölä	Julkistettu versio

Esipuhe

Tämä työ liittyy SerAPI-hankkeeseen (Palveluarkkitehtuuri ja Web-sovelluspalvelut Terveydenhuollon Ohjelmistotuotannossa ja -integraatiossa), jossa tutkitaan ja kehitetään web-sovelluspalvelujen ja palvelupohjaisen arkkitehtuurin hyödyntämistä terveydenhuollon tietojärjestelmätarpeisiin ja sovellusintegraatioon ja uusiin sekä olemassa oleviin ohjelmistotuotteisiin. Hanketta rahoittavat Tekes (päätos nro 40437/04) sekä joukko yrityksiä ja sairaanhoitopiirejä. Tämä selvitys on tehty tukemaan tarpeita päätöksentuen integroimiseksi potilastietojärjestelmiin, yhteistyössä EBMeDS-päätöksentukihankkeen ja eHealth Partners Finland-hankkeen kanssa. Erityiskiitos kuuluu EBMeDS-hankkeen ja Duodecimin puitteissa kohteeseen osallistuneille Peter Nybergille, Ilkka Kunnamolle ja Jorma Komulaiselle. Kiitokset myös Prowellness Oy:n Mika Sipilälle ja Savonia AMK:n Maritta Korhoselle. Suurkiitos kohteeseen osallistuneille SerAPI-projektin työntekijöille.

1 Johdanto ja määrittelyn tavoitteet

Tämä dokumentti on avoin rajapintamäärittely päätöksentuen liittämiseksi potilastietojärjestelmiin.

Kliinisen päätöksenteon tuen kehittämisen avulla pyritään tarjoamaan terveydenhuollon ammattilaisille näyttöön perustuvia ohjeita ja muistutuksia potilaskertomustietojen pohjalta sekä automatisoimaan terveydenhuollon rutiinitoimintoja. Käynnissä oleva EBMeDS-hanke (Evidence-based medicine clinical decision support) tuottaa tietämystä ja ratkaisuja päätöksentuen hyödyntämiseen sekä arvioi päätöksentuen vaikuttavuutta ja käytettävyyttä. Hankkeeseen liittyy myös päätöksentukiratkaisujen pilotointi eri sairaanhoitopiireissä ja päätöksentukea tarjoavan komponentin tuotteistaminen sekä liittäminen potilastietojärjestelmiin. Päätöksentuesta on olemassa aiemmin tehty prototyyppi (ZipIT-ojo-hanke, Duodecim), jota on kokeiltu diabetes-järjestelmään liittyen. SerAPI-projektissa tutkitaan ja kehitetään ratkaisuja päätöksentuen arkkitehtuuriin ja rajapintoihin. Kohdetta toteutetaan yhteistyössä myös ZipIT- ja ZipIT-ojo sekä eHealth Partners Finland -hankkeiden kanssa.

Määrittelyn luvussa 2 kuvataan valittuihin ratkaisuihin vaikuttaneet taustatekijät, tarkempi toteutuslähtökohtien arviointi ja keskeiset käsitteet. Luvussa 4 kuvataan toimintatapoihin ja oikeuksiin liittyvät vaatimukset. Luvussa 5 esitellään rajapintaratkaisun arkkitehtuuri ja sovellusosapuolet. Luvussa 6 määritellään sovellusosapuolten roolit ja vastuut sekä palvelun käyttötavat. Dokumentissa kuvataan rajapinnan ja tietosisällön sekä niiden taustojen (luku 3) lisäksi päätöksentuen EBMeDS-objektimalli, joka sisältää myös visiotasolla olevia osioita mahdollisesti hyödynnettävissä oleville tietosisällöille. Luku 7 kuvaa varsinaista rajapintamäärittelyä, eli tietosisältöä ja rajapintaa WSDL-kuvauksineen. Luvuissa 8 ja 9 määritellään toteutus- ja jatkokehityskohtaiset asiat.

2 Käsitteet ja tavoitteet

Luvussa 2.1 määritellään päätöksentukeen liittyvät keskeiset käsitteet ja kuvataan valittu etenemistapa. Luvussa 2.2 kuvataan tarkemmat taustakartoitukset, joiden perusteella on arvioitu ja päätetty kohteen edistämisseuranta.

2.1 Määrittelyn tausta, käsitteet ja toteutuslähtökohdat

SerAPI-projektissa on kartoitettu päätöksentuen arkkitehtuuri- ja rajapintaratkaisuja sekä miten päätöksentukikomponentin toteutus on integroitavissa potilaskertomusjärjestelmiin ja millaisiin seikkoihin tulisi kiinnittää huomiota integraation mahdollistamiseksi. Lisäksi taustaselvityksissä on vertailtu tarvittavia rajapintaratkaisuja päätöksentuen ja potilastietojärjestelmän välillä kiinnittäen erityistä huomiota päätöksentuen suorituskykyyn, helppoon liitettävyyteen potilastietojärjestelmiin ja päätöksentukiratkaisujen pitämiseen kompakteina ja yhtenäisinä. Kansainvälisiä ratkaisuja ja erilaisia lähestymistapoja on tarkasteltu Pro gradu -tutkielmassa "Kliinisen päätöksentuen integraation rajapinnat" (Suhonen 2006).

Päätöksentukihankkeen neuvottelukunnan laajennetuissa työkokouksissa elokuussa 2006 päätettyjä keskeisiä linjauksia ovat:

- Päätöksentuen käyttämiseksi sovelletaan kertakutsua, jolloin kaikki päätöksentuen tarvitsemat tiedot lähetetään yhdessä kutsussa. Useita kutsuja käytettäessä muuttunut tietosisältö merkataan sopivalla tavalla, jotta päätöksentukea pystytään soveltamaan muuttuneisiin tietoihin.
- Koodistoihin, tietojen kokoamiseen, toistuvien huomautusten välttämiseen ja sulkulistojen soveltamiseen käytetään asiakaskomponenttia.
- CDA:n sijaan käytetään pilotin Web services-tyyliä (WSDL/SOAP, laajennettavuudesta on kuitenkin huolehdittava).
- Paluuviesteihin on lisättävä päätöksentuen metadata, joka voi sisältää ainakin huomautusten yms. tunnisteet (sulkulistoja varten), linkkejä hoitosuositukseen, tietämyskuvausta, jatkotoimenpiteiden käynnistykseen tarvittavaa rakenteista tietoa jne.
- Koodistoihin liittyen rajapinnassa käytetään koodiston ja sen version yksilöintiä (OID-tunnisteiden avulla), jolloin vastaavuuksia voi määrittellä kumpaan päähän tahansa.

Rajapinnan edistämiseksi on tunnistettu kaksi vaihtoehtoa:

- 1: Käytetään vakiotietopakettia, joka on aina samanlainen. XML-skeema määrittelee varsinaiset tiedot ja mahdollistaa WSDL-tyyppisen ratkaisun sekä helpon automaattisen generoinnin. Tämä tarkoittaa myös sitä, että laajennuksien käyttämisessä kutsujan tulee tietää millä tiedoilla päätöksentukea voidaan kutsua. Palvelu voi joutua tukemaan monia eri skeemaversioita ja tietopaketin mukana on oltava tieto määrittelyn versiosta. Vähentää dokumentaation määrää kohtaan 2 verrattuna, jolloin elementit ovat ymmärrettävempiä nimeämisen avulla.

- 2: Tietoelementtien sovittu nimeämiskäytäntö. XML-skeema ei määrittele varsinaisia tietoja, vaan käytetään CDA-tyyliä, ei kuitenkaan suoraan CDA-lomakkeita. Uusien ja muuttuvien tietojen lisääminen on helpompaa, mutta paketin koneellinen validointi on vaikeampaa. Myös tässä tapauksessa on tiedettävä millä tiedoilla voidaan kutsua, sekä itse tietopakettissa että käytettävissä laajennuksissa. Määrittelyjen palvelun käyttämiselle on oltava tarkempia kohtaan 1 verrattuna.

Näistä vaihtoehdoista on valittu ensimmäinen, jolloin vakiotietosisällön lisäksi lähetettävät tiedot toimitetaan rajapinnassa määritellyn laajennuselementin sisällä (luku 7.2.11).

Keskeisiä taustakartoitusosiossa esiintyviä käsitteitä:

- *Sulkulista:* Käyttäjä- tai potilaskohtaiset sulkulistat määrittelevät, näytetäänkö käyttäjälle tiettyjä varoituksia tai huomautuksia. Sulkulistojen avulla käyttäjä voi esim. estää tiettyjen huomautusten näyttämisen jatkossa itselleen tai tietylle potilaalleen. Sulkulistojen hallinnasta tarkemmin luvussa 6.1.1.
- *Päivityspaketti:* Karsittu yksittäinen dokumentti, tietosisältö tai -elementti, jolla välitetään päätöksentuelle uusia tai muuttuneita tietoja. Oletuksena päivityspakettien hyödyntämisessä on, että päätöksentukipalvelu pitää yllä sessiokohtaista tilaa ja pystyy käsittelemään sille toimitettuja päivitystietoja sekä yhdistämään ne sessiossa käsiteltävään potilaaseen. Tässä rajapintamäärittelyksessä ei ole tukea sessionhallinnan avulla tapahtuvalle päivityspakettien käytölle.

2.1.1 Päätöksentuen ei-toiminnalliset vaatimukset

Päätöksentukihankkeen työkokouksessa (15.8.2006) määriteltyjä keskeisiä näkökohtia päätöksentuen hyödyntämisessä ovat seuraavat:

- Riittävä tehokkuus ja käytettävyys (vasteajat yms.)
- Personoitavuus, toistuvien huomautusten välttäminen
- Joustavuus ja avoimuus, joilla varmistetaan tulevaisuuden kehitys
- Riittävän helppo liitettävyys potilastietojärjestelmiin
- Jo tehdyn työn hyödyntäminen, mm. pilottitoteutukset

Päätöksentukisovelluksen tulee toimia täsmälleen sillä hetkellä kun tarvitaan tietoa päätöksen tai valinnan tueksi. Potilaskertomusjärjestelmän tai sen yhteydessä toimivan sovelluksen (kuten lääketietokanta- tai koodinhakuohjelman) tulee kutsua päätöksentukea (eli lähettää päätöksentukisovellukselle heräte) ja lähettää koodattua potilastietoa. Käynnistystilanteet on määritelty tarkemmin luvussa 2.2.5.

2.2 Toteutuslähtökohtien arviointi

Päätöksentukipalvelun arkkitehtuuriin ja rajapintoihin vaikuttavat monet eri seikat. Tässä luvussa esitellään neljä erilaista näkökulmaa, joita päätöksentukipalvelua rakennettaessa voidaan korostaa. Nämä näkökulmat ovat taustakartoituksia ja ne ovat toimineet pohjana keskusteluille, joiden pohjal-

ta nykyinen etenemissuunta on valittu. Taulukossa 1 esitellään lyhyesti valittavina olevien ratkaisujen vaikutukset tietoliikenteeseen, asiakaspuolen toteutukseen sekä palvelun toteutukseen. Tietoliikenteen kohdalla toteutuksen vaikeutta ei ole lähdetty arvioimaan, koska tietoliikenteen kannalta pääpaino on viestien määrässä ja koossa.

Taulukko 1: Arvio eri ratkaisuvaihtoehtojen vaikutuksista tietoliikenteeseen ja eri osien toteutuksen työmäärään.

Vaihtoehtojen vaikutusten kohteet	Tietoliikenne		Asiakassovelluksen toteutus		Palvelun toteutus	
	kevyt	raskas	helppo	vaikea	helppo	vaikea
CDA		x	x			x
Päätöksentuen CDA (HL7)	x			x	x	
Ei CDA	x			x	x	
Päivityspaketit	x			x		x
Kertakutsu		x	x		x	

Vaihtoehtojen vaikutusten kohteiden kuvaukset:

Tietoliikenne Millainen vaihtoehto on tietoliikenteen kannalta: siirretäänkö paljon turhaa dataa tai tarvitaanko useita viestejä (raskas) vai selvitäänkö pienellä ylikuormalla tai pienikokoisilla viesteillä (kevyt). Painopiste on siirrettävien viestien sisältämässä tietomäärässä.

Asiakaspuolen/palvelun toteutus

Helpossa toteutuksessa ei vaadita mittavaa tietojen hakua ja/tai muunnoksia. Lähetettävät ja vastaanotettavat viestit ovat yksinkertaisia ja tietosisällöltään mahdollisimman suppeita. Käytettävät tiedot ovat sekä sisällöltään että siirtomuodoltaan mahdollisimman lähellä asiakassovelluksessa muutenkin käytettäviä.

Palvelun tilallisuus on välttämätöntä vain päivityspaketteja käytettäessä.

2.2.1 Suurin mahdollinen joustavuus/avoimuus

Avoimessa ja käyttötilanteiden mukaan joustavassa ratkaisussa päätöksentukipalvelu voi sisältää useista eri lähteistä saatavaa tietämystä. Lisäksi tietämyksen ja lähteiden lisääminen järjestelmään on mahdollista useammalta kuin yhdeltä taholta. Päätöksentuen käyttäjä voi itse vaikuttaa siihen, mitä tietämystä ja mitä päätöksentukipalvelua käytetään päätöksentuessa. Asiakassovellus voi myös valita käytettävät koodistot ja koodistojen versiot, tai päätöksentukipalvelu tukee useita eri koodistoja ilman, että asiakassovelluksen tarvitsee huolehtia niiden valinnasta. Myös päätöksenteossa käytettävät tiedot ovat käyttäjän valittavissa. Valinnan voi tehdä esim. paikallisten tietojen ja kansallisen arkiston (ks. luku 2.4) tarjoamien tietojen käytön suhteen.

Tähän tavoitteeseen päästään rajapinnoilla, jotka mahdollistavat eri lähteistä peräisin olevan tietämyksen ja eri päätöksentukipalveluiden valinnan. Tämä vaatii kuitenkin arkkitehtuurin uudelleensuunnittelua: tarvitaan rajapintoja, joilla asiakas voi kysellä ja valita tietämystä tai päätöksentukikomponentteja, sekä avoin rajapinta tietämyksen liittämiseksi päätöksentukeen. Esimerkkinä tällaisesta on HSSP:n (Healthcare Services Specification Project, HL7 ja OMG) määrittelemä DSS (Decision Support Service). Se voidaan käsitteellisesti ymmärtää tietämysmoduulien "vartijana",

jonka jokainen tietämysmoduuli pystyy hyödyntämään koodattua potilastietoa koneellisesti tulkittavien potilaskohtaisten johtopäätösten luomiseksi. DSS-tietämysmoduulin tarkoitus on potilaan arviointi jollakin tietyllä osa-alueella, joka voi olla kapea, esimerkiksi diabetespotilaan sokeri-arvojen tarkistus, tai laaja, kuten lääkkeiden vastavaikutukset uutta reseptiä kirjoitettaessa.

HSSP-ratkaisussa päätöksentukipalvelua kutsuva asiakassovellus määrittelee arvioinnissa käytettävät tietämysmoduulit ja toimittaa moduulien tietovaatimusten mukaiset tiedot. Asiakkaan on tällöin mahdollista valita millaista päätöksentukea on tarkoitus soveltaa. Palautteena DSS-tarjoaja palauttaa potilaskohtaiset päätelmät moduulin etukäteen määrittelemässä muodossa. Tämäntapaisessa potilasarvioinnissa asiakkaan on saatava tuki-informaatiota DSS-tarjoajalta. Tietotarpeita ovat asiakastarpeita vastaavien tietämysmoduulien tunnistaminen, arvioinnin suorittamiseksi tarvittavat tietovaatimukset – eli mitä tietoja asiakkaan tulee toimittaa – ja arvioinnin palautteena syntyvien tulosten tarkoitus sekä muoto. Näiden pohjalta asiakassovellus saa tietää, millaista tietämystä ja päätöksentukea on saatavilla, mitä tietoja päätöksentuki tarvitsee toimiakseen sekä millaisia tuloksia arvioinnista saadaan. (HSSP 2005) Toteutus on monimutkainen sekä asiakassovelluksen että palvelun kannalta.

2.2.2 Helppo liitettävyys potilastietojärjestelmien kannalta

Liittymisen helppous määritellään potilastietojärjestelmiin tarvittavan suunnittelun ja muutosten toteutuksen työmäärän perusteella. Helppointa potilastietojärjestelmän kannalta on välittää sisältämänsä tiedot "sellaisenaan" päätöksentukipalvelulle. Riippuen siitä, käyttääkö nykyinen potilastietojärjestelmä CDA-dokumentteja vai ei, joudutaan pohtimaan eri vaihtoehtoja.

Jos potilastietojärjestelmässä on CDA-tuki, sen sisältämiä dokumentteja voidaan hyödyntää sellaisenaan. Tällöin viestiliikenteen ylikuorma kasvaa, mutta asiakasjärjestelmään ei jouduta tekemään suuria muutoksia; tarvitaan vain päätöksentukipalvelua kutsuva ja potilastietojärjestelmästä (tai arkistosta) CDA-dokumentteja noutava sovellus, ja päätöksentuen kutsujen liittäminen sopiviin kohtiin potilastietojärjestelmää. Päätöksentukipalvelulle välitettävien viestien koostaminen on varsin yksinkertaista, koska siirrettävä tieto on CDA-dokumenteissa, jotka voidaan siirtää sellaisenaan. Tietojen salauksesta on kuitenkin huolehdittava. Hieman monimutkaisempi tapa on koostaa päätöksentuki-CDA -dokumentteja, jolloin potilastietojärjestelmän CDA-ratkaisut ovat osittain hyödynnettävissä (HL7 2006). Oletuksena tässä lähestymistavassa on luonnollisesti, että päätöksentukipalvelu pystyy ottamaan vastaan ja käsittelemään CDA-dokumentteja, joka ei ole tällä hetkellä mahdollista.

Jos potilastietojärjestelmässä ei ole CDA-tukea, helpoin tapa asiakassovelluksen ja päätöksentukipalvelun väliseen tiedonvälitykseen on toteuttaa WSDL/SOAP-rajapinta. Tällaista toteutusta hyödyntää myös prototyyppi. Tällainen tapauskohtainen ratkaisu on käytössä nopea pienen viestiliikenteen ylikuorman myötä, mutta se on altis muutoksille ja siten vaikea ylläpidettävä. Lisäksi koodistojen ja niiden versioiden yhdenmukaisuudesta huolehtiminen joudutaan ratkaisemaan, vaikka siihen olisi valmis osaratkaisu CDA-dokumenteissa.

On huomioitava, että tässä dokumentissa käsitelty lähestymistapa, jossa asiakassovellus kokoaa päätöksentuen tarvitsemat tiedot, ei ole kaikkein helpoin asiakassovelluksen kannalta, jos päätöksentuelle halutaan välittää mahdollisimman kattavat ja ajantasaiset tiedot. Lisäksi esimerkiksi erikoissairaanhoidon potilastietojärjestelmien sisältämät potilastiedot voivat olla suppeita, jolloin lisätietojen haku päätöksentukea varten muista järjestelmistä on välttämätöntä. Potilastietojärjestelmän kannalta helpointa olisikin, jos tietojen hakemisen suorittaisi joko päätöksentukipalvelu tai asiakassovelluksen ja päätöksentukipalvelun välissä toimiva erillinen komponentti.

2.2.3 Helppo siirtyminen nykypilotista

Kaikkein yksinkertaisimmillaan siirtyminen tapahtuu noudattamalla suoraan pilotissa (Komulainen 2004) esitettyjä ratkaisuja. Tässä dokumentissa käsiteltyihin vaihtoehtoihin heijastaen se tarkoittaa käytännössä seuraavaa:

- Potilastietojärjestelmän CDA-dokumentteja ei hyödynnetä suoraan, vaan tiedot siirretään WSDL/SOAP-rajapinnalla, ks. pohdinta yllä.
- Päätöksentukipalvelua käytetään ns. kertakutsuperiaatteella: päivityspaketteja ei käytetä, vaan kaikki tarvittavat tiedot siirretään jokaisella kutsulla. Tämä voi johtaa suorituskykyongelmiin sekä tiedonsiirrossa että suoritettavan tietämyksen valinnassa.
- Asiakassovellus tai siihen liittyvä komponentti kokoaa päätöksentukipalvelun tarvitsemat tiedot.
- Päätöksentukeen tarvittavat tiedot ovat myös tulevaisuudessa samoja kuin rajapintakuvauksessa kuvatut tiedot eikä tietosisältöön tehdä laajennuksia laajennusmekanismien tai päivitettyjen rajapintakuvausten avulla.
- Päätöksentukipalvelun vastaus on tekstimuodossa: rakenteista formaattia (esim. XML) ei käytetä eikä palautteiden vakavuusasteita tai tunnisteita sisällytetä palautteisiin. Näin ei käytetä sulkulistoja eikä automaattisesti päätöksentuen palautteen perusteella potilastietojärjestelmässä käynnistettäviä toimintoja.
- Päätöksentukikomponentti on passiivinen ja nojautuu asiakassovelluksen herätteisiin.

2.2.4 Tehokkuuden maksimointi käyttäjän kannalta

Tehokkuutta voidaan tarkastella päätöksentukiratkaisun suorituskyvyn (vasteajat ym.) sekä käyttäjän kokeman "työn tukemisen tehokkuuden" suhteen. Päätöksentukipalvelun käyttäjän kannalta vasteaikoihin vaikuttavat selkeimmin seuraavat tekijät:

- asiakassovelluksen nopeus
 - tietojen haku potilastietojärjestelmistä
 - päätöksentukipalvelulle lähetettävän viestin muodostaminen
- päätöksentukipalvelun nopeus
 - tietojen purkaminen viestistä
 - suoritettavan tietämyksen valintaan käytetty aika
 - tietämyksen suorittamiseen käytetty aika
 - palvelun kuormitus ja suorituskyky
- asiakassovelluksen ja päätöksentukipalvelun välisen tiedonsiirron nopeus
 - mitä enemmän "turhaa" tai laajaa tietoa, sitä hitaampi siirto
 - tiedonsiirtokanavan nopeus ja kuormitus

Käytettäessä päätöksentuki-CDA:ta ja päivityspaketteja päätöksentukipalvelulle lähetetään vain välttämättömimmät tiedot. CDA:n käyttämisen sijaan voidaan luoda pelkkä SOAP/WSDL-rajapinta. Tällöin CDA-dokumentteja ei siirretä, vaan kaikki tieto siirtyy yksinkertaisissa SOAP-viesteissä, ja turhaa tietoa ei siirretä tai siirretään hyvin vähän. Molemmat ratkaisut ovat asiakkaan, palvelun sekä tiedonsiirron kannalta nopeimpia. Lisäksi, jos asiakassovellus valitsee, mitä tietämystä halutaan suoritettavan, ei päätöksentukipalvelun tarvitse käydä läpi kaikkea mahdollisesti sovel-

lettavaa tietämystä. Tämä kuitenkin monimutkaistaa osien välisiä rajapintoja ja voi monimutkaistaa myös ratkaisun käyttöä (valittava, mitä tietämystä halutaan soveltaa).

Päätöksentukipalvelun toiminta voi hidastua, jos päätöksentukipalvelua varten on haettava tietoja useista eri lähteistä (esim. potilastietojärjestelmä ja kansallinen arkisto). Riippumatta siitä hakeeko tiedot asiakassovellus, palvelu vai jokin kolmas osapuoli, laaja tietojenhaku vie aikaa. Jos päivityspaketteja ei käytetä, päätöksentuki joutuu sovellettavaa tietämystä valitessaan käymään aina läpi kaikki saatuihin tietoihin mahdollisesti sovellettavat skriptit tai tietämyksen käynnistävät tiedot (ei pelkästään muuttuneita). Palvelun käytössä on siis tehtävä valinta nopeuden ja tarkkuuden välillä.

Käyttäjän työn tukemisen tehokkuuteen vaikuttavat lisäksi monet muut seikat. Päätöksentuen palauttamien tietojen perusteella voidaan automatisoida esim. lääkemääräyslomakkeen avaus tai siirtyminen linkin kautta lisätietoihin. Muistutteen suodattaminen sekä tiettyjen päätöksentukitoimintojen poissulkeminen käyttäjän mieltymysten mukaan tehostaa osaltaan käyttöä. Päätöksentuen käynnistystilanteiden ja paluarvojen integrointi saumattomasti osaksi potilastietojärjestelmän käyttöä on erityisen tärkeää koko ratkaisun käytettävyyden ja hyödynnettävyyden kannalta.

2.2.5 Päätöksentuen käynnistystilanteet

Päätöksentukisovelluksen tulee toimia täsmälleen sillä hetkellä kun tarvitaan tietoa päätöksen tai valinnan tueksi. Potilaskertomusjärjestelmän tai sen yhteydessä toimivan sovelluksen (kuten lääketietokanta- tai koodinhakuohjelman) tulee kutsua päätöksentukea (eli lähettää päätöksentukisovellukselle heräte) ja lähettää koodattua potilastietoa. Duodecim (Kunnamo 2005) on määritellyt päätöksentuen käynnistystilanteet, ja ne voidaan kuvata yleisellä käyttötapauksella seuraavasti:

- *Potilaan tietojen avaaminen.* Tässä vaiheessa siirretään kaikki päätöksentuen tarvitsemat potilaskertomuksen ydintiedot. Viimeisimpien tulosten lisäksi voidaan joissakin tapauksissa tarvita arvoja pidemmältä aikaväliltä. Tulosten on oltava käytettävissä lähettävässä järjestelmässä (esimerkiksi laboratoriotulokset ja seuranta-arvot). Jos vastuu tietojen kokoamisesta on asiakassovelluksella, tulee sen saada tiedot myös muista järjestelmistä.
- *Uuden lääkkeen valinta lääkemääräystä varten.* Lääkevalmisteen valinta lääketietokannasta aktivoi päätöksentuen, joka antaa lääkevalinnan laukaisemana suoritettua tietämyksen tuotamat palautteet ennen kuin lääkevalmiste hyväksytään lääkemääräykseen. Siirtyvänä tietona ovat lääkevalmisteen tiedot siinä muodossa kuin ne esiintyvät lääkityslistalla ja sähköisessä reseptissä. Annostukseen liittyvän päätöksentuen tarjoaminen tässä vaiheessa edellyttää, että myös lääkkeen annostus on valittu.
- *Diagnoosin valinta.* Herätteenä on joko uuden diagnoosin vienti potilaan diagnoosilistalle tai aiemmin kirjatun diagnoosin valitseminen aktiiviseksi diagnoosilistalta. Siirtyvänä tietona on diagnoosilistalla oleva diagnoosi ja aiemmat diagnoosit, joilla on sama episoditunnus. Hyödynnettävänä vähimmäistietosisältönä on diagnoosin ICD-10- ja ICPC-2 -koodit (jos tapahtuu automaattinen kaksoiskoodaus), diagnoosin pysyvyys, diagnoosin varmuusaste ja diagnoosin asettamisajankohta (jolla erotellaan uudet ja aiemmin asetetut diagnoosit).
- *Tutkimuksen tai toimenpiteen valinta.* Herätteenä on laboratoriotutkimuksen, muun diagnostisen tutkimuksen tai hoitotoimenpiteen valinta (tilaaminen). Siirtyvänä tietona on tutkimuksen tai toimenpiteen koodi sekä tekoajankohta, jos se on määritetty (ellei ole, oletusarvona on, että tutkimus tehdään heti), sekä muiden pyydettyjen mutta vielä tekemättömien tutkimusten tai toimenpiteiden lista. Tämä lista tarvitaan muista samanaikaisista tutkimuksista (corollary orders) muistuttamista varten.

- *Lähetteen tai konsultaatiopyynnön tekeminen.* Herätteenä on diagnoosin valinta lähetteeseen tai konsultaatiopyyntöön. Siirtyvänä tietona ovat diagnoosilistan tiedot (ks. kohta diagnoosin valinta).

Edellä kuvattujen potilaskertomusjärjestelmästä lähtevien herätteiden ohella tarvitaan kaksi tapaa käynnistää päätöksentukisovellus lääkärin tai muun terveydenhuollon ammattilaisen aloitteesta:

- *Lääkäri käynnistää yksittäisen päätöksentukitoiminnon suorituksen.* Suoritettavan päätöksentukitoiminnon (tietämyksen) käynnistyskomento voi olla linkkinä hoitosuosituksessa, alueellisessa hoito-ohjelmassa (hoitoketjussa) tai mahdollisessa tietämyksenkuvaustietokannasta, jotka lääkäri on hakenut luettavaksi.
- *Kaikkien päätöksentukitoimintojen tai valitun -toimintojoukon suoritus* eräajona valitusta potilasjoukosta (ns. virtuaalinen terveystarkastus). Tämän toiminnon suorittamiseksi kaikista valituista potilaista tulee muodostaa täydellinen päätöksentuen tarvitsema potilastietopaketti. Päätöksentuen tuottamat palautteet tallennetaan joko erilliseen tilapäistiedostoon tai suoraan potilaiden jatkohoitosuunnitelmiin, joihin ne tallennetaan ehdotettuina jatkohoitoina (jälkimmäinen menettely on suositeltavampi ja tulisi ottaa käyttöön siinä vaiheessa kun potilaskertomusjärjestelmässä on rakenteinen jatkohoitosuunnitelma). Kummassakin tapauksessa tallentuu lista niistä potilaista, joista on syntynyt palautetta. Lista voidaan järjestää palautteiden tärkeyden mukaan.

Tietämyksen metatietoihin on suunniteltu liitettäväksi hakusanoiksi ne potilastiedot ja yllä kuvatut hoitoprosessin vaiheet, jotka toimivat herätteenä, eli laukaisevat tietämyksen suorituksen. (muokattu Kunnamo 2005).

2.3 Taustaa: CDA-lomakkeiden käyttö päätöksentuen rajapinnassa

Tässä osiossa kuvataan lyhyesti HL7 Finlandin määrittely CDA:n ja päivityslomakkeiden käyttämiseksi päätöksentuessa sekä arvioidaan CDA:n käyttöä potilas- ja päivitystietojen välityksessä. HUOM. Tämän dokumentin luvuissa 4-8 määrittely ratkaisu ei hyödynnä suoraan CDA-dokumentteja tai päivityspaketti-ratkaisumallia, jota tässä taustaosiossa käsitellään. WSDL:n yksittäisten syötetietojen käyttö rajapinnoissa CDA-lomakemäärittelyjen sijaan auttaa olennaisten koodattujen tietojen hahmottamista tarkan rajapinnan kannalta. Lisäksi WSDL-kuvausten käyttäminen ohjelmointivälineissä tarjoaa selkeämmän API-tyyppisen rajapintamallin, jonka pohjalta toteutusten tuottaminen on yksinkertaista muutenkin sovelluskehityksessä käytetyillä välineillä.

2.3.1 HL7 Finland: Päätöksentuen CDA R2

HL7 Finland ry on tuottanut "Päätöksentuen CDA" -määrittelyn (HL7 2006), jossa kuvataan potilastietojärjestelmän ja päätöksentuen välistä tiedonsiirtoa CDA R2-lomakkeiden ja "päivitystietolomakkeen" avulla. Toteutettu prototyypiratkaisu ei hyödynnä HL7 CDA-määrittelyä.

CDA R2-muotoisissa dokumenteissa ydintiedot rakenteisessa muodossa ovat aina otsikkotasolla entries-osiossa (clinical statements). Ydintietoja voi siis tulla minkä tahansa lomakkeen mukana. Seuraaville päätöksentuessa tarvittaville tietokokonaisuuksille on erityismäärittelyt:

- lähete/hoitopalaute OID: 1.2.246.777.11.2006.9
- lääkityslista OID: 1.2.246.777.11.2006.8

mentit lähetetään paketoituna, dokumenttien paketointi kutsua varten on asiakassovelluksen vastuulla

Tämä lähestymistapa on kuitenkin tietoliikenteen kannalta raskas. CDA-dokumenteissa on paljon ylimääräistä tietoa (esim. potilaan henkilötiedoissa), jonka siirtäminen lisää turhaan tietoliikenteen laajuutta. Lisäksi dokumentin mukana tuleva näyttömuoto lisää turhan tietoliikenteen määrää. Potilastietojärjestelmistä suoraan noudattavat CDA-dokumentit sisältävät myös potilaan yksilöivää tietoa ja henkilötietoja, joten tietoliikenteen tulisi olla salattua.

Päivityspakettien käyttö ei potilastietojärjestelmän CDA-dokumentteja suoraan käytettäessä ole mahdollista, sillä potilaskertomus ei tunne käsitettä "päivityspaketti". Toisaalta siirtymäpolku siihen, että päätöksentuki hakisi esim. kansallisesta arkistosta tai järjestelmien yleisen CDA-pyyntörajapinnan kautta tarvittavia tietoja, helpottuu. Jos päätöksentuki perustuu CDA-dokumenttien siirtoon, erityistä huomiota on kiinnitettävä tietosuojaan, koska CDA-dokumenteista selviää aina kenen tiedoista on kysymys.

CDA-lomakkeissa mukana olevat koodiston tunnisteet ovat olennaisia, koska päätöksentukipalvelun tulee voida tarkistaa, että koodistot ja niiden versiot ovat oikeita (samoja, joita päätöksentuki-toiminnot käyttävät).

2.3.3 CDA:n käyttämättä jättäminen

Jos CDA:ta ei käytetä, on järkevää noudattaa tiedon muodossa nykyisessä prototyypissä käytettyä lähestymistapaa. Vain päätöksentuen kannalta tarpeelliset tiedot siirretään sopivalla tavalla koodatuna, esim. WSDL-parametrimäärittelyillä. Jos päätöksentuessa tarvittavat tiedot voidaan määrittellä kattavasti ja tarkasti, WSDL-rajapintakuvauksesta saadaan nopeasti generoitua pohja, kun päätöksentukea liitetään eri sovelluksiin. Myös tällöin tieto siirtyy järjestelmien välillä SOAP-viestejä käyttämällä, mutta viestit ovat hyvin erilaisia verrattuna CDA:ta sisältäviin SOAP-viesteihin. Lähestymistapaa voidaan käyttää sekä päivityspakettien kanssa että ilman niitä. Jos käytetään päivityspaketteja, erilliset, toisiinsa liittyvät viestit pitää pystyä liittämään toisiinsa.

Kutsujan tulee koostaa lähetettävään viestiin kaikki päätöksentukipalvelun tarvitsemat tiedot. Tiedot potilasjärjestelmässä voivat olla järjestelmän tietokannassa tai myös CDA-dokumenteissa, jolloin kutsujan tulee osata tulkita ko. dokumentit ja etsiä niistä tarvittavat tiedot. Vain olennaisten tietojen lähettäminen tekee tietoliikenteestä tehokasta ja vähän kuormittavaa.

Käsiteltävien tietojen koodaus on välttämätöntä, koska käytettyjen koodistojen tulee olla yhdenmukaiset potilastietojen ja tietämyksen suhteen. Ydintietomäärittelyissä käytettyjen koodistojen noudattaminen on järkevää, vaikka tiedot eivät olisikaan CDA-lomakkeilla.

2.4 Taustaa: Kansallisen arkiston näkökulmat

Valtakunnallisesta arkistosta voidaan tulevaisuudessa hakea potilastietoja ennen päätöksentuen kutsua joko kutsuvan sovelluksen tai väliohjelmiston avulla. Arkistosta ja potilastietojärjestelmästä saatavien tietojen tulee olla erotettavissa toisistaan myös näytettäessä tietoja käyttäjälle. Mikäli tiedonsiirrossa käytetään väliohjelmistoa, sen käyttämiseksi tarvitaan rajapinta. Lisäksi tällöin voi olla mahdollista, että päätöksentukipalvelu hyödyntää itse väliohjelmistoa esimerkiksi virtuaalisessa terveystarkastuksessa.

Sosiaali- ja terveysministeriö (STM 2005) kuvaa arkistossa olevien asiakirjojen käyttöä seuraavasti:

Kaikki terveydenhuollon toimintayksiköt siirtävät valmiit potilasasiakirjat kansalliseen arkistopalveluun rekisterinpitäjittäin samassa elinkaarensa vaiheessa eli asiakirjan syntyhetkellä, millä tarkoitetaan sitä hetkeä, kun asiakirja allekirjoitetaan tai merkitään valmiiksi. Asiakirjojen vastaanotosta arkistoon on aina tultava kuittaustieto siirtäjälle, jotta siirtäjä tietää aineiston menneen perille.

Keskeneräiset asiakirjat säilytetään ensisijaisesti perusjärjestelmissä. Potilastietojen käyttö yli organisaatorajojen potilaan hoitoprosessin tukena vaatii, että asiakirjan arkistointi käynnistyy, kun sen sisältö on tarpeen saattaa muiden toimijoiden käytettäväksi eli kun asiakirja sisältää sellaista tietoa, josta on hyötyä muille toimijoille ja jonka on tarpeen olla muiden toimijoiden käytettävissä. Toisaalta asiakirjan sisältö potilastietojärjestelmässä saattaa täydentyä toistuvasti potilaan hoidon edetessä. Esimerkkinä voivat olla potilaan hoitokertomukseen tehtävät päivittäiset merkinnät, laboratoriotulokset jne.

Keskeneräisten potilasasiakirjojen rekisteröintipalvelu sisältää tiedon työn alla olevista potilasasiakirjoista. Arkistopalvelusta poiketen tämä palvelu siis sisältää tiedot asiakirjoista, joita ei ole valmis-tilassa kansallisessa potilasasiakirjojen arkistopalvelussa. Keskeneräisten potilasasiakirjojen rekisteröintipalvelun tulee sisältää tieto siitä, mistä tietolähteestä löytyy keskeneräistä mahdollisesti muiden terveydenhuollon toimijoiden tarvitsemaa potilaskertomustietoa. Tämän tiedon perusteella voidaan tarvittaessa pyytää arkistoimatonta tietoa organisaatioiden operatiivisista järjestelmistä. Samalla luovutettu keskeneräinen tietosisältö arkistoidaan asiakirjana. (STM 2005)

Keskeneräisten asiakirjojen ja uusimpien tietojen suhteen lisäksi on ratkaistava, miten toimitaan kun tarvitaan esimerkiksi seuranta-arvoja pidemmältä aikaväliltä. Päätöksentuen kannalta tarvitaan sekä pysyvästi arkistoitujen että keskeneräisten potilasasiakirjojen tietoja. Pysyväisarkistosta tehtävissä "virtuaalisissa terveystarkastuksissa" voidaan käyttää vain pysyvästi arkistoituja tietoja.

Tässä rajapintamäärittelyksessä ei ole eroteltu tilanteita, milloin potilastietojärjestelmä (asiakassovelus) tai mahdollinen asiakaskomponentti hakee tietoja kansallisesta arkistosta.

3 Rajapinnan ja tietosisällön taustaa

Luvussa kuvataan taustaa ja periaatteita rajapinnan kutsuun ja vastaukseen vaikuttaviin seikkoihin. Näitä ovat päätöksentuen objektimalli operaatioineen ja skriptien toiminta, luokittelu sekä syntyvät paluuarvot. Objektimallin perusteella muodostettu laajennettu tietosisältö ja rajapinta kuvataan tarkemmin luvussa 7. Aiemmassa pilottiprojektissa toteutetun päätöksentukipalvelun prototyypin abstrakti rajapinta on nimeltään DSService (Komulainen 2004). Rajapinnassa on yksi operaatio, `ExecuteDS`, joka käyttää kahta viestiä:

- `ExecuteDSSRequest` on syöte, joka sisältää potilaan ydintiedot.
- `ExecuteDSSResponse` on palaute, joka palauttaa mahdolliset muistutukset tai varoitukset.

Tähän määrittelyyn operaatiot ja parametrit on korvattu luvussa 7 kuvatulla tavalla.

Komulaisen (Komulainen 2004) mukaan päätöksentukipalvelulle välitetään seuraavat tiedot yhtenä ydintietopakettina:

- `patient`: potilaan perustiedot: syntymäaika, pituus, paino ym.
- `diagnoseList` ICD-10(ICPC), `medicationList`, `allergyList`, `testList`, `procedureList`: apuelementit, jotka sisältävät ne elementit, jotka voivat toistua useammin kuin kerran ydintietopakettissa (esim. kun potilaalla useita lääkityksiä.)
- `diagnoses`: potilaalle tehdyt diagnoosit.
- `medications`: lääkitykset ainesosineen, annosteluineen ym. tietoineen.
- `allergies`: allergiat (lääkeaineille).
- `tests`: potilaalle suoritettut/suoritettavat tutkimukset tuloksineen.
- `procedures`: toimenpiteet.
- `treatmentPlan`: jatkohoitosuunnitelma (tietosisältöä ei vielä määriteltä)

3.1 Kutsun tuottamisen pohjana käytetty objektimalli

Kutsun tietosisältö on johdettu päätöksentuen objektimallista¹, joka kuvataan seuraavassa. Objektimalli sisältää osittain visiotasolla olevia aineksia, joita rajapinnan ensimmäisestä versiosta on jätetty pois. Välttämättömiä osioita ovat tällä hetkellä *Patient*-objekti ja siihen kuuluvat *Characteristics*, *MedicationList*, *ProblemList* ja *TestList*.

Koska CDA:ta ei käytetä suoraan kutsussa, on järkevää noudattaa tiedon muodossa nykyisessä prototyypissä (Komulainen 2004) käytettyä lähestymistapaa. Vain päätöksentuen kannalta tarpeelliset tiedot siirretään sopivalla tavalla koodattuna, WSDL-parametrimäärittelyillä. Jos päätöksentuessa tarvittavat tiedot voidaan määritellä kattavasti ja tarkasti, WSDL-rajapintakuvauksesta saadaan nopeasti generoitua pohja, kun päätöksentukea liitetään eri sovelluksiin. Myös tällöin tieto siirtyy järjestelmien välillä SOAP-viestejä käyttämällä, mutta viestit ovat hyvin erilaisia verrattuna CDA:ta sisältäviin SOAP-viesteihin.

Kutsujan tulisi koostaa lähetettävään viestiin kattavasti päätöksentukipalvelussa tarvittavia tietoja. Tiedot potilasjärjestelmässä voivat olla järjestelmän tietokannassa tai myös CDA-dokumenteissa, jolloin kutsujan tulee osata tulkita ko. dokumentit ja etsiä niistä tarvittavat tiedot.

¹ Luku perustuu pääosin Duodecimin dokumentaatioon ja keskusteluihin Peter Nybergin kanssa.

Käsiteltävien tietojen koodaus on välttämätöntä, koska käytettyjen koodistojen tulee olla yhdenmukaiset potilastietojen ja tietämyksen kesken. Ydintietomäärittelyissä käytettyjen koodistojen noudattaminen on järkevää, vaikka tiedot eivät olisikaan CDA-lomakkeilla.

EBMeDS-objektimalli (<http://www.mrex.fi/EBMeDS/EBMeDSOM/index.html>) on alustava selvitys siitä, mitkä osatekijät ovat keskeisiä päätöksenteon tuessa. Alun perin se perustuu mm. HL7-määrittelyjen sisältöön sekä käytännön kokemuksiin päätöksentukiskriptien notaatiotarpeista. Myöhemmin sitä on pyritty yhdenmukaistamaan Antti Komulaisen (Komulainen 2004) päätöksentukipalvelun Java-luokkien kanssa.

Käytännön skriptityössä on käynyt selväksi, että sähköisen potilaskertomuksen ydintietojen hyödyntämiseksi tarvitaan hyviä objektien metodeja päätöksentukipalvelussa. Esimerkkejä:

1) Ydintietojen mukana tulee potilaan syntymäaika. Skripteissä oleellista on kuitenkin yleensä potilaan ikä vuosissa tai lasten kohdalla kuukausissa, viikoissa, päivissä tai jopa minuuteissa (synnytyksen yhteydessä). Skripteissä on turhaa joka kerta laskea ikä syntymäaikatiedoista, vaan paljon helpompaa on liittää päätöksentukipalveluun metodeja/funktioita kuten `getAgeInYears`, `getAgeInWeeks` jne.

2) Lääkkeiden annostus ilmaistaan ydintiedoissa mm. kerta-annoksella ja annosvälillä. Skripteissä oleellisempaa on useimmiten esim. päiväännos tai viikkoannos, joiden laskemiseksi olisi hyvä olla päätöksentukipalvelussa metodit/funktiot.

Objektimallin sisällössä on valmistauduttu kattavan tietosisällön hyödyntämiseen päätöksentuessa. Tällä hetkellä osaa tiedoista ei ole kuitenkaan vielä tarkasti määritelty. Objektimallin hierarkia kuvataan seuraavassa (EBMeDS 2006):

- [EBMeDS Object Model](#)
- [Guideline](#)
- [Record](#)
 - [CareProvider](#)
 - ▶ [Organization](#)
 - ▶ [Professional](#)
 - [Patient](#)
 - [AidList](#)
 - ▶ [Aid](#)
 - [CertificateList](#)
 - ▶ [Certificate](#)
 - [Characteristics](#)
 - ▶ [Language](#)
 - [ImagingTestList](#)
 - ▶ [ImagingTest](#)
 - [ImplantList](#)
 - ▶ [Implant](#)
 - [MedicationList](#)
 - [Medication](#)
 - ▶ [Dosage](#)
 - [PreventionList](#)
 - ▶ [Vaccination](#)

- ▶ [BirthControl](#)
- ▶ [NutritionalGuidance](#)
- ◀ [ProblemList](#)
- ▶ [DiagnosisList](#)
- ▶ [Diagnosis](#)
- ▶ [ChiefComplaintList](#)
- ▶ [ReasonForContact](#)
- ▶ [ProcedureList](#)
- ▶ [Procedure](#)
- ▶ [ReferralList](#)
- ▶ [Referral](#)
- ◀ [RiskList](#)
- ▶ [OccupationalRiskList](#)
- ▶ [OccupationalRisk](#)
- ▶ [RiskDiseaseList](#)
- ▶ [RiskDisease](#)
- ▶ [RiskFactors](#)
- ▶ [DrugAdverseEffectList](#)
- ▶ [DrugAdverseEffect](#)
- ▶ [TestList](#)
- ▶ [Test](#)
- ◀ [LastResult](#)
- ◀ [PreviousResult](#)
- ▶ [TreatmentPlanList](#)
- ▶ [TreatmentPlan](#)
- ▶ [Encounter](#)
- [Script](#)
- ▶ [Function](#)
- ▶ [Message](#)

EBMeDS-objektimalli (EBMeDS-OM) on hierarkkinen objektimalli EBMeDS-järjestelmästä. Huomattava on, että objektimalli ei kuvaa potilaskertomusta (EPR), vaikkakin objektimallin Record-haara voi sisältää vain elementtejä, jotka on määritelty potilaskertomuksen ydintiedoiksi.

Objektimallin vihreät nuolet ovat objekteja, jotka ovat jo käytössä. Keltaisilla nuolilla merkatut objektit ovat osittain käytössä ja punaisella nuolella merkatut eivät ole vielä käytössä. Rajapintaan määritellyt osat on kartoitettu SerAPI- ja EBMeDS-hankkeiden yhteistyössä, joten täydellistä objektimallia ei kuvata tässä dokumentissa.

3.2 Päätöksentuen palaute

Päätöksentukipalvelimella tuotetut palautteet sisältävät keskeisimpinä asiakassovelluksen käyttäjälle näytettävät huomautukset ja varoitukset tunnisteineen. Paluuviestiä määriteltessä tulee ottaa huomioon myös muita kuin suoraan näytettäviä tietoja. Nämä tiedot voivat olla esim. linkki lisätietoihin tai jonkin toiminnon käynnistäminen asiakassovelluksessa (esimerkiksi päätöksentukitoiminnon ehdottaman laboratoriotutkimuksen pyytäminen). Palautteiden ja niiden mahdollisesti sisältämien muiden tietojen muoto on otettava huomioon arkkitehtuurin eri osissa; esim. kutsuvan asiakassovelluksen on pystyttävä käsittelemään palautteet oikein ja tietämyksen on tuotettava näitä

palautteita sovitussa muodossa. Myös tieto huomautuksiin ja varoituksiin liittyvistä sulkulistoista voisi olla osana palautetta, mutta määritellyn arkkitehtuuriratkaisun perusteella sulkulistoista huolehtii asiakaskomponentti tai asiakassovellus, ja palvelu palauttaa vain tunnisteita ja tietoja, joilla sulkulistoja voidaan hallita.

Palautteen muotoon vaikuttavat palautteiden sisällöt. Runsaasti erilaista tietoa sisältävä palaute on käsittelyn helpottamiseksi paras välittää XML-muodossa. Esiintyvien koodien ja rajapinnan paluuarvojen perusteella asiakasjärjestelmä voi käynnistää jonkin toiminnon, jolloin palautteissa esiintyvien koodien on myös oltava asiakassovelluksen ymmärtämässä ja käsiteltävissä olevassa muodossa. Syntyvissä palautteissa esiintyvät koodit lisäävät vastuuta sekä tietämykselle että asiakasjärjestelmälle. Palautteena voi olla myös potilaskertomuksen tai lähetteen osaksi tarkoitettua tekstiä. Tällainen tietosisältö tulee merkitä tunnisteilla, joista asiakassovellus tietää, mitä sisällöllä tulee tehdä.

Palautteissa voitaisiin käyttää myös HTML- tai XHTML-muotoa, koska palautteiden on tarkoitus sisältää linkkejä hoitosuositukseen. Palautteista muodostuva HTML voi myös toimia suoraan palautteen näyttömuotona. Palautteen osana voi olla hoitosuosituslinkkien lisäksi mahdollista kuvausta tietämyksestä (esim. skriptikuvaus), jolloin on mahdollista tarkastella millaiseen näyttöön perustuvaan tietoon huomautus pohjautuu. Tarkempi kuvaus palautteiden sisällöstä on luvussa 7.3.

3.3 E-Utilities: tietämyksen haku ja katselu linkkien kautta

Käypä Hoito -suositukseen pääsemistä varten on suunniteltu E-Utilities-niminen hakurajapinta, jota voidaan käyttää ja liittää sovelluksiin, jotka tukevat XML:ää ja http-protokollaa (Landgren 2005). Tarkempi dokumentaatio E-Utilities-rajapinnan hyödyntämiseen on saatavilla Duodecimilta. Useimmat ohjelmointikielet ja kehitysvälineet tukevat näitä standardeja joko suoraan tai laajennusten avulla. Eutil-rajapinta koostuu kolmesta proseduurista: haku (search), info, ja näytä (show), joita voidaan kutsua Internetin yli yksinkertaisella http-kutsulla.

Haku-proseduuri suorittaa haun Käypä Hoito -suositusten sisältöön. Proseduuri tarvitse vain yhden parametrin: hakutermit. Kyselyn tulokset järjestetään tietokannan ja relevanssin mukaan ja ne palautetaan yksinkertaisena xml-dokumenttina.

Hoitosuositusten hakuparametrit (molemmat pakollisia):

- p_db - Sovellus, johon haku tehdään.
- p_term - Vapaat hakutermit, kuten astma, a32, tai diabetes.

Info-proseduuri näyttää jotakin tietoa nimetystä artikkelista. Artikkelin tunnus on proseduurin tärkein parametri. Kuten haussa, info palauttaa tuloksensa xml-dokumenttina.

Artikkelin sisältöhaun parametrit (pakollisia):

- p_db - Sovellus, josta infoa etsitään.
- p_id - Artikkelin yksilöllinen tunnus, muotoa ebm12345. Tunnus löytyy kunkin artikkelin lopusta.

Näytä-proseduuria käytetään tietyn artikkelin, hakutulosten tai Käypä Hoito -suosituksen tietyn osion näyttämiseen käyttöliittymässä. Näytä-proseduurilla on useita vapaaehtoisia parametreja, joissa

artikkelin tunnus on yleisimmin käytetty. Toisin kuin haku- ja info-proseduurit, näytä-proseduuri ohjautuu EBMG-sovelluksen käyttöliittymään, joten sitä tulee käyttää vain selainympäristössä. Käyttäjällä tulee myös olla voimassaoleva lisenssi artikkelien lukemiseen.

Artikkelin näyttämisen parametrit:

- p_db - Käytettävä sovellus. Pakollinen.
- p_id - Artikkelin yksilöllinen tunnus, muotoa ebm12345. Vapaaehtoinen, mutta yleisimmin käytetty.
- p_target - Artikkelin sisällön kohdeosa. Vapaaehtoinen.
- p_term - Haku EBMG-käyttöliittymään. Vapaaehtoinen.
- p_browse - Näyttää Käypä Hoito -suosituksen tietyn osan. Vapaaehtoinen. (Landgren 2005)

3.4 Esimerkki tietämyksen suorittamisesta: päätöksentukiskriptit

Tietämyksen suoritustapa kuuluu päätöksentuen sisäiseen toteutukseen. Erilaisia päätöksentukiratkaisuja voidaan toteuttaa mm. neuroverkkojen, tilastollisten menetelmien ja ohjelmointikielten tai skriptien avulla. Tässä luvussa esitetään perusteita päätöksentukiratkaisun sisäiselle logiikalle, kuvaten Suomessa toteutettua skriptipohjaista tietämysmallia esimerkkinä päätöksentuen tietämyksen suorittamisesta ja seikoista, jotka vaikuttavat osin myös rajapintaan.

Päätöksentukipalvelu sisältää skriptitietokannan, jonka skriptien perusteella valitaan päätöksentuen käytettävät tiedot sekä tuotetaan huomautukset ja muut päätöksentuen tulokset. Skriptisyntaksissa on kolme lohkoa.

- *Variable*-lohkossa hyödynnetään objektimallia, joka perustuu HL7-yhdistyksen Avoimet rajapinnat -hankkeessa Kansallisen terveyshankkeen osana tekemiin määrittelyksiin. Muuttujamäärittelyssä annetaan objekteille yksiselitteiset nimet.
- *Message*-lohko: Tässä vaiheessa päätöksentuki tuottaa tekstimuotoisia muistutuksia eikä pyri suoraan vaikuttamaan sähköisen potilaskertomuksen toimintoihin.
- *Flow*-lohko: Toimintalogiikka, joka on yleensä yksinkertaisten if-lohkojen yhdistelmä. Uloin if-lohko sisältää skriptin suorittamisen ehdon, jonka toteutuminen on edellytys sille, että skripti kannattaa ajaa. Tämä tieto säilytetään myös skriptin ulkopuolella skriptitietokannan indeksissä, niin että indeksin perusteella voidaan poimia suoritukseen vain ne skriptit, joiden suoritusehto vastaa dataa on potilaan tiedoissa.

Flow-lohkon viesteissä termit *Reminder*, *Prompt* ja *Alert* määrittelevät skriptin tärkeysasteen / vakavuusasteen. Skriptien tuotannossa on huomioitava muistutusten luokitus ja muistutusten vakavuus.

Skriptiobjektin metodit:

- Run
- setStatus
- setTarget

Function-objektissa olevat skriptin laukaisemat toiminnot

- `getGlobalFunction()`
- `getSubFunction()`
- `getInteractiveForm()`

Päätöksentuen objektimallissa skriptien viestinlähetykseen liittyvät metodit:

- `setAlert`
- `setPrompt`
- `setReminder`
- `addToAlert`
- `addToPrompt`
- `addToReminder`

3.4.1 Käypä hoito -suositukseen ja EBMG-artikkeleihin liittyvät palautteessa hyödynnettävät tiedot

Objektimallissa on kuvattu hoitosuositukseen liittyvät vakio-metadatat.

- *ID* hoitosuosituksen yksilöivä ID-tunnus
 - hoitosuositukset on yksilöity ja luokiteltu:
 - § lääkärin käsikirja
 - § hoitosuositukset
 - § paikalliset hoitosuositukset
- *Source* hoitosuositukselta vastuullinen organisaatio
- *DateFounded* hoitosuosituksen ensimmäisen version julkaisupäivämäärä
- *DateUpdated* viimeisimmän päivityksen päivämäärä
 - 2 aikaleimaa, viimeisin ja perusteellinen päivitys
- *Target* hoitosuosituksen kohdeväestö
 - paikallinen - paikalliset hoito-ohjelmat ja hoitoketjut
 - kansallinen
 - globaali
- linkkitieto
 - syntaksi
 - kuvaava nimi

4 Rajapinnan käytön edellytyksiä ja vaatimuksia

Luvussa kuvataan palvelun käyttöön ja toimintatapoihin liittyvät sekä tekniset periaatteet.

4.1 Oikeuksiin, tietoihin ja toimintatapoihin liittyvät vaatimukset

- Käyttöoikeuksista ja palvelun lisensoinnista sekä tietämyksen ja päätöksentukimoottorin hallinnoinnista vastaa päätöksentukipalvelun tarjoaja.
- Tietojen saatavuus järjestelmistä on asiakassovelluksen vastuulla. Asiakassovelluksen on mahdollista kutsua päätöksentukea osittaisilla tiedoilla (tietosisältöön määritellyt pakollisuudet, luku 7), jolloin on huomioitava ettei päätöksentuki toimi tällöin täydellisesti. Olenneiden varoitusten hyödyntämiseksi tulisi kutsussa toimittaa mahdollisimman täydelliset tiedot.

Koodistojen (ja niiden versioiden) käsittely ja yhdenmukaisuus arkkitehtuurin osien välillä on ratkaistava ja sovittava, jotta tietämystä voidaan hyödyntää suhteessa potilastietoihin. Samat koodistot voivat esimerkiksi olla saatavilla ja käytettävissä kaikille osille. Tämä on mahdollista esimerkiksi alueellisen tai valtakunnallisen koodistopalvelimen avulla, josta kaikkien osien käyttämät koodistot päivitetään yhdenmukaisiksi. Ellei samoja koodistoja käytetä eri osissa, koodistojen yhdenmukaisuus voidaan myös toteuttaa vastaavuuksien määrittelyllä. Vastaavuuksien määrittelyssä esimerkiksi potilastiedoissa käytetty koodi muunnetaan vastaavaa asiaa tarkoittavaksi tietämyksen sisältämäksi koodiksi (toisesta koodistosta tai koodiston versiosta), jolloin lopputuloksena arkkitehtuurin eri osien käyttämät koodistot ovat samoja tietämyksen soveltamisvaiheessa. Vastaavuuksien määrittely voi tapahtua osana päätöksentukipalvelua, osana potilastietojärjestelmää (tai asiakaskomponenttia, ks. luvut 5 ja 6) tai esim. erillistä terminologiapalvelua käyttämällä. Koodistot ja niiden alle lukeutuvat yksittäiset koodit sekä uusimmat koodistoversiot on saatavissa kansalliselta koodistopalvelimelta.

Päätöksentukilogiikan suorituksen yhteydessä päätöksentuki vertaa potilastiedoista löytyviä koodeja tietämyksen sisältämiin koodeihin. Tällöin sekä potilastiedoista että tietämyksestä löytyvien koodien on oltava yhdenmukaisia keskenään.

- Kutsuvan asiakassovelluksen on toimitettava tietosisällössä koodattuihin tietoihin liittyen
 - Koodiarvo
 - Käytettävä koodisto
 - Koodiston versio.

4.2 Tekniikkaan liittyvät vaatimukset

Tässä dokumentissa rajapintaratkaisu on esitetty WSDL/SOAP-tekniikalla. Tämä tarkoittaa rajapintojen kuvaamista Web Services Description Language -kielellä ja rajapintojen käyttämistä Simple Object Access Protocol -protokollalla. Molemmat kielet ovat XML-pohjaisia, ja ne muodostavat yhdessä infrastruktuuritoimittajien suunnalta tuetuimman sovelluspalvelutekniikan hajautettujen Internet-pohjaisten ohjelmistorajapintojen toteuttamiseen.

WSDL ja SOAP ovat alustariippumattomia ja avoimia tekniikoita, joten tässä dokumentissa kuvatut rajapinnat voidaan toteuttaa moneen ympäristöön ja eri tekniikoilla (kuten .NET ja J2EE) tehtyihin ohjelmistoihin. Päätöksentuki on näin liitettävissä uusiin ja vanhoihin sovelluksiin verkkopalveluna.

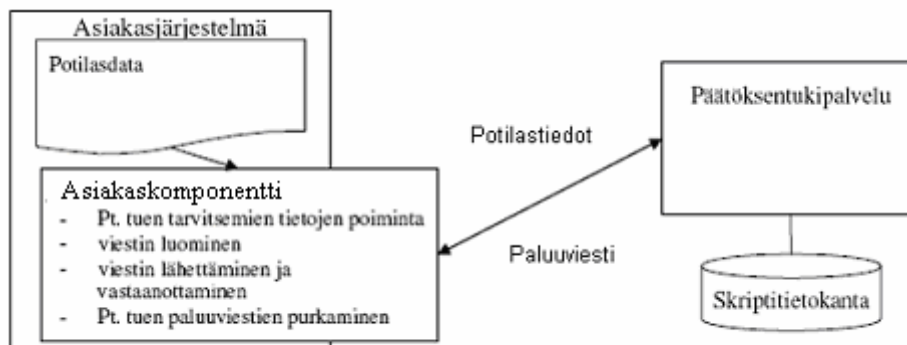
Rajapinnan toteuttamiseen osallistuvien organisaatio-osapuolien tulee toteuttaa WSDL/SOAP-sovellusinfrastruktuuri. Yhteentoimivuuden parantamiseksi toteutuksessa tulisi noudattaa WS-I Basic Profile 1.0 -säännöstöä (SerAPI 2004).

5 Rajapintaratkaisun kuvaus

Tässä luvussa kuvataan päätöksentukirajapinnan perusratkaisu ja periaatteet kahdesta eri arkkitehtuurisesta näkökulmasta: sovellusarkkitehtuuri ja tekninen arkkitehtuuri.

5.1 Sovellusosapuolet ja sovellusarkkitehtuuri

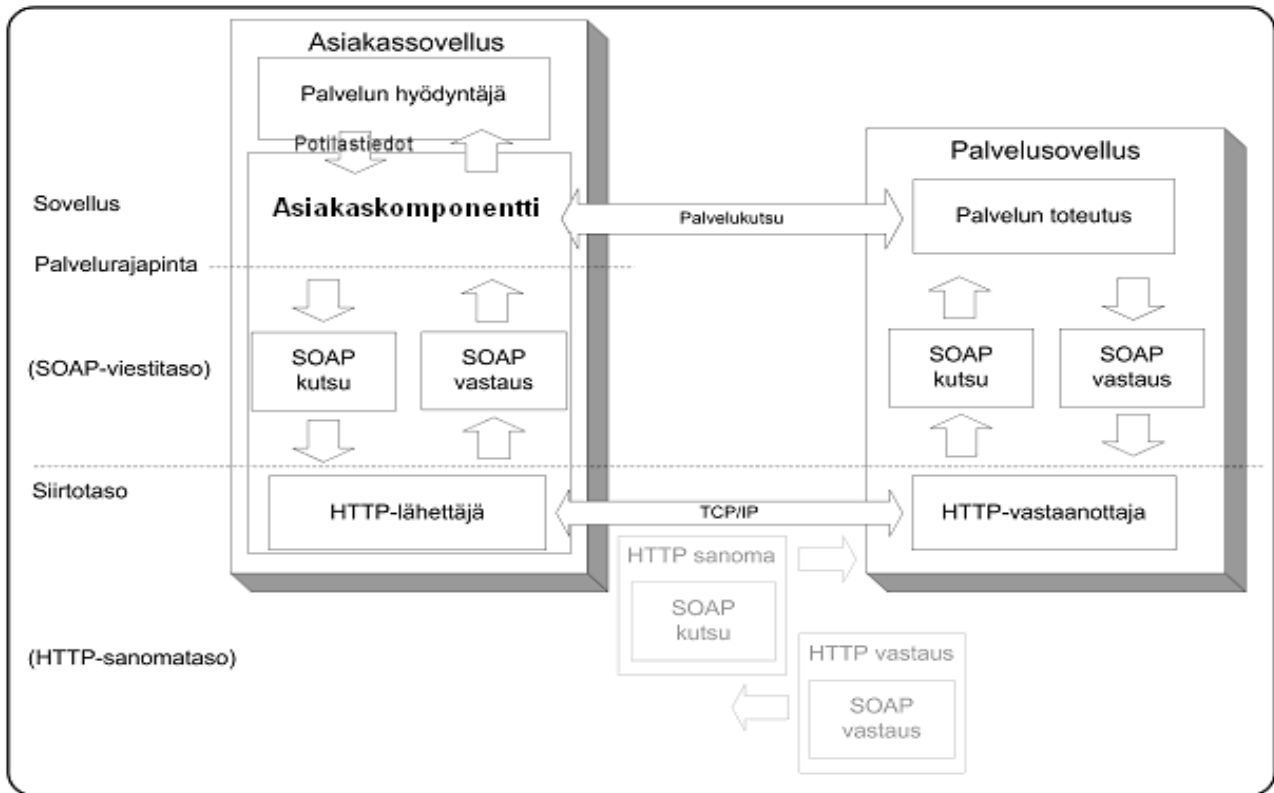
Päätöksentukiratkaisun sovellusarkkitehtuuri on kuvattu kuvassa 1. Rajapinnan näkökulmasta arkkitehtuuri voidaan jakaa kahteen osaan, jotka ovat päätöksentukipalvelu ja asiakasjärjestelmä tai -komponentti. Asiakaskomponentti on kuvassa integroituna asiakasjärjestelmään, eikä tässä määrittelyssä oteta kantaa mahdolliseen asiakasjärjestelmän ja asiakaskomponentin väliseen rajapintaan tai tekniikoihin. Päätöksentukipalvelu taas hyödyntää suoritettavaa tietämystä, joka voidaan toteuttaa esimerkiksi skriptien avulla. Myöskään tämä tietämysrajapinta ei ole osa tätä rajapintaratkaisua. Osien vastuut on kuvattu luvussa 6.



Kuva 1: Toteutetun päätöksentukipilotin perusarkkitehtuuri (muokattu, Komulainen 2004)

5.2 Tekninen arkkitehtuuri

Päätöksentukea käyttävän sovelluksen ja päätöksentukipalvelun välinen tekninen arkkitehtuuri on tyypillinen web services -arkkitehtuuri, jossa käytetään WSDL-rajapintakuvausta ja HTTP-protokollan yli tapahtuvaa SOAP-sanomavälitystä. Kuvan 1 asiakasovelluksena voi siis toimia perusjärjestelmä tai asiakaskomponentti, jonka on huolehdittava myös palvelukutsun vastauksen käsittelystä. Kuvassa 2 asiakaskomponentti on asiakasovelluksen sisällä vastaten useista toiminnoista ja kerroksista; tällöin varsinainen rajapinta kätketään perusjärjestelmältä.



Kuva 2: Tekninen web services -arkkitehtuuri (muokattu SerAPI 2005).

6 Toiminnallisuus

Luvussa kuvataan kutsujan ja palveluntarjoajan vastuut sekä vastuulliset toiminnalliset osat. Luvussa määritellään myös palvelun käyttötavat ja päätöksentukityypit.

6.1 Roolit ja vastuut

Kuvassa 1 esitetystä arkkitehtuurista voidaan tunnistaa seuraavat keskeiset elementit:

- päätöksentukea käyttävä asiakasjärjestelmä eli jatkossa asiakassovellus
- asiakaskomponentti, joka voi vastata monista samoista rajapinnan käyttöön liittyvistä toiminnoista kuin asiakassovellus
- päätöksentuki eli päätöksentukipalvelu tai -komponentti, jota asiakassovellus kutsuu päätöksentukirajapinnan avulla
- tietämys (kuvassa 1 skriptitietokanta), jonka perusteella päätöksentuki tekee johtopäätöksiä ja palauttaa kutsuvalle asiakassovellukselle huomautuksia ja varoituksia
- potilastiedot, jotka voivat olla hajautettuna useampaan kuin yhteen tietovarastoon.

6.1.1 Kutsujan vastuut

Kutsujan vastuulla on tarjota päätöksentukipalvelulle sen tarvitsemat tiedot oikeassa muodossa.

Käyttäjä

Käyttäjällä tarkoitetaan henkilöä, joka käyttää päätöksentukea hyödyntävää potilastieto- tv. järjestelmää. Käyttäjä on henkilö, jolle päätöksentukipalvelun synnyttämät palautteet esitetään.

Asiakassovellus

Päätöksentuen tarvitsemien tietojen tulee olla yhtenäisenä tietopakettina, määritellyssä muodossa, jonka asiakassovellus toimittaa kutsussa.

Käyttäjä- tai potilaskohtaiset sulkulistat määrittelevät, näytetäänkö käyttäjälle tiettyjä varoituksia tai huomautuksia. Sulkulistojen avulla käyttäjä voi esim. estää tiettyjen huomautusten näyttämisen jatkossa itselleen tai tietylle potilaalleen. Tällöin määritellään käyttäjäkohtaisesti (tai jossakin tapauksissa myös potilaskohtaisesti), että tiettyjä huomautuksia ei näytetä tai että huomautuksia tuottavaa päätöksentukea ei suoriteta. Tässä tulee ottaa huomioon, missä tieto sulkulistasta säilytetään (asiakaskomponentti) ja tapahtuuko sulkulistojen huomioiminen käyttäjä- vai sovelluskohtaisesti. Huomioitavaa on myös sulkulistojen muodostaminen, muokkaaminen ja poistaminen sekä toiminnot näiden toteuttamiseksi.

Tieto huomautuksista (esim. tunniste) on osana päätöksentuen palautetta, ja asiakassovelluksen vastuulla on huomautusten poissulkeminen käyttäjäkohtaisen paikallisen sulkulistan perusteella. Koska käyttäjän tunnistaminen kuuluu yleensä potilastietojärjestelmän vastuulle, tämä on suositeltava ratkaisutapa. Se tosin voi aiheuttaa "turhien" päättelyjen suorituksen päätöksentuessa. Jos sulkulistoja käytetään, päätöksentukipalvelun ja tietämyksen toteuttajan vastuulla on liittää tietämykseen tunniste tai koodi (esimerkiksi vakavuusaste, tietämyksen tunnus, yksittäisen palautteen/huomautuksen

tunnus), jonka avulla huomautukset ja varoitukset lisätään sulkulistalle tai esitetään toisistaan erottuvalla tavalla käyttäjälle.

Asiakassovelluksen vastuulla ovat koodistomuunnokset tarvittaessa, mikäli kutsuja ei käytä samaa koodistoa kuin palveluntarjoaja.

Asiakaskomponentti

Päätöksentukea hyödyntävä sovellus voi olla myös asiakaskomponentti. Asiakaskomponentin vastuulla voivat olla monet rajapinnan asiakaspäässä tarvittavat toiminnot. Näitä ovat ainakin päätöksentukipalvelun kutsuminen, sulkulistojen hallinta ja palautteiden käsittely. Asiakaskomponentti ei yleensä ole suorassa yhteydessä loppukäyttäjään, vaan hoitaa ja yksinkertaistaa päätöksentukiliitintä asiakassovelluksen kannalta. Jos asiakaskomponenttia ei ole käytössä, asiakassovellus vastaa toiminnallisuudesta, ks. yllä. Asiakaskomponentin vastuulla voi olla tarpeellisten tietojen poimiminen potilastietojärjestelmästä tai potilastietojärjestelmän toimittamista tiedoista ja muuntaminen päätöksentukipalvelun hyväksymään muotoon. Päätöksentukipalvelun rajapintaa voi hyödyntää asiakassovellus tai asiakaskomponentti, jolloin palvelulle ei ole merkitystä, kumpi sitä kutsuu.

6.1.2 Palveluntarjoajan vastuut

Palveluntarjoajan vastuulla on tarjota päätöksentukipalvelu ja sen käyttämisen mahdollistava rajapinta.

Päätöksentukirajapinta

Rajapinta, joka välittää saamansa tiedot päätöksentukipalvelulle (sovelluspalvelulle) ja palauttaa asiakassovellukselle päätöksentukipalvelulta saamansa palautteet.

Päätöksentukipalvelu

Päätöksentukipalvelu on palvelimella toimiva sovellus, joka tarjoaa asiakkaalle rajapinnan palvelun käyttämiseksi. Päätöksentukipalvelu ottaa vastaan ja purkaa asiakkaan lähettämät viestit, valitsee suoritettavat päätöksentukiskriptit tietokannasta sekä suorittaa ne. Tämän lisäksi päätöksentukipalvelu kokoaa ja lähettää skriptien tuloksena syntyneet palautteet asiakkaalle.

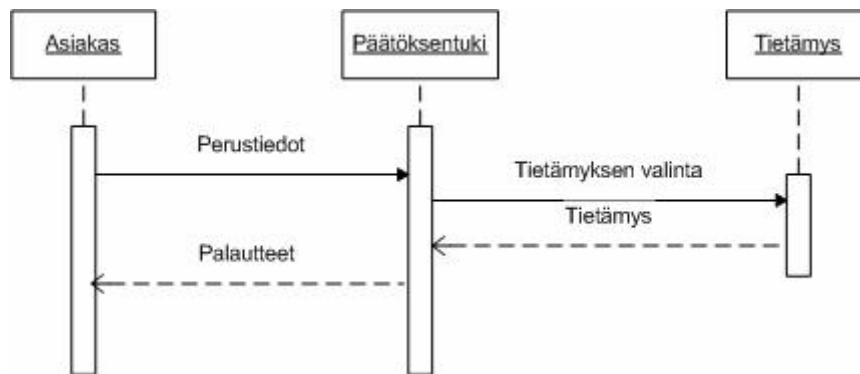
Päätöksentukipalvelun vastuita ovat

- Potilastietojen vastaanottaminen
- Suoritettavien päätöksentukitoimintojen valinta
- Tietämyksen soveltaminen ja päätöksentukitoimintojen suorittaminen
- Palautteiden tuottaminen ja lähettäminen kutsuvalle sovellukselle

6.2 Palvelun käyttötavat

6.2.1 Kertakutsu

Rajapinnassa on yksi operaatio, (luku 7.2), jolla asiakassovellus kutsuu päätöksentukea ja kaikki päätöksentuen tarvitsemat tiedot siirretään kerralla. Operaation paluuarvona saadaan päätöksentukitoimintojen tuottamat palautteet. Tällaista kertakutsuperiaatetta noudattaa myös toteutettu prototyyppi.



Kuva 3: Palvelukutsu

6.2.2 Uusien ja muuttuneiden tietojen toimitus

Rajapinnassa ei ole erillistä operaatiota päivityspakettien käyttämiseen. Jos päätöksentukipalvelulle lähetetään päivitettyjä tietoja, tulee tiedot merkata kutsussa luvussa 7.1 määritellyn attribuutin avulla. Päivitettyjen tietojen hyödyntämistä ei ole tässä vaiheessa määritelty, mutta merkitsemiseen käytetyn attribuutin avulla siihen on varauduttu jo tässä vaiheessa.

6.3 Päätöksentukityypit

Rajapintaan vaikuttavat erilaiset päätöksentukityypit, joiden käyttämiseksi päätöksentukipalvelu joutuu soveltamaan erilaisia kutsun tietosisällön osia. Myös erityyppisen päätöksentuen palautteet ovat erilaisia. Tunnistettuja päätöksentukityyppejä ja rajapintaan, kutsuun ja palautteeseen tällöin vaikuttavia tekijöitä (Päätöksentuen neuvottelukunnan laajennettu työkokous 13.2.2007) ovat seuraavat:

- Linkitys hoitosuositukseen: hoitosuosituksen linkittäminen asiakassovellukseen esimerkiksi diagnoosikoodin avulla E-Utilities -rajapintatyyppisesti (kuvattu luvussa 3.3).
- Lääkehoidon aputietokannat: lääkekoodit ja interaktiot sekä vasta-aiheet, sisältäen myös allergiat lääkeaineille sekä muut lääkitykseen vaikuttavat seikat. Tähän mennessä aputietokantojen hyödyntämistä on sisäänrakennettu eri potilastietojärjestelmiin.
- Monimutkaisempi päätöksentuki (pätöksentukipalvelu): Skriptit ja muu päätöksentukilogiikka, joka voi sisältää myös edelliset sekä tarkempaa päättelyä muista kutsussa päätöksentuelle toimitetuista tiedoista.
- Alueellinen päätöksentuki ja alueelliset hoitosuositukset: Päätöksentukipalvelu voi sisältää ja tarjota hyödynnettäväksi myös alueellisia hoitosuosituksia, joiden käyttäminen vaatii kutsujalta aluetunnuksen. Aluetunnuksen avulla tunnistetaan päätöksentukea kutsuva asiakassovellus, jolloin kutsujalla on oikeus käyttää esimerkiksi omien aluekäytäntöjen mukaan määriteltyä päätöksentukea.

Tämä rajapinta keskittyy päätöksentukipalvelun rajapintaan (kolmas kohta). Ratkaisuissa on kuitenkin huomioitu myös se, että päätöksentuen paluarvoissa voidaan palauttaa linkkejä hoitosuositukseen, päätöksentuki voi palauttaa lääkkeisiin liittyviä interaktio-, allergia- ym. huomautuksia, ja pää-

töksentuessa voidaan huomioida myös alueellista tai paikallista tietämystä. Määritellyn kutsun tietosisältö ottaa huomioon nämä käytötavat, ja on edelleen laajennettavissa.

7 Rajapinta ja sen operaatiot

Tässä luvussa kuvataan päätöksentukirajapinnan operaatiot ja tiedot sekä niissä huomioitavia seikkoja ja käyttötapoja. Rajapinta toimii kertakutsu-periaatteella, jolloin päätöksentuen tarvitsemat tiedot lähetetään yhdellä kertaa kutsun yhteydessä.

Mikäli kertakutsun mukana tarvitaan erikseen tietoa siitä, mitkä tietoelementtejä on muutettu käyttötilanteessa, voidaan kutsun yhteydessä osoittaa uudet tai muuttuneet tiedot. Uusien ja muuttuneiden tietojen toimituksessa sovelletaan "new"-attribuuttia, joka on kuvattu luvussa 7.1.

Luvussa määritellään päätöksentuen palvelukutsussa lähetettävät ja vastauksessa palautettavat tiedot. Kutsussa käytettävät tiedot on muodostettu aiempien päätöksentuen määritysten ja erityisesti päätöksentuen pilottitoteutuksen pohjalta, ja niihin on lisätty CDA-määrityksissä käytettyjä tietoja. Tässä tietosisällön versiossa on huomioitu myös EBMeDSin tuottama päätöksentuen objektimalli ja päätöksentuen demosovellus sekä niiden sisältämät ja käyttämät tiedot. Tietosisällön jaottelua ja rakennetta sekä tietojen merkityksiä on täsmennetty SerAPI- ja EBMeDS-hankkeiden yhteistyössä.

Tietyt osat tietosisällöstä (jatkohoitosuunnitelma, fysiologiset mittaukset) ovat tässä vaiheessa määrittelemättä. Nyt määriteltyihin tietoihin on mahdollista tehdä laajennuksia luvun 7.2.11 mukaisesti. Tällöin laajennettu tietosisältö on kuvattava ja sovittava tarkasti rajapinnan tarjoajan (ja hyödyntäjän) dokumentaatiossa.

Rajapinnan DSSRequest-operaation tarkoituksena on toimittaa potilastiedot päätöksentukipalvelulle sekä palauttaa vastaus kutsuvalle asiakassovellukselle. Kutsussa lähetetään aina samaa muotoa noudattava tietopaketti.

7.1 Vakiotietotyypit ja attribuutit

Tietosisällön koodatuissa tiedoissa käytetään Code-tietotyyppiä, johon määritellään koodiarvon lisäksi käytettävä koodisto ja koodiston versio. Koodatussa tiedossa arvo on aina pakollinen.

Ote WSDL-dokumentista:

```
<complexType name="Code">
  <sequence>
    <element name="CodeValue" type="string"/>
    <element name="CodeSystem" type="string" minOccurs="0"/>
    <element name="CodeVersion" type="string" minOccurs="0"/>
  </sequence>
</complexType>
```

Tietosisällössä käytettävät "new"-attribuutit ilmaisevat muuttuneita tietoja. Kuhunkin tietosisällön elementtiin (listaan), esimerkiksi diagnoosilistaan, liittyy "new"-attribuutti, jonka oletusarvo on false. Jos listalla olevista tiedoista jokin on uusi tai muuttunut, ja jos tätä tietoa voidaan hyödyntää palvelussa, voidaan se ilmoittaa attribuutin arvolla true.

Ote WSDL-dokumentista:

```
<attribute name="new" type="boolean" default="false"/>
```

Lääkityksen määrän ilmaisemiseen käytetään Quantity-tietotyyppiä.

Ote WSDL-dokumentista:

```
<complexType name="Quantity">
  <sequence>
    <element name="Value" type="string" minOccurs="0"/>
    <element name="Unit" type="string" minOccurs="0"/>
  </sequence>
</complexType>
```

Muut tietosisällöissä hyödynnettävät tietotyypit ovat standardeja XML-tietotyyppinä. Tietoihin liittyvät aikaleimat ovat dateTime-tyyppisiä, joilla ilmoitetaan sekä päivämäärä että kellonaika. Jos dateTime-tietotyyppiä halutaan käyttää pelkkien päivämäärien ilmaisemiseen, tulee kellonaikaan liittyvät osuudet korvata nolilla.

7.2 Kutsu (DSSRequest)

Lähetettävä parametri on Record-elementti, johon sisältyvät potilaasta päätöksentuessa hyödynnettävät tiedot (Patient) ja tiedot käynnistä (Encounter).

Ote WSDL-dokumentista:

```
<complexType name="Record">
  <sequence>
    <element name="Patient" type="dss:Patient" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="Encounter" type="dss:Code" minOccurs="0"/>
  </sequence>
</complexType>
```

Patient-elementti sisältää potilastiedot:

- Characteristics - potilaan perustietoja
- MedicationList - lääkitystiedot
- ProblemList - diagnoosit ja ongelmat
- TestList - tiedot tutkimuksista ja niiden tuloksista
- RiskList - riskitiedot
- ProcedureList - tiedot toimenpiteistä ja komplikaatioista
- PreventionList - tiedot rokotteista ja rokotusreaktioista
- PhysioMeasureList - tietoja fysiologisista mittauksista
- HealthFactors - tietoja terveyteen vaikuttavista tekijöistä
- PatientDataExtensionList - mahdolliset tietosisällön laajennukset.

Ote WSDL-dokumentista:

```
<complexType name="Patient">
  <sequence>
    <element name="Characteristics" type="dss:Characteristics"
      minOccurs="0"/>
    <element name="MedicationList" type="dss:MedicationList"
      minOccurs="0"/>
  </sequence>
</complexType>
```

```
        minOccurs="0"/>
    <element name="ProblemList" type="dss:ProblemList" minOccurs="0"/>
    <element name="TestList" type="dss:TestList" minOccurs="0"/>
    <element name="ProcedureList" type="dss:ProcedureList"
        minOccurs="0"/>
    <element name="RiskList" type="dss:RiskList" minOccurs="0"/>
    <element name="HealthFactors" type="dss:HealthFactors"
        minOccurs="0"/>
    <element name="PreventionList" type="dss:PreventionList"
        minOccurs="0"/>
    <element name="PhysioMeasureList" type="dss:PhysioMeasureList"
        minOccurs="0"/>
    <element name="PatientDataExtensionList"
        type="dss:PatientDataExtensionList" minOccurs="0"/>
</sequence>
</complexType>
```

Kutsun Encounter-elementtiin määritellään yhteydenottotapa, eli tieto käynnin tyypistä, joita voivat olla käynti (terveyskeskuksessa), puhelin, sähköposti, internet tai muu yhteydenottotapa.

- Encounter - Yhteydenottotapa
 - Hoitoon pääsy - PTH Yhteydenottotapa 2005 (OID:1.2.246.537.6.42)

Ote WSDL-dokumentista:

```
<element name="Encounter" type="dss:Code" minOccurs="0"/>
```

7.2.1 Characteristics

Potilaan ominaistiedot ovat muuttumattomia tietoja, jotka pysyvät aina samana. Ominaisiedot vastaavat objektimallin *Characteristics*-elementtiä. Objektimallin osat, jotka eivät ole muuttumattomia tietoja, määritellään muihin osioihin, kuten testeihin. Tiedot eivät sisällä elementtejä, joilla potilas voitaisiin tunnistaa yksiselitteisesti, esimerkiksi henkilötunnusta.

- Potilaan ominaisuudet (0...1)
 - syntymäaika (0...1) - birthdate
 - sukupuoli (0...1) - gender
 - luokitus ISO 5218, ISO-OID 1.2.246.537.5.1
 - Ammatti (0...1) - profession
 - Ammattikoodi ja luokitus (TK - Ammattiluokitus 2001, OID: 1.2.246.537.6.74)

Lisäksi objektimallin Characteristics-objektissa on määritelty tietoja kielistä, joista etenkin potilaan äidinkieli on hyödyllinen esimerkiksi testitulosten tv. ilmoittamisen yhteydessä. Kielien koodaamiseen käytetään TK kielikoodi -koodistoa (OID: 1.2.246.537.6.76):

- ContactLanguage - kontaktin kieli
- MotherTongue - äidinkieli

Ote WSDL-dokumentista:

```
<complexType name="Characteristics">
```

```
<sequence>
  <element name="birthdate" type="date" minOccurs="0"
    nillable="true"/>
  <element name="gender" type="dss:Code" minOccurs="0"
    nillable="true"/>

  <element name="ContactLanguage" type="dss:Code" minOccurs="0"
    nillable="true"/>
  <element name="MotherTongue" type="dss:Code" minOccurs="0"
    nillable="true"/>
  <element name="Profession" type="dss:Code" minOccurs="0"
    nillable="true"/>
</sequence>
</complexType>
```

7.2.2 MedicationList

MedicationList on potilaan lääkityslista, johon voi sisältyä yksi tai useampia lääkityselementtejä (Medication). Yhteen potilaaseen voi liittyä vain yksi lääkityslista, mutta lääkityslistan alle voidaan määrittellä useampia lääkityselementtejä. New-attribuutti (ks. 7.1) määrittelee onko listassa käyttötilanteessa muuttuneita tietoja. Lääkitystietoja toimitettaessa lääkekoodi on pakollinen.

Ote WSDL-dokumentista:

```
<complexType name="MedicationList">
  <sequence>
    <element name="Medication" type="dss:Medication" minOccurs="0"
      maxOccurs="unbounded" nillable="true"/>
  </sequence>
  <attribute name="new" type="boolean" default="false"/>
</complexType>
```

Lääkitykseen liittyviä elementtejä ovat seuraavat:

(Lääkityslistan CDA R2-rakenne, V1.07, 12.10.2004)

- **Lääkehoito** (0...n) - Medication
 - lääkekoodi (1...1) - ProductCode
 - luokitus ATC
 - lääkkeen tila - Status
 - boolean-tyyppi, jolla ilmaistaan lisättävä tai muutettava lääke, esim. onko kyseessä uusi lääke lääkityslistalla
 - lääkkeen pysyvyys (0...1) - Permanence
 - boolean-tyyppi: onko kyseessä pysyväislääkitys vai ei
 - o Tarkempaa luokittelua (jatkuva, pysyvä, tarvittaessa, väliaikainen) ei voida toteuttaa tässä vaiheessa koodiston puuttumisen vuoksi
 - lääkityksen aloitus- tai muutospäiväys (1...1) - PrescriptionDate
 - o aikaleima
 - lääkityksen vaikutusaika - EffectiveTime
 - lääkeindikaation (lääkehoidon aloituksen tai muutoksen syyn) koodi (0...1) - Indication
 - luokitus ICD-10 / ICPC / MedDRA
 - lääkityksen lopetuspäiväys (0...1) - EndDate
 - aikaleima
 - lääkkeen vahvuus - Strength
 - lääkkeen vahvuusyksikkö - StrengthUnit

- yleinen aikaleima - TimeStamp

Ote WSDL-dokumentista:

```
<complexType name="Medication">
  <sequence>
    <element name="Dosage" type="dss:Dosage" minOccurs="0"/>
    <element name="EffectiveTime" type="dateTime" minOccurs="0"/>
    <element name="Indication" type="dss:Code" minOccurs="0"/>
    <element name="Permanence" type="boolean" minOccurs="0"/>
    <element name="PrescriptionDate" type="dateTime" minOccurs="0"/>
    <element name="ProductCode" type="dss:Code">
    <element name="Status" type="boolean" minOccurs="0"/>
    <element name="EndDate" type="dateTime" minOccurs="0"/>
    <element name="Strength" type="string" minOccurs="0"/>
    <element name="StrengthUnit" type="string" minOccurs="0"/>
    <element name="TimeStamp" type="dateTime" minOccurs="0"/>
  </sequence>
</complexType>
```

Lääkitykseen liittyvä lääkeannostus on määritelty seuraavasti:

- lääkeannostus (1...1)
 - o Form - antomuoto
 - § yhteisvaikutustietokannassa otetaan huomioon tämä
 - § AR/LÄÄKITYS - Lääkkeenanto tapa 2003 (OID: 1.2.246.777.5.165.2003)
 - o Time - aikaleima
 - o Quantity - määrä
 - § käytännössä esimerkiksi päiväannos (1x40mg / päivä), tiputus esim. mg/min
 - o EffectiveTime
 - § Lääkitysannostuksen päättymishetki
 - o Route - lääkkeenantoreitti
 - § valtimoon/laskimoon/ym.
 - § AR/LÄÄKITYS - Lääkkeenanto reitti 2003 (OID: 1.2.246.777.5.162)

Ote WSDL-dokumentista:

```
<complexType name="Dosage">
  <sequence>
    <element name="Form" type="dss:Code" minOccurs="0"/>
    <element name="Time" type="dateTime" minOccurs="0"/>
    <element name="EffectiveTime" type="dateTime" minOccurs="0"/>
    <element name="Quantity" type="dss:Quantity" minOccurs="0"/>
    <element name="Route" type="dss:Code" minOccurs="0"/>
  </sequence>
</complexType>
```

7.2.3 ProblemList

ProblemList pitää sisällään diagnoosilistan (DiagnosisList) ja kontaktin syy -listan (ReasonForContactList). Lisäksi hoidon syyn (ReasonForTreatmentList) koodia, jos sellainen on luotettavasti käytössä, voidaan hyödyntää mahdollisessa hoitoketjuja ohjaavassa päätöksentukipalvelussa. Hoidon syyn (tulosity), kontaktin syyn ja diagnoosin erottaminen on tärkeää, koska kontaktin syy voi olla

eri, kuin hoidon syy. Hoidon syytä (tulosyytä) ei välttämättä tiedetä, kirjata tai sitä ei synny lainkaan.

Ote WSDL-dokumentista:

```
<complexType name="ProblemList">
  <sequence>
    <element name="DiagnosisList" type="dss:DiagnosisList"
      minOccurs="0"/>
    <element name="ReasonForContactList" type="dss:ReasonForContactList"
      minOccurs="0"/>
    <element name="ReasonForTreatmentList"
      type="dss:ReasonForTreatmentList" minOccurs="0"/>
  </sequence>
</complexType>
```

7.2.3.1 ReasonForTreatmentList

Hoidon syyt määritellään koodeilla, joita voidaan toimittaa listassa yksi tai useampia.

- **Hoidon syyt (0...n) - ReasonForTreatmentList**
 - Hoidon syyn koodi (1...n) - ReasonForTreatmentCode
 - luokitus ICD-10 / ICPC / SHTaL

Ote WSDL-dokumentista:

```
<complexType name="ReasonForTreatmentList">
  <sequence>
    <element name="ReasonForTreatmentCode" type="dss:Code" maxOc-
curs="unbounded"/>
  </sequence>
  <attribute name="new" type="boolean" default="false"/>
</complexType>
```

7.2.3.2 DiagnosisList

Diagnoosilistalla toimitetaan potilaalle tehdyt tai tehtävät diagnoosit, joita voi olla nolla tai useampia. Diagnoosiin liittyvä pakollinen elementti on koodi (Code-tyyppi). Yksi diagnoosi-rakenne voi sisältää eri koodistojen koodeja (esim. ICD ja ICPC), jos käytössä on automaattinen kaksoiskoodaus.

- **Diagnoosi (0...n) - Diagnosis**
 - diagnoosin ensisijaisuus (0...1) - Priority
 - luokitus ISO OID 1.2.246.777.5.40005.2003
 - diagnoosikoodi (1...n) - DiagnosisCode
 - luokitus ICD-10 / ICPC2, MedDRA, ICD-9
 - diagnoosin varmuusaste (0...1) - Certainty
 - luokitus ISO-OID 1.2.246.777.5.40004.2003 (epäily, todennäköinen, varma)
 - diagnoosin pysyvyys (0...1) - Permanence
 - luokitus ISO-OID 1.2.246.777.5.40003.2003 (kertaluonteinen, pysyvä)
 - diagnoosin tyyppi (0...1) - Type
 - luokitus ISO-OID 1.2.246.777.5.40007.2003
 - diagnoosin episodikoodi (0...1) - EpisodeCode

- luokitus ISO-OID (1.2.246.537.10.xxx.14)
- diagnoosin kirjaaja - Creator
- diagnoosin asettamispäiväys (0...1) - SetDate
- diagnoosin poistamispäiväys (0...1) - RemoveDate

Ote WSDL-dokumentista:

```
<complexType name="Diagnosis">
  <sequence>
    <element name="Certainty" type="dss:Code minOccurs="0"/>
    <element name="DiagnosisCode" type="dss:Code" maxOccurs="unbounded"/>
    <element name="Creator" type="string" minOccurs="0"/>
    <element name="EffectiveTime" type="dateTime" minOccurs="0"/>
    <element name="EpisodeCode" type="dss:Code" minOccurs="0"/>
    <element name="Permanence" type="dss:Code" minOccurs="0"/>
    <element name="Priority" type="dss:Code" minOccurs="0"/>
    <element name="SetDate" type="dateTime" minOccurs="0"/>
    <element name="RemoveDate" type="dateTime" minOccurs="0"/>
    <element name="Type" type="dss:Code" minOccurs="0"/>
  </sequence>
</complexType>
```

7.2.3.3 ReasonForContactList

Kontaktien syyt (koodit) määritellään ReasonForContactList-listaan.

- **Kontaktin syy** (0...n) - ReasonForContactList
 - Kontaktin syyn koodi - ReasonForContactCode
 - luokitus ASL, ICD-10, ICPC, tarkennettava toteutuskohtaisesti

Ote WSDL-dokumentista:

```
<complexType name="ReasonForContactList">
  <sequence>
    <element name="ReasonForContactCode" type="dss:Code"
      maxOccurs="unbounded"/>
  </sequence>
  <attribute name="new" type="boolean" default="false"/>
</complexType>
```

7.2.4 RiskList

Riskilista määrittelee potilaan riskitiedot, joita voivat olla esimerkiksi

- Allergiat
- Riskitaudit
- Immuunipuutostilat
- Eristystarve
- Moniresistentit mikrobit
- Radioaktiivinen säteily - riski ympäristölle
- Lääkitykseen ja hoitoon vaikuttavat muut asiat

- Keinoelimet, siirtoelimet, vierasesineet
- Muut riskit ja hoidossa huomioitavat tiedot
- Siirtoelin

Allergiatiedot tarkoittavat potilaalla olevia - ei lääkeaineisiin liittyviä - allergioita. Allergiakoodi on pakollinen tieto.

- **Potilaan riskitiedot** (0..1) - RiskList
 - **Allergia** (0..n) - Allergy
 - allergiakoodi (1..1) - AllergyCode
 - o luokitus ICD-10 / ICPC
 - diagnoosin asettamispäiväys (0..1) - SetDate
 - diagnoosin poistamispäiväys (0..1) - RemoveDate

Lääkeyliherkkyys on päätöksentuen kannalta tärkein tieto. Yliherkkyydellä tarkoitetaan laajempaa käsitettä, jolla käsitetään lääkkeiden haittavaikutuksia. Näihin lukeutuvat: lääkeallergia, yliherkkyys ja sivuvaikutukset. Lääkeyliherkkyteen liittyvissä tiedoissa lääkekoodi on pakollinen.

- **Lääkeyliherkkyys** (0..n) (Lääkityslistan CDA R2-rakenne, V1.07, 12.10.2004)
 - lääkkeen koodi (1..1) - DrugCode
 - o luokitus ATC
 - herkkyys/haittavaikutusreaktion koodi - ReactionCode
 - o luokitus ICD-10 / ICPC
 - diagnoosin asettamispäiväys (0..1) - SetDate
 - diagnoosin poistamispäiväys (0..1) - RemoveDate

Työperäinen riskitieto on saatavissa myös potilaan perustietojen ammattiluokituksen perusteella. Ammattitauteihin liittyvä tarkempi riskitieto (alla) on kuitenkin ammattiluokitusta kattavampi ja tärkeämpi. Riskin koodi on pakollinen.

- **Työperäinen riski** (0..n) - ProfessionRelatedRisk
 - riskin koodi (1..1) - ProfessionRelatedRiskCode
 - o luokitus ICD-10 / ICPC
 - diagnoosin asettamispäiväys (0..1) - SetDate
 - diagnoosin poistamispäiväys (0..1) - RemoveDate

Riskitaudit merkitään pakollisella diagnoosikoodilla. Riskitaudilla tarkoitetaan useimmiten potilaan ympäristölle aiheuttamaa riskiä.

- **Riskitauti** (0..n) - RiskDisease
 - taudin koodi (1..1) - RiskDiseaseCode
 - o luokitus ICD-10 / ICPC
 - diagnoosin asettamispäiväys (0..1) - SetDate
 - diagnoosin poistamispäiväys (0..1) - RemoveDate
- **Keinoelin, siirtoelin, vierasesine** (0..n)
 - keinoelimen koodi (1..1) - ImplantCode
 - o luokitus ICD-10 / ICPC
 - o toimenpideluokitus NCSP
 - diagnoosin asettamispäiväys (0..1) - SetDate

- diagnoosin poistamispäiväys (0...1) - RemoveDate
- **Muu riski (0...n) - Risk**
 - riskin koodi (1...1) - RiskCode
 - o luokitus ICD-10 / ICPC / muut
 - diagnoosin asettamispäiväys (0...1) - SetDate
 - diagnoosin poistamispäiväys (0...1) - RemoveDate
- **Verituotteiden annossa huomioitavaa (0...n)**
 - veriryhmä ABO (0...1) - BloodType
 - veriryhmä Rh (0...1) - RhDBloodType
 - o ISO-OID (1.2.246.777.5.40021.2003) / (1.2.246.777.5.40022.2003)

Ote WSDL-dokumentista:

```
<complexType name="RiskList">
  <sequence>
    <element name="BloodType" type="dss:Code" minOccurs="0"/>
    <element name="RhDBloodType" type="dss:Code" minOccurs="0"/>
    <element name="Allergy" type="dss:Allergy" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="DrugHypersensitivity" type="dss:DrugHypersensitivity"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="ProfessionRelatedRisk"
      type="dss:ProfessionRelatedRisk" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="RiskDisease" type="dss:RiskDisease" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="Implant" type="dss:Implant" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="Risk" type="dss:Risk" minOccurs="0"
      maxOccurs="unbounded"/>
  </sequence>
  <attribute name="new" type="boolean" default="false"/>
</complexType>
```

Allergia

```
<complexType name="Allergy">
  <sequence>
    <element name="AllergyCode" type="dss:Code"/>
    <element name="SetDate" type="dateTime" minOccurs="0"/>
    <element name="RemoveDate" type="dateTime" minOccurs="0"/>
  </sequence>
</complexType>
```

Lääkeyliherkkyys

```
<complexType name="DrugHypersensitivity">
  <sequence>
    <element name="DrugCode" type="dss:Code"/>
    <element name="ReactionCode" type="dss:Code" minOccurs="0"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

```
<element name="SetDate" type="dateTime" minOccurs="0"/>
<element name="RemoveDate" type="dateTime" minOccurs="0"/>
</sequence>
</complexType>
```

Ammattitauti

```
<complexType name="ProfessionRelatedRisk">
  <sequence>
    <element name="ProfessionRelatedRiskCode" type="dss:Code"/>
    <element name="SetDate" type="dateTime" minOccurs="0"/>
    <element name="RemoveDate" type="dateTime" minOccurs="0"/>
  </sequence>
</complexType>
```

Riskitauti

```
<complexType name="RiskDisease">
  <sequence>
    <element name="RiskDiseaseCode" type="dss:Code"/>
    <element name="SetDate" type="dateTime" minOccurs="0"/>
    <element name="RemoveDate" type="dateTime" minOccurs="0"/>
  </sequence>
</complexType>
```

Implantti

```
<complexType name="Implant">
  <sequence>
    <element name="ImplantCode" type="dss:Code"/>
    <element name="SetDate" type="dateTime" minOccurs="0"/>
    <element name="RemoveDate" type="dateTime" minOccurs="0"/>
  </sequence>
</complexType>
```

Muu riski

```
<complexType name="Risk">
  <sequence>
    <element name="RiskCode" type="dss:Code"/>
    <element name="SetDate" type="dateTime" minOccurs="0"/>
    <element name="RemoveDate" type="dateTime" minOccurs="0"/>
  </sequence>
</complexType>
```

7.2.5 HealthFactors

HealthFactors kuvaa terveyteen vaikuttavia tekijöitä. Päätöksentuen kannalta raskaus ja tupakointi ovat tärkeimmät. Tupakoinnista on nähty riittäväksi kyllä/ei -tyyppinen tieto, samoin päihteiden

käyttöön liittyen. Raskaustiedot voidaan päätellä muuta kautta, mutta tässä myös kyllä/ei -tyyppinen luokitus on hyödyllinen. Muihin terveyteen vaikuttaviin tekijöihin, kuten liikunta ja ravitseminen, ei ole määritelty koodauksia, mutta ne ovat mukana merkkijoina, joten niihin voidaan sopia yksinkertainen koodaustapa.

- **Tupakointi** (0..1) - Smoking
 - kyllä (true) / ei (false)
- **Päihitteet** (0..n) - Intoxicants
 - kyllä / ei
- **Liikunta** (0..1) - Exercise
- **Raskaus** (0..1) - Pregnancy
 - kyllä/ei -tyyppinen
- **Ravitseminen** (0..n) - Nutrition
 - tyyppi ja luokitus
 - virheravitseminen, vegaani tms.

Ote WSDL-dokumentista:

```
<complexType name="HealthFactors">
  <sequence>
    <element name="Exercise" type="string" minOccurs="0"/>
    <element name="Intoxicants" type="boolean" minOccurs="0"/>
    <element name="Smoking" type="boolean" minOccurs="0"/>
    <element name="Pregnancy" type="boolean" minOccurs="0"/>
    <element name="Nutrition" type="string" minOccurs="0"/>
  </sequence>
  <attribute name="new" type="boolean" default="false"/>
</complexType>
```

7.2.6 PreventionList

Esimerkki prevention hyödyntämisestä voi olla olemassa olevien rokotteiden vahvistaminen, eli muistutus siitä että potilaan rokotus tarvitsee tehostusta. Preventioita voi olla yksi tai useampia ja ne määritellään preventiolistaan. Myös rokotusreaktiot ovat osa preventiolistaa.

Ote WSDL-dokumentista:

```
<complexType name="PreventionList">
  <sequence>
    <element name="Prevention" type="dss:Prevention"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="new" type="boolean" default="false"/>
</complexType>
```

- **Rokote** (0..n) - Prevention
 - rokotekoodi (1..1) - PreventionCode
 - luokitus ATC
 - rokotteen antamispäiväys - PreventionDate
 - rokotusreaktion koodi (0..1) - ReactionCode
 - luokitus ICD-10
 - rokotusreaktion päiväys - ReactionDate

Ote WSDL-dokumentista:

```
<complexType name="Prevention">
  <sequence>
    <element name="PreventionCode" type="dss:Code" minOccurs="0"/>
    <element name="PreventionDate" type="dateTime" minOccurs="0"/>
    <element name="ReactionCode" type="dss:Code" minOccurs="0"/>
    <element name="ReactionDate" type="dateTime" minOccurs="0"/>
  </sequence>
</complexType>
```

7.2.7 TestList

TestList sisältää nolla tai useampia Test-elementtejä, jotka ovat potilaalle tehtyjä laboratorio- tms. kokeita tai tutkimuksia tuloksineen. Päätöksentuen oletusarvo kutakin testiä kohti palautettavista tutkimustuloksista on 2 viimeisintä tutkimustulosta. Tällöin tutkimuskoodi on sama kahdessa Test-elementissä, mutta aika ja muut tiedot voivat luonnollisesti muuttua. Kunkin tutkimuksen osalta on sovittava tuloksen sekä viitearvojen ylä- ja alarajojen tarkka muoto.

Ote WSDL-dokumentista:

```
<complexType name="TestList">
  <sequence>
    <element name="Test" type="dss:Test" minOccurs="0"
      maxOccurs="unbounded" nillable="true"/>
  </sequence>
  <attribute name="new" type="boolean" default="false"/>
</complexType>
```

(Lähetteen ja hoitopalautteen CDA R2-rakenne, V 2.10, 29.11.2004)

(Laboratoriovastauksien CDA R2-rakenne, V 0.25, 18.11.2004)

- **Tutkimus** (0...n) - Test
 - tutkimuskoodi - TestCode
 - luokitus ISO-OID 1.2.246.537.6.aa.yyyy (aa = tutkimusluokituksen koodi)
 - tutkimustulos - TestResult
 - tutkimuksen päiväys - ResultDateTime
 - Viitearvorajat
 - UNL - yläraja
 - LNL - alaraja

Ote WSDL-dokumentista:

```
<complexType name="Test">
  <sequence>
    <element name="TestCode" type="dss:Code" minOccurs="0"/>
    <element name="LNL" type="string" minOccurs="0"/>
    <element name="TestResult" type="string" minOccurs="0"/>
    <element name="UNL" type="string" minOccurs="0"/>
    <element name="ResultDateTime" type="dateTime" minOccurs="0"/>
  </sequence>
  <attribute name="new" type="boolean" default="false"/>
</complexType>
```

7.2.8 PhysioMeasureList

Fysiologisten mittausten tarkempi määrittely on käynnissä kansallisella tasolla. Tässä vaiheessa fysiologisten mittausten osalta koodaustapaa ei ole olemassa, mutta rajapinnassa on kuitenkin varauduttu siihen, että mittauksille tulee olemaan koodisto tulevaisuudessa. Fysiologisia mittauksia ovat esimerkiksi:

- Pituus, cm
- Paino, kg
- Verenpaine (suositus: yksi mittaus yhdessä PhysioMeasure-elementissä, ylä- ja alapaine erillisissä MeasuredValue-elementeissä).
- Syke

Fysiologisten mittausten tietosisällön rakenne on seuraava:

- **Fysiologinen mittaus (0...n) - PhysioMeasure**
 - mittauskoodi (1...1) - MeasureCode
 - luokitus (pituus / paino / verenpaine / ...)
 - mittausyksikkö (cm, tuuma, kg, jne.) (0...1) - MeasureUnit
 - mittaustulos (1...n) - MeasuredValue
 - mittaustapa (0...1) - MeasureStyle
 - luokitus (seisten / istuen / makuulla / ...)
 - mittauksen aikaleima (1...1) - MeasureDate

Ote WSDL-dokumentista:

```
<complexType name="PhysioMeasure">
  <sequence>
    <element name="MeasureCode" type="dss:Code"/>
    <element name="MeasureUnit" type="dss:Code" minOccurs="0"/>
    <element name="MeasuredValue" type="string" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="MeasureStyle" type="dss:Code" minOccurs="0"/>
    <element name="MeasureDate" type="dateTime" minOccurs="0"/>
  </sequence>
</complexType>
```

7.2.9 ProcedureList

Potilaalle tehty toimenpiteet toimitetaan kutsussa ProcedureList-elementissä, jonka sisällä voi olla nolla tai useampia Procedure-elementtejä. Toimenpiteen status voi olla määritelty tulevaisuuteen, jolloin planned-attribuutin arvoksi määrittyy true.

Ote WSDL-dokumentista:

```
<complexType name="ProcedureList">
  <sequence>
    <element name="Procedure" type="dss:Procedure" minOccurs="0"
      maxOccurs="unbounded"/>
  </sequence>
  <attribute name="new" type="boolean" default="false"/>
</complexType>
```

(Lähetteen ja hoitopalautteen CDA R2-rakenne, V 2.10, 29.11.2004)

- **Toimenpide** (0...n) - Procedure
 - toimenpidekoodi (1...1) - ProcedureCode
 - luokitus esim. Toimenpideluokitus 2007 OID: 1.2.246.537.6.2
 - komplikaation koodi - ComplicationCode
 - luokitus ICD-10
 - päiväys (0...1) - TimeStamp

Ote WSDL-dokumentista:

```
<complexType name="Procedure">
  <sequence>
    <element name="ProcedureCode" type="dss:Code"/>
    <element name="ComplicationCode" type="dss:Code" minOccurs="0"/>
    <element name="TimeStamp" type="dateTime" minOccurs="0"/>
  </sequence>
  <attribute name="planned" type="boolean" default="false"/>
</complexType>
```

7.2.10 Jatkohoito - TreatmentPlan

Jatkohoidon suunnitelmaan kirjatus, ajankohtaisen tai toteutumatta jääneen jatkohoidon havaitseminen ja siitä muistuttaminen terveydenhuollon ammattilaiselle tai suoraan potilaalle ja suunnitellun hoidon edellyttämien prosessien käynnistäminen ovat jatkohoitoon liittyviä päätöksentekitoimintoja. Ennen hoitosuosituksen perustuvien, hoitoa tai seurantaa koskevien muistutusten antamista päätöksentekijärjestelmä tarkistaa, onko ehdotettava asia jo mukana potilaan jatkohoidon suunnitelmassa – jos on, ei (tarpeetonta) muistutetta anneta. Päätöksentuki voi olla paikallistettua, jolloin se mm. tarkistaa, missä suunniteltuja palveluja on saatavilla omalla alueella. (Komulainen 2007)

(Lähetteen ja hoitopalautteen CDA R2-rakenne, V 2.10, 29.11.2004)

- **Jatkohoidon syy** (1...1)
 - jatkohoidon syyn koodi (1...1)
 - luokitus ICD-10 / ICPC

Jatkohoidon suunnitelmaa *ei ole toteutettu rajapintaan*, vaan tietosisällössä voidaan hyödyntää potilaalle tulevaisuudessa tehtäviä tutkimuksia ja toimenpiteitä esimerkiksi merkitsemällä tietosisällössä em. päivämäärä tulevaisuuteen tai planned-attribuutin avulla. Lisäksi kutsun laajennusmekanismi on käytettävissä, kun hoitosuunnitelman rakenne tarkennetaan.

7.2.11 Kutsun laajennus

Kutsun laajennus määritellään PatientDataExtension-elementtinä, jonka attribuutteina ovat itse tietosisältö (AnyPatientData) ja laajennukseen liittyvä ID-tyyppinen tunniste (PatientDataExtensionID). Laajennusten ID-tunnisteet erotellaan ja määritellään attribuutteina. Lähetettäviä laajennuksia voi olla nolla tai useampia. Laajennusten avulla voidaan määritellä uusia tietosisältöjä, kuten jatkohoitosuunnitelma. Kutakin ID-tunnistetta on vastattava esimerkiksi XML Schema -määrittely,

joka kuvaa PatientDataExtension-elementin rakenteen. Vastaavaa mekanismia on hyödynnetty myös Potilaslista-rajapintojen määrittelyssä.

Sovittaminen tiettyihin käyttökohteisiin tapahtuu viestien laajennusosien avulla. Näiden skeemojen avulla määritetään laajennusten tietokentät eli palvelutoteutusten kutsujen ja vasteiden laajennusosioiden xml-elementit. Palveluntoteutus tunnistaa eri soveltamiskohdekohtaiset laajennukset ja kyselylle sovellettavan säännösten kyselyviestissä välitetyn tunnisteiden mukaan.

Määritetty laajennuksen ilmentymä voi periaatteessa olla minkäläinen tahansa xml-dokumentti tai sen osio, joka voidaan skeemalla määrittää ja noudattaa edellisessä luvussa kuvattua, viestin varsinaisessa skeemassa asetettua namespace-rajoitetta. Eli laajennusosiossa täytyy määrittää dokumentille eri nimiavaruus kuin viestin kohdenimiavaruus, jotta viestin ”vakio-osion” ja sen mukana tulevan laajennusosion xml-elementit eivät sekoitu keskenään.

Tämä laajennusosio ja sen skeema eivät ole sidottuna rajapintakuvaukseen tai sen tietokuvaukseen, vaan ne ovat hyödynnettävissä ja toteutettavissa erillään määrittämisessä asetetuista vakio-osioista. Laajennusosion skeema on toteutettava laajennusosion hyödyntämisen mahdollistamiseksi mm. toteutustyössä ja laajennusosion muusta viestistä erillään suoritettavaa validointia varten, sekä liitettävä osaksi sovelluskohdekohtaista dokumentaatiota: soveltamisprofiilia. (SerAPI 2007)

Ote WSDL-dokumentista:

```
<complexType name="PatientDataExtensionList">
  <sequence>
    <element name="PatientDataExtension" type="dss:PatientDataExtension"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="new" type="boolean" default="false"/>
</complexType>
```

Ote WSDL-dokumentista:

```
<complexType name="PatientDataExtension">
  <sequence>
    <any namespace="urn:serapi:DSS"/>
  </sequence>
  <attribute name="PatientDataExtensionID" type="ID" use="required"/>
</complexType>
```

7.3 Vastaus (DSSResponse)

Päätöksenteon kutsuun palautettavan vastauksen (DSSResponse) sisälle määritellään paluuarvot, joita ovat seuraavat:

- ResponseType: Palautteen vakavuusaste (0 tai useampia)
 - *"Reminder", "Prompt" tai "Alert" - skripteille*
- FamilyID: Muistuteperheen tunniste (0 tai useampia)
- ResponseMessageText: Näytettävän palautteen tekstimuoto (0 tai useampia)

- ScriptID: Huomautuksen laukaisseen skriptin tunnus (0 tai useampia)
- ReminderID: Huomautuksen tunnus (0 tai useampia, yksi skripti voi palauttaa useita huomautuksia)
- KnowledgeURL: Linkkitieto tietämykseen (0 tai useampia)

Palautteessa voidaan hyödyntää luvussa 6.3 kuvattuja päätöksentukitapauksia, eli haetaanko esimerkiksi pelkkiä hoitosuosituksia, jolloin esimerkiksi palautteessa oleva skriptitunnus tai linkkitieto voi toimia parametrina, jolla hoitosuositus avataan potilaskertomusjärjestelmään.

Ote WSDL-dokumentista:

```
<complexType name="ResponseMessage">
  <sequence>
    <element name="ResponseMessageText" type="string" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="FamilyID" type="string" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="ScriptID" type="string" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="ReminderID" type="string" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="ResponseType" type="string" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="KnowledgeURL" type="string" minOccurs="0"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

7.4 Virhetilanteet

Päätöksentukipalvelu voi asiakassovelluksen tekemän palvelupyynnön suorittamisen aikana ajautua virhetilanteeseen. Tällöin palautetaan virheilmoitus (*Exception*), joka sisältää seuraavat tiedot:

- virheen tyyppi (*type*) merkkijonona
- mahdollisesti virheen tarkempi kuvaus (*explanation*) merkkijonona

Virhetyyppejä ovat:

- *GeneralFailure* – erittelemätön virhe
- *MissingInputParameter* - pakollinen tieto puuttuu

Virhetilanteessa luodaan SoapException. Itse virheilmoitus on määritelty *Exception*-elementiksi SoapException:n Detail-osaan. Elementti *Exception* sisältää yhden *type-elementin* ja nolla tai yksi *explanation-elementtiä*. Elementti *type* sisältää virhetyypin merkkijonona. Elementti *explanation* sisältää virheen tarkemman kuvauksen merkkijonona.

Ote WSDL-dokumentista:

```
<element name="Exception">
  <complexType>
    <sequence>
```

```
        <element name="type" type="string"/>
        <element name="explanation" type="string" minOccurs="0"/>
    </sequence>
</complexType>
</element>
```

7.5 Esimerkki SOAP-viestistä

Esimerkin SOAP-viestillä kutsutaan päätöksentukipalvelua seuraavilla tiedoilla:

- Potilaan syntymäpäivä
 - 12.3.1991
- Potilaalla oleva lääkitys: lääkityksessä pakollinen tieto on koodiarvo (ATC-koodisto)
 - J01GA01 Streptomysiini
 - A02BC01Omepratsoli
- Potilaalla olevat diagnoosit: diagnoosissa pakollinen tieto on diagnoosikoodi (ICD-10-koodisto)
 - K21.9 Ruokatorven refluksisairaus ilman ruokatorvitulehdusta

SOAP-kutsu:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DSSRequest xmlns="urn:serapi:DSS">
      <Record>
        <Patient>
          <Characteristics>
            <birthdate>1991-03-12</birthdate>
            <gender>
              <CodeValue>1</CodeValue>
              <CodeSystem>1.2.246.537.5.1</CodeSystem>
            </gender>
          </Characteristics>
          <MedicationList>
            <Medication>
              <ProductCode>
                <CodeValue>A02BC01</CodeValue>
                <CodeSystem>1.2.246.537.6.32.2003</CodeSystem>
              </ProductCode>
            </Medication>
            <Medication>
              <ProductCode>
                <CodeValue>J01GA</CodeValue>
                <CodeSystem>1.2.246.537.6.32.2003</CodeSystem>
              </ProductCode>
            </Medication>
          </MedicationList>
          <DiagnosisList>
            <Diagnosis>
              <DiagnosisCode>
                <CodeValue>K21.9</CodeValue>
                <CodeSystem>1.2.246.537.6.1.1999</CodeSystem>
              </DiagnosisCode>
            </Diagnosis>
          </DiagnosisList>
        </Patient>
      </Record>
    </DSSRequest>
  </soap:Body>
</soap:Envelope>
```

```
        </Diagnosis>
    </DiagnosisList>
</Patient>
</Record>
</DSSRequest>
</soap:Body>
</soap:Envelope>
```

8 Toteutuskohtaiset asiat

Tässä luvussa käydään läpi rajapintatoteutusten laatuun ja tehokkuuteen liittyviä asioita. Lisäksi rajapinnan toteuttajien ja hyödyntäjien avuksi käydään läpi asioita, joiden avulla rajapintojen toteuttaja tai hyödyntäjä voi halutessaan varmentaa, vastaako toteutus tätä määrittelyä. Toteutuskohtaisesti tulee myös huomioida vaadittava tietoturvan taso ja sen saavuttamiseksi tarvittavat standardit (esim. HTTPS, WS-I Basic Security Profile).

8.1 Rajapinnan toteutuksessa huomioitavat asiat

Asiakassovelluksen vastuulla on huolehtia siitä, että tiedot välitetään päätöksentukipalvelulle määritellyssä muodossa, esimerkkinä tietojen oikeellisuus ja muoto sekä koodistojen mukaisuus.

Palvelun tarjoaja dokumentoi tuote- ja versio- tai asennuskohtaisesti mm. seuraavat asiat:

- palvelun osoite
- mitä koodistoja ja niiden versioita palvelu edellyttää käytettäväksi ja tukee kutsuissa, ja miten toimitaan asiakassovellukselta vaadittavien koodistojen suhteen (onko palvelua mahdollista kutsua eri koodistoilla, onko käytettävissä asiakaskomponentteja jotka hoitavat myös koodistojen vastaavuuksia jne.),
- mikä on tarkka lääkitys- ja tutkimustietojen koodaustapa, ml. annostukset, tutkimustulokset, viitearvot jne.
- asetetaanko palvelun käyttäjäsovelluksille lisärajoitteita tai -vaatimuksia esim. päätöksentuen massiivisen toistuvan käytön tai muiden palvelun tarjoajaa runsaasti kuormittavien toimintojen suhteen
- käytössä olevat laajennukset
- aiheuttaako palvelulle tuntematon kutsun laajennus aina virhetilanteen

Erillisissä palvelutasosopimuksissa tulisi lisäksi kuvata mm.:

- vastuut palvelun tarjoajan ollessa esim. teknisen vian vuoksi saavuttamattomissa
- tarjottavat tukipalvelut (esim. tekninen neuvonta, virheiden korjaus)

Turvallisuusseikkojen osalta asiakassovelluksen ja palvelun välisessä kommunikoinnissa ei siirretä tietoja, joiden perusteella tiedot voitaisiin suoraan yhdistää tiettyyn henkilöön. Palvelu voi kuitenkin vaatia asiakassovelluksilta tunnistautumis- tai lisensointimekanismien toteuttamista.

8.2 Määrittelyksen mukaisuus

Kohteen rajapintojen toteuttaja tai hyödyntäjä voi halutessaan varmentaa toteutuksen ja tämän määrittelyksen vastaavuuden. Tässä luvussa kerrotaan, mitä kohtia varmentamisessa on käytävä läpi.

Pyynnöissä ja vastauksissa käytettyjen parametrien järjestyksen tulee olla tämän dokumentin mukainen. Lisäksi rajapintaa tarjoavien ja käyttävien sovellusten on noudatettava tämän dokumentin mukaisia standardeja ja koodistoja.

Palvelun toteutus on tämän määrittelyn mukainen, jos sen toiminta ja palauttavat arvot ovat määrittelyn mukaisia. Tämä todetaan testausraportista, joka on tuotettu joko kehitys- tai käyttöympäristössä. Hyväksymistestauksen suorittaminen on suositeltavaa käyttöympäristössä.

Palveluja käyttävän sovelluksen määrittelyjen mukaisuus voidaan todeta testausraportilla, jossa palvelun onnistuneiden kutsujen lisäksi selviää palvelua käyttävän sovelluksen toiminta palvelun palauttaessa virhetilanteita tai ollessa saavuttamattomissa.

Tietojen riittävyys ja kattavuus päätöksentuen soveltamiseksi on tärkeää, mutta vaihtelee myös tilanne- ja tietämyskohtaisesti. Lisäksi nykyisellään ei voida vaatia kaikkia päätöksentuessa mahdollisesti tarvittavia tietoja potilastietojärjestelmiltä yhtenäisessä muodossa. Monien tietojen vapaaehtoisuutta päätöksentuen kutsuissa voidaan vähentää, kun yhdenmukaiset tiedot tulevat käyttöön potilastietojärjestelmissä. Lisäksi on mahdollista kehittää tarkempi tietämyksen ja tietojen valintaprotokolla päätöksentukea varten, kuten esimerkiksi kansainvälisessä Decision Support Services-määrittelyssä Healthcare Services Specification Project-hankkeessa.

9 Määrittelyksen jatkokehitys ja toteutukset

Tämä määrittely on tehty siten, että sen pohjalta voidaan toteuttaa avoimia rajapintoja ohjelmistoihin ja sitä voidaan käyttää esim. julkisissa tarjouspyynnöissä, kun se on julkisesti saatavilla.

Määrittely voidaan julkistaa, hyväksyä standardiksi tai ottaa pohjaksi kehitettäessä uusia versioita SerAPI-projektissa. Uusien versioiden kehittäminen ja standardointi ovat mahdollisia myös julkistetun määrittelyksen pohjalta projektin ulkopuolella.

Määrittelyksen käyttökokemuksia ja mahdollisia korjaus-, muutos- ja jatkokehitystarpeita pyydetään välittämään tekijöille.

Tiedossa olevia käyttötapoihin liittyviä tarkennuksia ovat mm. (nämä seikat syytä dokumentoida toteutuskohtaisesti):

- ovatko riskidiagnoosit aina mukana myös diagnoosilistassa,
- tietojen koodaamistapa ja muoto muutamille päätöksenteossa hyödynnettäville lisätiedoille: tutkimustulokset, ravitsemus, liikunta, jatkohoidon suunnitelma,
- halutaanko päätöksentuen paluarvoissa palauttaa myös kooditettuna suositellun lääkityksen (RecommendedMedication), toimenpiteiden (RecommendedProcedure) tai diagnoosien (RecommendedDiagnosis) (todennäköisyyksineen) arvoja esim. lääkemääräyksen kirjoittamisen automatisointia ym. toimintoja varten.

Rajapinnassa esitetyn laajennusmekanismin avulla rajapintaan on mahdollista liittää uusia kokonaisuuksia, esim. jatkohoidon suunnitelma. Paluarvojen laajentamiseen tarvitaan kuitenkin määrittelyksen jatkokehitystä.

Lähteet

- EBMeDS 2006 EBMeDS-projekti. *Päätöksentuen objektimalli*. 2006.
<http://www.mrex.fi/EBMeDS/EBMeDSOM/index.html>
- HL7 2006 Tarhonen Timo. *Open CDA 2005, Päätöksentuen CDA R2*. Terveyshanke. 28.2.2006.
<http://virtual.vtt.fi/virtual/hl7/cda/05-open-cda2006/ptuki-2.00.doc>
- HSSP 2005 Healthcare Services Specification Project (HSSP): *Service Functional Model, Specification Decision Support Service (DSS) Version 0.43 December 2, 2005 DRAFT*. 2005.
[http://lists.hl7.org/read/attachment/79849/1/HL7%20Decision%20Support%20Ser-vice%20\(DSS\)%20Service%20Functional%20Model%20\(SFM\),%20v0.43.doc](http://lists.hl7.org/read/attachment/79849/1/HL7%20Decision%20Support%20Service%20(DSS)%20Service%20Functional%20Model%20(SFM),%20v0.43.doc)
- Komulainen 2007 Komulainen Jorma. *Jatkohoidon suunnitelma*, luonnos. 2007.
- Komulainen 2004 Komulainen Antti. Päätöksenteon tuki -projekti. *Päätöksenteon tuki: Selvitys päätöksentukipalvelun toteuttamisesta*. 2004.
http://www.kaypahoito.fi/paatoksentuki/ptt_selvitys.pdf.
- Kononen 2005 Kononen Juha: Päätöksenteon tuki -projekti. *Päätöksenteon tuki -hankkeen lisäselvitys: Java-kieliset päätöksentukiskriptit*. 2005.
- Kunnamo 2005 Kunnamo Ilkka: *Potilaskertomuksen ydintietojen hyödyntäminen päätöksentussa - potilaskertomuksen ja päätöksentukiskripteiksi koodattujen hoitosuositusten yhteiset rakenne- ja sanomamäärittelyt*. Suomalainen Lääkäriseura Duodecim, 29.1.2005
<http://www.kaypahoito.fi/paatoksentuki/paatoksentukiraportti.pdf>
- Landgren 2005 Landgren Juuso: *Evidence-Based Medicine Guidelines E-utilities*. Kustannus Oy Duodecim. 2005.
- SerAPI 2007 SerAPI. *Potilaslista - rajapintamäärittely v1.0*. 2007.
http://serapi.fi/mater/Potilaslista_rajapintav10.doc
- Sipilä 2006 Sipilä Mika. Prowellness. SerAPI-päätöksentukityöpaja 7.4.2006.
- STM 2004 Sosiaali- ja terveysministeriö. *Sähköisten potilasasiakirjajärjestelmien toteuttamista ohjaavan työryhmän loppuraportti*, Työryhmämuistioita 2004:18.
<http://www.stm.fi/Resource.phx/publishing/store/2005/01/pr1105954774022/paasthru.pdf>
Liite 1: Potilaskertomuksen ydintiedot.
<http://www.stm.fi/Resource.phx/hankk/hankt/terveyshanke/trm04-18-liite1.htx.i204.xls>
- STM 2005 Sosiaali- ja terveysministeriö: *Kansallinen sähköinen potilastietojen arkistopalvelu, Toiminnallinen vaatimusmäärittely*, Työryhmämuistioita 2005:21.
<http://www.stm.fi/Resource.phx/publishing/store/2020/12/pr1138954229387/pa>

[ssthru.pdf](#)

Suhonen 2006 Suhonen Marko: *Kliinisen päätöksentuen integraation rajapinnat*, Pro gradu -
tutkielma. 2006

Liite 1: WSDL-rajapintakuvaus

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:dss="urn:serapi:DSS"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="DSSv1" targetNamespa-
ce="urn:serapi:DSS">
  <types>
    <schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:dss="urn:serapi:DSS"
targetNamespace="urn:serapi:DSS" elementFormDefault="qualified">
      <element name="DSSRequest">
        <complexType>
          <sequence>
            <element name="Record" type="dss:Record" minOccurs="0"/>
          </sequence>
        </complexType>
      </element>
      <element name="DSSResponse">
        <complexType>
          <sequence>
            <element name="ResponseMessage" type="dss:ResponseMessage" minOc-
curs="0" maxOccurs="unbounded"/>
          </sequence>
        </complexType>
      </element>

      <element name="Exception">
        <complexType>
          <sequence>
            <element name="type" type="string"/>
            <element name="explanation" type="string" minOccurs="0"/>
          </sequence>
        </complexType>
      </element>

      <complexType name="Record">
        <sequence>
          <element name="Patient" type="dss:Patient" minOccurs="0"
maxOccurs="unbounded"/>
          <element name="Encounter" type="dss:Code" minOccurs="0"/>
        </sequence>
      </complexType>

      <complexType name="Patient">
        <sequence>
          <element name="Characteristics" type="dss:Characteristics" minOc-
curs="0"/>
          <element name="MedicationList" type="dss:MedicationList" minOc-
curs="0"/>
          <element name="ProblemList" type="dss:ProblemList" minOccurs="0"/>
          <element name="TestList" type="dss:TestList" minOccurs="0"/>
          <element name="ProcedureList" type="dss:ProcedureList" minOc-
curs="0"/>
          <element name="RiskList" type="dss:RiskList" minOccurs="0"/>
          <element name="HealthFactors" type="dss:HealthFactors" minOc-
curs="0"/>
          <element name="PreventionList" type="dss:PreventionList" minOc-
curs="0"/>
          <element name="PhysioMeasureList" type="dss:PhysioMeasureList" mi-
nOccurs="0"/>
        </sequence>
      </complexType>
    </schema>
  </types>

```

```

        <element name="PatientDataExtensionList" ty-
pe="dss:PatientDataExtensionList" minOccurs="0"/>
    </sequence>
</complexType>

<complexType name="Characteristics">
    <sequence>
        <element name="birthdate" type="date" minOccurs="0" nilla-
ble="true"/>
        <element name="gender" type="dss:Code" minOccurs="0" nilla-
ble="true"/>
        <element name="ContactLanguage" type="dss:Code" minOccurs="0" nilla-
ble="true"/>
        <element name="MotherTongue" type="dss:Code" minOccurs="0" nilla-
ble="true"/>
        <element name="Profession" type="dss:Code" minOccurs="0" nilla-
ble="true"/>
    </sequence>
</complexType>

<complexType name="MedicationList">
    <sequence>
        <element name="Medication" type="dss:Medication" minOccurs="0" ma-
xOccurs="unbounded" nillable="true"/>
    </sequence>
    <attribute name="new" type="boolean" default="false"/>
</complexType>

<complexType name="Medication">
    <sequence>
        <element name="Dosage" type="dss:Dosage" minOccurs="0"/>
        <element name="EffectiveTime" type="dateTime" minOccurs="0"/>
        <element name="Indication" type="dss:Code" minOccurs="0"/>
        <element name="Permanence" type="boolean" minOccurs="0"/>
        <element name="PrescriptionDate" type="dateTime" minOccurs="0"/>
        <element name="ProductCode" type="dss:Code"/>
        <element name="Status" type="boolean" minOccurs="0"/>
        <element name="EndDate" type="dateTime" minOccurs="0"/>
        <element name="Strength" type="string" minOccurs="0"/>
        <element name="StrengthUnit" type="string" minOccurs="0"/>
        <element name="TimeStamp" type="dateTime" minOccurs="0"/>
    </sequence>
</complexType>

<complexType name="Dosage">
    <sequence>
        <element name="Form" type="dss:Code" minOccurs="0"/>
        <element name="Time" type="dateTime" minOccurs="0"/>
        <element name="EffectiveTime" type="dateTime" minOccurs="0"/>
        <element name="Quantity" type="dss:Quantity" minOccurs="0"/>
        <element name="Route" type="dss:Code" minOccurs="0"/>
    </sequence>
</complexType>

<complexType name="ProblemList">
    <sequence>
        <element name="DiagnosisList" type="dss:DiagnosisList" minOc-
curs="0"/>
        <element name="ReasonForContactList" type="dss:ReasonForContactList"
minOccurs="0"/>
    </sequence>
</complexType>

```

```

        <element name="ReasonForTreatmentList" ty-
pe="dss:ReasonForTreatmentList" minOccurs="0"/>
    </sequence>
</complexType>

<complexType name="DiagnosisList">

    <sequence>
        <element name="Diagnosis" type="dss:Diagnosis" minOccurs="0" maxOc-
curs="unbounded" nillable="true"/>
    </sequence>
    <attribute name="new" type="boolean" default="false"/>
</complexType>
<complexType name="Diagnosis">
    <sequence>
        <element name="Certainty" type="dss:Code" minOccurs="0"/>
        <element name="DiagnosisCode" type="dss:Code" maxOc-
curs="unbounded"/>
        <element name="Creator" type="string" minOccurs="0"/>
        <element name="EffectiveTime" type="dateTime" minOccurs="0"/>
        <element name="EpisodeCode" type="dss:Code" minOccurs="0"/>
        <element name="Permanence" type="dss:Code" minOccurs="0"/>
        <element name="Priority" type="dss:Code" minOccurs="0"/>
        <element name="SetDate" type="dateTime" minOccurs="0"/>
        <element name="RemoveDate" type="dateTime" minOccurs="0"/>
        <element name="Type" type="dss:Code" minOccurs="0"/>
    </sequence>
</complexType>

<complexType name="ReasonForContactList">

    <sequence>
        <element name="ReasonForContactCode" type="dss:Code" minOccurs="0"
maxOccurs="unbounded"/>
    </sequence>
    <attribute name="new" type="boolean" default="false"/>
</complexType>
<complexType name="ReasonForTreatmentList">

    <sequence>
        <element name="ReasonForTreatmentCode" type="dss:Code" minOccurs="0"
maxOccurs="unbounded"/>
    </sequence>
    <attribute name="new" type="boolean" default="false"/>
</complexType>
<complexType name="TestList">

    <sequence>
        <element name="Test" type="dss:Test" minOccurs="0" maxOc-
curs="unbounded" nillable="true"/>
    </sequence>
    <attribute name="new" type="boolean" default="false"/>
</complexType>

<complexType name="Test">
    <sequence>
        <element name="TestCode" type="dss:Code" minOccurs="0"/>
        <element name="LNL" type="string" minOccurs="0"/>
        <element name="TestResult" type="string" minOccurs="0"/>
        <element name="UNL" type="string" minOccurs="0"/>
        <element name="ResultDateTime" type="dateTime" minOccurs="0"/>
    </sequence>

```

```

        </sequence>
        <attribute name="new" type="boolean" default="false"/>
    </complexType>

    <complexType name="ProcedureList">
        <sequence>
            <element name="Procedure" type="dss:Procedure" minOccurs="0" maxOc-
curs="unbounded"/>
        </sequence>
        <attribute name="new" type="boolean" default="false"/>
    </complexType>
    <complexType name="Procedure">
        <sequence>
            <element name="ProcedureCode" type="dss:Code"/>
            <element name="ComplicationCode" type="dss:Code" minOccurs="0"/>
            <element name="TimeStamp" type="dateTime" minOccurs="0"/>
        </sequence>
        <attribute name="planned" type="boolean" default="false"/>
    </complexType>

    <complexType name="RiskList">
        <sequence>
            <element name="BloodType" type="dss:Code" minOccurs="0"/>
            <element name="RhDBloodType" type="dss:Code" minOccurs="0"/>
            <element name="Allergy" type="dss:Allergy" minOccurs="0" maxOc-
curs="unbounded"/>
            <element name="DrugHypersensitivity" type="dss:DrugHypersensitivity"
minOccurs="0" maxOccurs="unbounded"/>
            <element name="ProfessionRelatedRisk" ty-
pe="dss:ProfessionRelatedRisk" minOccurs="0" maxOccurs="unbounded"/>
            <element name="RiskDisease" type="dss:RiskDisease" minOccurs="0" ma-
xOccurs="unbounded"/>
            <element name="Implant" type="dss:Implant" minOccurs="0" maxOc-
curs="unbounded"/>
            <element name="Risk" type="dss:Risk" minOccurs="0" maxOc-
curs="unbounded"/>
        </sequence>
        <attribute name="new" type="boolean" default="false"/>
    </complexType>

    <complexType name="Allergy">
        <sequence>
            <element name="AllergyCode" type="dss:Code"/>
            <element name="SetDate" type="dateTime" minOccurs="0"/>
            <element name="RemoveDate" type="dateTime" minOccurs="0"/>
        </sequence>
    </complexType>

    <complexType name="DrugHypersensitivity">
        <sequence>
            <element name="DrugCode" type="dss:Code"/>
            <element name="ReactionCode" type="dss:Code" minOccurs="0"
maxOccurs="unbounded"/>
            <element name="SetDate" type="dateTime" minOccurs="0"/>
            <element name="RemoveDate" type="dateTime" minOccurs="0"/>
        </sequence>
    </complexType>

    <complexType name="ProfessionRelatedRisk">
        <sequence>

```

```

        <element name="ProfessionRelatedRiskCode" type="dss:Code"/>
        <element name="SetDate" type="dateTime" minOccurs="0"/>
        <element name="RemoveDate" type="dateTime" minOccurs="0"/>
    </sequence>
</complexType>

<complexType name="RiskDisease">
    <sequence>
        <element name="RiskDiseaseCode" type="dss:Code"/>
        <element name="SetDate" type="dateTime" minOccurs="0"/>
        <element name="RemoveDate" type="dateTime" minOccurs="0"/>
    </sequence>
</complexType>

<complexType name="Implant">
    <sequence>
        <element name="ImplantCode" type="dss:Code"/>
        <element name="SetDate" type="dateTime" minOccurs="0"/>
        <element name="RemoveDate" type="dateTime" minOccurs="0"/>
    </sequence>
</complexType>

<complexType name="Risk">
    <sequence>
        <element name="RiskCode" type="dss:Code"/>
        <element name="SetDate" type="dateTime" minOccurs="0"/>
        <element name="RemoveDate" type="dateTime" minOccurs="0"/>
    </sequence>
</complexType>

<complexType name="HealthFactors">
    <sequence>
        <element name="Exercise" type="string" minOccurs="0"/>
        <element name="Intoxicants" type="boolean" minOccurs="0"/>
        <element name="Smoking" type="boolean" minOccurs="0"/>
        <element name="Pregnancy" type="boolean" minOccurs="0"/>
        <element name="Nutrition" type="string" minOccurs="0"/>
    </sequence>
    <attribute name="new" type="boolean" default="false"/>
</complexType>

<complexType name="PreventionList">
    <sequence>
        <element name="Prevention" type="dss:Prevention" minOccurs="0" ma-
xOccurs="unbounded"/>
    </sequence>
    <attribute name="new" type="boolean" default="false"/>
</complexType>

<complexType name="Prevention">
    <sequence>
        <element name="PreventionCode" type="dss:Code" minOccurs="0"/>
        <element name="PreventionDate" type="dateTime" minOccurs="0"/>
        <element name="ReactionCode" type="dss:Code" minOccurs="0"/>
        <element name="ReactionDate" type="dateTime" minOccurs="0"/>
    </sequence>
</complexType>

<complexType name="PhysioMeasureList">

```

```

        <sequence>
            <element name="PhysioMeasure" type="dss:PhysioMeasure" minOccurs="0"
maxOccurs="unbounded" />
        </sequence>
        <attribute name="new" type="boolean" default="false" />
    </complexType>

    <complexType name="PhysioMeasure">
        <sequence>
            <element name="MeasureCode" type="dss:Code" />
            <element name="MeasureUnit" type="dss:Code" minOccurs="0" />
            <element name="MeasuredValue" type="string" minOccurs="0" maxOc-
curs="unbounded" />
            <element name="MeasureStyle" type="dss:Code" minOccurs="0" />
            <element name="MeasureDate" type="dateTime" minOccurs="0" />
        </sequence>
    </complexType>

    <complexType name="PatientDataExtensionList">

        <sequence>
            <element name="PatientDataExtension" type="dss:PatientDataExtension"
minOccurs="0" maxOccurs="unbounded" />
        </sequence>
        <attribute name="new" type="boolean" default="false" />
    </complexType>
    <complexType name="PatientDataExtension">
        <sequence>
            <any namespace="urn:serapi:DSS" />
        </sequence>
        <attribute name="PatientDataExtensionID" type="ID" use="required" />
    </complexType>
    <complexType name="ResponseMessage">
        <sequence>
            <element name="ResponseMessageText" type="string" minOccurs="0" ma-
xOccurs="unbounded" />
            <element name="FamilyID" type="string" minOccurs="0" maxOc-
curs="unbounded" />
            <element name="ScriptID" type="string" minOccurs="0" maxOc-
curs="unbounded" />
            <element name="ReminderID" type="string" minOccurs="0" maxOc-
curs="unbounded" />
            <element name="ResponseType" type="string" minOccurs="0" maxOc-
curs="unbounded" />
            <element name="KnowledgeURL" type="string" minOccurs="0" maxOc-
curs="unbounded" />
        </sequence>
    </complexType>

    <complexType name="Code">
        <sequence>
            <element name="CodeValue" type="string" />
            <element name="CodeSystem" type="string" minOccurs="0" />
            <element name="CodeVersion" type="string" minOccurs="0" />
        </sequence>
    </complexType>

    <complexType name="Quantity">
        <sequence>
            <element name="Value" type="string" minOccurs="0" />

```

```

        <element name="Unit" type="string" minOccurs="0"/>
    </sequence>
</complexType>

</schema>
</types>
<message name="DSSRequestMessage">
    <part name="parameters" element="dss:DSSRequest"/>
</message>
<message name="DSSResponseMessage">
    <part name="parameters" element="dss:DSSResponse"/>
</message>
<message name="DSSExceptionMessage">
    <part name="fault" element="dss:DSSException"/>
</message>
<portType name="DSSv1ServiceOperations">
    <operation name="DSSRequest">
        <input message="dss:DSSRequestMessage"/>
        <output message="dss:DSSResponseMessage"/>
        <fault name="DSSException" message="dss:DSSExceptionMessage"/>
    </operation>
</portType>
<binding name="DSSv1ServiceSOAPBinding" type="dss:DSSv1ServiceOperations">
    <wsdlsoap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="DSSRequest">
        <wsdlsoap:operation soapAction="urn:serapi:DSS#DSSRequest"/>
        <input>
            <wsdlsoap:body use="literal"/>
        </input>
        <output>
            <wsdlsoap:body use="literal"/>
        </output>
        <fault name="DSSException">
            <wsdlsoap:fault name="DSSException" use="literal"/>
        </fault>
    </operation>
</binding>
<service name="DSSv1Service">
    <port name="DSSv1ServiceSOAPPort" binding="dss:DSSv1ServiceSOAPBinding">
        <wsdlsoap:address location="http://localhost:8080/codeserverWS/services/DSSv1Service"/>
    </port>
</service>
</definitions>

```