

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Open CDA 2006**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Tiedonsiirto ja tiedon välitys**

**Versio 2.1.4**

**28.2.2006**

**OID: 1.2.246.777.11.2006.6**

**Versiohistoria:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versio:** | **Pvm:** | **Laatijat:** | **Muutokset:** |
| 0.1 | 20.12.2004 | ti | Raportin luonnos |
| 0.2 | 27.12.2004 | ti | Täydennystä |
| 0.3 | 12.1.2005 | ti | Tulleiden kommenttien huomioonottaminen |
| 0.4 | 15.1 | ti | Tulleiden kommenttien mukaisia täydennyksiä |
| 1.0 | 15.1.1005 | ti | Scheman täydennys, viimeistely |
| 2.0 | 15.12.2005 | ti | Tiedonvälityksen toiminnallisuus lisätty |
| 2.1 | 21.12.2005 | ti | Pyynnön synkroninen toimintamalli lisätty Poikkeustilanteet ja niissä menettelyt, perustana dokumentti "HSS virhetilanteet, Timo Kaskinen Itella Suomi ja Timo Lukumaa Reaktor Innovations, 6.10.2005". |
| 2.1.1 | 10.1.2006 | ti | Tulleiden kommenttien huomioonottaminen, pieniä täsmennyksiä |
| 2.1.2 | 12.1.2006 | ti | Poikkeustilanteista toipumisen hienosäätöä |
| 2.1.3 | 30.1.2006 | ti | Tiedonvälityksessä on sallittu käyttää myös PULL-mallia, muta sitä ei tässä dokumentissa määritellä.  Luettelo Service- ja Action-parametreista erotettu omaksi dokumentikseen ylläpidon helpottamiseksi |
| 2.1.4 | 16.2.2006 | ti | Kommenttien perusteella täydennykset |
|  |  |  |  |

ae= Antero Ensio

ti = Timo Itälä

tt = Timo Tarhonen

av= Aino Virtanen

**SISÄLLYSLUETTELO**

[1. Johdanto 5](#_Toc128121686)

[2. Vuorovaikutustilanteet 6](#_Toc128121687)

[3. Siirtosyntaksi 6](#_Toc128121688)

[3.1 Yleinen rakenne 6](#_Toc128121689)

[3.2 Kuljetuskehys 8](#_Toc128121690)

[3.3 MessageHeader 8](#_Toc128121691)

[3.3.1 From (lähettäjä) 10](#_Toc128121692)

[3.3.2 TO (vastaanottaja) 11](#_Toc128121693)

[3.3.3 CPAId (sovellusmäärittelyn tunnus) 11](#_Toc128121694)

[3.3.4 ConversationId (transaktion tunnus) 11](#_Toc128121695)

[3.3.5 MessageId:n ja ConversationId:n käytöstä 11](#_Toc128121696)

[3.3.6 Service (palvelu) 13](#_Toc128121697)

[3.3.7 Action (tapahtuma) 13](#_Toc128121698)

[3.3.8 MessageData (sanoman tunnistetiedot) 13](#_Toc128121699)

[3.4 AckRequested 14](#_Toc128121700)

[3.5 Acknowledgment 14](#_Toc128121701)

[3.6 HL7FIAcknowledgment 15](#_Toc128121702)

[3.7 HL7FIBodyCount 15](#_Toc128121703)

[3.8 SOAP-Headerin elementtien pakollisuus 16](#_Toc128121704)

[3.8.1 CPAId – sovellusmäärittelyn tunnus 16](#_Toc128121705)

[3.8.2 Service – palvelu ja action – tapahtuma 16](#_Toc128121706)

[3.9 Palvelukehys 16](#_Toc128121707)

[4. Vuorovaikutustilanteet ja sanomat 17](#_Toc128121708)

[4.1 Yleisperiaatteet 17](#_Toc128121709)

[4.2 Aineiston lähetys 18](#_Toc128121710)

[4.3 Aineiston vastaanoton kuittaukset 18](#_Toc128121711)

[4.3.1 Kuittaussanomat 18](#_Toc128121712)

[4.3.2 Onnistuneen vastaanoton kuittaus 19](#_Toc128121713)

[4.3.3 Virheilmoitukset ja -koodit 20](#_Toc128121714)

[4.3.4 Kuittaus- ja toipumismenettelyt 23](#_Toc128121715)

[4.4 Yksilöidyn xml-dokumentin pyyntö 23](#_Toc128121716)

[4.5 Xml-dokumenttien pyyntö lomaketunnuksella 24](#_Toc128121717)

[4.6 Xml-dokumentti vastauksena kyselyyn 25](#_Toc128121718)

[4.7 Tiedonsiirtoyhteyden toimivuuden testaussanoma 26](#_Toc128121719)

[5. Siirtomenetelmät 27](#_Toc128121720)

[5.1 Siirron luonne 27](#_Toc128121721)

[5.2 http(s) 27](#_Toc128121722)

[5.3 Siirtotason kuittaukset 28](#_Toc128121723)

[5.4 Yhteenvetokuvat tietovirroista ja niiden kuittauksista 30](#_Toc128121724)

[5.4.1 Asynkroninen lähetys 30](#_Toc128121725)

[5.4.2 Synkroninen lähetys 31](#_Toc128121726)

[5.4.3 Aineiston pyyntö (synkroninen) 32](#_Toc128121727)

[5.5 Sanomien salaus 32](#_Toc128121728)

[5.6 Xml-dokumenttien version hallinta 33](#_Toc128121729)

[6. Tiedonvälitys 33](#_Toc128121730)

[7. Tarve tiedonvälitykselle 34](#_Toc128121731)

[7.1 Tietoliikenneadapterit ja sovellusadapterit 34](#_Toc128121732)

[7.2 Tietoliikenneadapterien parametrointi 34](#_Toc128121733)

[7.3 Tiedonvälityspalvelun tarve 35](#_Toc128121734)

[7.4 Rajaukset 36](#_Toc128121735)

[8. Tiedonvälityksen konfigurointi 36](#_Toc128121736)

[9. Tiedonvälityksen toiminnallisuus 37](#_Toc128121737)

[9.1 Vuorovaikutustilanteet 37](#_Toc128121738)

[9.2 Asynkroninen ja synkroninen tiedonsiirto 38](#_Toc128121739)

[9.3 Aineiston lähetys 38](#_Toc128121740)

[9.4 Aineiston lähetys yksityiskohtaisesti 39](#_Toc128121741)

[9.5 Aineiston pyyntö 39](#_Toc128121742)

[9.5.1 Aineiston asynkroninen pyyntö yksityiskohtaisesti 40](#_Toc128121743)

[9.5.2 Aineiston pyyntö synkronisesti 42](#_Toc128121744)

[9.6 Kuittausten käsittelyn yksityiskohtia 43](#_Toc128121745)

[9.7 Virheilmoitukset ja -koodit 43](#_Toc128121746)

[9.8 Menettely erilaisissa poikkeustilanteissa 44](#_Toc128121747)

[10. Muita toteutusnäkökohtia 47](#_Toc128121748)

[11. PULL-malli 48](#_Toc128121749)

# Johdanto

HL7 Avoimet Rajapinnat- dokumentissa (Avoin\_Rajapinta\_V1.1.2, 3.2.2003) määritellään täsmällisesti tietojen siirtäminen perusjärjestelmien ja viitetietojärjestelmien välillä (perustyö Timo Tarhonen ja Esko Eloranta, Tietotarha Oy). HL7 OpenCDA- dokumentissa (Open\_CDA\_V1.0.doc, 2.2.2004) yleistetään tätä määrittelyä tietojen siirtämiseen yleensä terveydenhuollon tietojärjestelmien välillä eri käyttötarkoituksissa. Tavoitteena on, että määrittelyt muodostavat valtakunnallisen standardin, jonka mukaisesti potilastietojärjestelmät vaihtavat tietoja keskenään. Näiden määritysten yksityiskohtia on lisäksi täydennetty useammassa eri dokumentissa ja saatteessa. Tähän dokumenttiin on koottu yhteen tiedonsiirron toteuttamiseksi tarvittavat määritykset.

Järjestelmien välinen tiedonsiirto voidaan jakaa kahteen tasoon: Ensimmäisen tason muodostaa tiedon lähettäminen ja vastaanottaminen ja toisen tason muodostaa lähetettävän tiedon kokoaminen lähetystä varten ja vastaanotetun tiedon jatkokäsittely. HL7 OpenCDA -dokumentissa on määritelty yleiskäyttöinen toiminnallisuus tiedon lähettämisen ja vastaanottamisen tasolle. Lähettämistä ja vastaanottamista varten toteutettua toiminnallisuutta voidaan kutsua **tietoliikenneadapteriksi.** Tietoliikenneadapteri on tarpeen toteuttaa tietojärjestelmään vain kerran. Tietoliikenneadapteria voidaan käyttää eri käyttötarkoituksia varten koottujen tietosisältöjen lähettämiseen ja vastaanottamiseen tietojärjestelmien välillä.

Kutakin tietojen siirtämisen käyttötarkoitusta varten on määriteltävä toiminnallisuus lähetettävien tietojen kokoamiseksi ja vastaanotettujen tietojen käsittelemiseksi. Tätä toiminnallisuutta voidaan kutsua **sovellusadapteriksi**. Näitä käyttötarkoituksia ovat mm. seuraavat:

* + - * Potilastietojärjestelmä kokoaa viitetiedot potilaiden tiedoista ja lähettää ne viitetietojärjestelmälle, joka vie ne viitetietokantaan
      * Käyttäjä valitsee potilaan viitteen, jonka viitetietojärjestelmä lähettää potilastietojärjestelmälle. Tämä muodostaa vastaavat potilastiedot sisältävän CDA dokumentin ja lähettää sen viitetietojärjestelmälle, joka esittää dokumentin sisällön käyttäjälle
      * Käyttäjä kirjaa lähetteen potilastietojärjestelmään. Se kokoaa lähetteestä CDA dokumentin ja lähettää sen vastaanottavaan potilastietojärjestelmään, joka asettaa lähetteen vastaanotettujen lähetteiden työjonoon.
      * Potilastietojärjestelmä kokoaa potilaan hoitoprosessista seurantatiedot hoidon saatavuuden seurantaa ja tilastointia varten. Se lähettää seurantatiedot Stakesin vastaanottokeskukseen, joka vie aineiston raportoinnin tietokantaan.
      * Lääkäri laatii työtapaturma- tai liikennevahinkopotilaasta ilmoituksen hoitoon otosta sekä e-lausunnon ja maksusitoumuspyynnön potilastietojärjestelmään. Se lähettää ne vakuutusyhtiön tietojärjestelmään, jossa korvauskäsittelijä ottaa ne käsittelyyn.
      * Potilastietojärjestelmä kokoaa potilaan kertomustiedot ja lähettää ne arkistoon, joka tallettaa ne pitkäaikaista säilytystä varten.
      * Lääkäri laati lääkemääräyksen potilastietojärjestelmään, joka lähettää sen Kelan reseptitietokantaan.

Kutakin käyttötarkoitusta varten tarvittavan sovelluksen toiminnallisuuden määritteleminen ja toteuttaminen on asianomaisten tietojärjestelmien toimittajien ja käyttäjien tehtävä. Tässä dokumentissa kuvattavat määritykset on tarkoitettu tietoliikenneadapterin toteuttamista varten.

# Vuorovaikutustilanteet

Sovellusten välisiä vuorovaikutustilanteita on periaatteessa kaksi tärkeätä tilannetta: Tiedon lähetys ja tiedon pyyntö.

|  |
| --- |
|  |
| Kuva 1: Tiedon lähetys |
|  |
| Kuva 2: Tiedon pyyntö |

Lähetys ja pyyntö ovat yleisnimityksiä vuorovaikutustilanteille, joihin kuhunkin liittyy tapauskohtaisia yksityiskohtia. Kerrallaan siirrettävä aineisto voi muodostua yhdestä tai useammasta tietoalkiosta. Tiedonsiirto voi olla luonteeltaan synkronista, jolloin lähettävä järjestelmä odottaaa, kunnes saa kuittauksen vastaanottaneelta järjestelmältä tai asynkronista, jolloin lähettävä järjestelmä ei jää odottamaan kuittausta vaan myöhemmin saa sen vastaanottaneelta järjestelmältä. Tiedonsiirtoon liittyy myös täsmälliset määritykset siitä, miten osapuolten välinen vuorovaikutus etenee poikkeustilanteissa, esimerkiksi virheelliseksi osoittautuvan tiedon tapauksissa. Nämä vuorovaikutuksen yksityiskohdat kuvataan jäljempänä.

# Siirtosyntaksi

## Yleinen rakenne

Aineistot siirretään SOAP -kehystettyinä, varsinainen hyötykuorma (payload) sijaitsee SOAP:in body-osuudessa palvelukehyksessä. Kuljetukehyksessä eli SOAP:in header-osuudessa on reititystietoja, joiden tietosisältö on suurelta osin peräisin ebXML:stä, mutta se ei ole varsinaista ebXML:ää. Käytettävä SOAP-versio on 1.1. Sitä käytetään web services-tyyppisesti siten, että SOAP-Bodyn alussa on on luokan ja metodin tunnus.

Oheinen kuva esittää SOAP-kehystetyn sanoman rakennetta.

|  |
| --- |
|  |
| Kuva 3: Tiedon siirron yleisrakenne |

Varsinainen hyötykuorma on XML-dokumentti. Se voi olla CDA R1 tai CDA R2 dokumentti, mutta se voi olla muukin XML-tiedosto kuten MEDDIS-pohjainen lähete/hoitopalautesanoma tai jokin HL7 V3 sanoma.

Kuljetuskehyksellä ja varsinaisella hyötykuormalla on omat erilliset schemansa. Se mahdollistaa kuljetuskehyksen validoinnin ja käsittelyn riippumatta kuljetettavan sanoman sisällöstä. Kuljetuskehys on kaikille sanomille yhteinen ja sen käsittely voidaan toteuttaa yleiskäyttöisellä tietoliikenneadapterilla.

## Kuljetuskehys

Avoimet Rajapinnat -tiedonsiirrossa SOAP-ENV:Header sisältää reititykseen ja liikenteen hallintaan tarvittavia elementtejä. Niitä varten on oma schemansa, joka muodostaa varsinaisen Avoimet Rajapinnat -kuljetuskehyksen scheman. Schemat ovat raportin liitteenä.

|  |
| --- |
|  |
| Kuva 4: Kuljetuskehyksen schema |

Käyttämämme kuljetuskehyksen Header-määrittelyt on rakennettu siten, että elementit lainataan pääosin ebXML:stä.

Header koostuu seuraavista elementeistä, joissa HL7FI-alku osoittaa alkuperäksi avoimet adapterit - projektin, ei ebxml:n:

* + - * MessageHeader
      * AckRequested
      * Acknowledgment
      * HL7FIAcknowledgment
      * HL7FIBodyCount

Headerin tarkka rakenne riippuu sanoman käyttötarkoituksesta, esim. Acknowledgment ja HL7FIAcknowledgment esiintyvät pelkästään kuittaussanomissa.

## MessageHeader

Tämä header-elementti esiintyy kaikissa sanomissa. Oheinen kuva esittää sen rakennetta.

|  |
| --- |
|  |
| Kuva 5: MessageHeader schema |

Sen schemaan sisältyvät seuraavat elementit:

<element name="MessageHeader">  
 <complexType>  
 <sequence>  
 <element ref="ar:From"/>  
 <element ref="ar:To"/>  
 <element ref="ar:CPAId"/>  
 <element ref="ar:ConversationId"/>  
 <element ref="ar:Service"/>  
 <element ref="ar:Action"/>  
 <element ref="ar:MessageData"/>  
 </sequence>  
 <attribute ref="SOAP-ENV:mustUnderstand"/>  
 </complexType>  
</element>

Ohessa on esimerkki MessageHeader-elementistä:

<ar:MessageHeader SOAP-ENV:mustUnderstand="1">

<ar:From>

<ar:PartyId>1.2.246.10..lähettävä organisaat io.18.2002.1

</ar:PartyId>

<ar:Role>adapter</ar:Role>

</ar:From>

<ar:To>

<ar:PartyId>1.2.246.10..viitetietojärjestelmäpalvelu.18.2002.1

</ar:PartyId>

<ar:Role>refdb</ar:Role>

</ar:To>

<ar:CPAId>1.2.246.777.11.2004.1</ar:CPAId>

<ar:ConversationId>1</ar:ConversationId>

<ar:Service>refdb</ar:Service>

<ar:Action>R2HEADERS</ar:Action>

<ar:MessageData>

<ar:MessageId>1</ar:MessageId>

<ar:Timestamp>2003-01-07T08:23:00</ar:Timestamp>

</ar:MessageData>

</ar:MessageHeader>

### From (lähettäjä)

<element name="From">  
 <complexType>  
 <sequence>  
 <element ref="ar:PartyId"/>  
 <element name="Role" type="ar:non-empty-string"/>  
 </sequence>  
 </complexType>  
</element>

<element name="PartyId">  
 <complexType>  
 <simpleContent>  
 <extension base="ar:non-empty-string">  
 <attribute name="type" type="ar:non-empty-string"/>  
 </extension>  
 </simpleContent>  
 </complexType>  
</element>

<simpleType name="non-empty-string">  
 <restriction base="string">  
 <minLength value="1"/>  
 </restriction>  
</simpleType>

PartyId:llä ilmoitetaan järjestelmän tunniste, joka tässä tapauksessa tarkoittaa sanomaliikenteen osapuolta. Viitetietokantaympäristössä käytämme PartyId-elementissä OID-tunnusta, jossa solmutyyppi on 18 tai sen alisarja (esim. 18005). Esim. organisaation 1567535-0 tietyn järjestelmän sanomaliikenteen osapuolitunnus voisi olla 1.2.246.537.10.1567535.18.2004.12 (vuonna 2004 käyttöön otettu sanomaliikenteen järjestelmä 12). Vuosisarjan kohdalla voi käyttää myös arvoa 1, jos vuosisarjat eivät ole tarpeen.

Role-elementissä käytetään vakiorooleja, joita ovat:

|  |  |
| --- | --- |
| **Role** | **Tehtävä** |
| adapter | Perusjärjestelmäadapteri, joka lähettää viitteitä ja vastaa viitteen avulla tehtävään kyselyyn |
| refdb | Viitetietojärjestelmä |
| requester | Lähetteen lähettävä eli hoitoa tilaava osapuoli. Myös vastaanottaa hoitopalautteen. |
| provider | Lähetteen vastaanottava eli hoitoa tarjoava osapuoli. Myös lähettää hoitopalautteen. |
|  |  |

### TO (vastaanottaja)

<element name="To">  
 <complexType>  
 <sequence>  
 <element ref="ar:PartyId"/>  
 <element name="Role" type="ar:non-empty-string"/>  
 </sequence>  
 </complexType>  
</element>

Vastaanottajaan pätee samat määrittelyt kuin lähettäjään.

### CPAId (sovellusmäärittelyn tunnus)

<element name="CPAId" type="ar:non-empty-string"/>

Seuraavia sovellusmäärittelyitä on nykyään käytössä:

|  |  |
| --- | --- |
| **CPAId** |  |
| 1.2.246.777.11.2002.2 | Avoimet Rajapinnat -määritys |
| 1.2.246.777.11.2003.2 | Lähete- ja hoitopalaute-määritys |
| 1.2.246.777.11.2004.1 | OpenCDA 2005 määritys |
| 1.2.246.777.11.2005.17 | Hoitotakuun/tilastosanoman määritys |
|  |  |

### ConversationId (transaktion tunnus)

<element name="ConversationId" type="ar:non-empty-string"/>

Transaktion tunnus on sama lähete/palautesanomalle ja sen kuittaukselle. Lähettäjä saa täysin vapaasti määrätä kentän sisällön.

### MessageId:n ja ConversationId:n käytöstä

MessageId on yksikäsitteinen tunniste kullekin sanomalle. Sen perusteella sanoma löydetään lokitietokannasta tarvittaessa ja sitä tarvitaan myös kuittausmekanismissa. Jokainen sanoman lähettäjä generoi sanomalleen uniikin MessageId:n.  
  
ConversationId on "keskustelutunnus". Keskustelujahan on esim. viitetietokantaympäristössä 2 erilaista:

1) viiteaihion lähetys ja sen kuittaus

2) viitatun sisällön kysely ja viitatun sisällön palautus

ConversationId:n perusteella samaan transaktioon kuuluvat sanomat pystytään linkittämään toisiinsa. Viiteaihion lähettäjä generoi uuden ConversationId:n ja tämä arvo palautetaan viiteaihioiden kuittauksessa. Viitatun sisällön kysyjä generoi uuden ConversationId:n ja se palautetaan viitatun aineiston mukana.  
  
Esimerkki:  
  
Sanoma1, MessageId=X1, ConversationId=C1    (viiteaihiot)  
Sanoma2, MessageId=X2, ConversationId=C1, RefToMessageId= X1  (viiteaihioiden kuittaus)  
Sanoma3, MessageId=X3, ConversationId=C2    (viitatun sisällön kysely)  
Sanoma4, MessageId=X4, ConversationId=C2    (viitattu sisältö, varsinainen CDA-dokumentti).

Viiteaihioiden kuittaussanomassa palautetaan myös alkuperäisen sanoman MessageId elementissä RefToMessageId.

Kunkin tunnistenumeron pitää olla yksikäsitteinen oman kontekstinsa sisällä. Tämä vaatimus voidaan täyttää monella tavalla, mutta selkein tapa on 18-sarjan OID-koodien käyttö. 18-sarjan OID-koodeja voidaan käyttää sanomaliikenteen osapuolten tunnistamisen lisäksi myös muihin tietoliikenteen tunnistenumeroihin. Käytännön konfigurointityön ja hallinnan minimoimiseksi paras menettelytapa on käyttää keskustelun aloittavan sanoman tunnistenumeroa (MessageId) myös keskustelun tunnistenumerona (ConversationId).

Lisäksi MessageID kannattaa johtaa lähettävän järjestelmän sanomaliikenteen osapuolitunnisteesta (From PartyId) vuosisarjoittain:

%FromPartyId.vvvv.juoksevanumero,

esim. jos From PartyId on 1.2.246.537.10.1567535.18.2004.12,

vuoden 2005 10356 sanoman MessagId olisi

1.2.246.537.10.1567535.18.2004.12.2005.10356.

Seuraavassa esimerkissä yritys 0628061-3 lähettää viiteaihioita yritykselle 0784698-1 ja kyselyt tapahtuvat toiseen suuntaan. Kummassakin yrityksessä tietohallinto-osasto on päättänyt, että sanomaliikenteen osapuolitunnuksen solmutyyppi on 18 (tasan) ja sen alle kukin ohjelmisto saa generoida omia alisarjojaan. Ensimmäisessä yrityksessä ohjelmiston numero on 2004.225 ja toisessa 2004.6543. Nämä tunnistenumerot on määritelty ohjelmistojen konfigurointivaiheessa. Kummassakin ohjelmistossa on oma sisäinen vuosikohtainen sanomanumerogeneraattori, joka antaa numeroituvasti uuden kokonaisluvun sanomatunnisteille. Sarjat ovat vuosisarjoja, joten vuosiluvun vaihtuessa juokseva numerointi alkaa 1:stä ja vuosisarjaa osoittava numero luonnollisesti muuttuu. Kunkin keskustelun ConversationID on keskustelun aloittavan sanoman MessageId.

Sanoma1, viiteaihiot:

From, PartyId = 1.2.246.777.10.6280613.18.2004.225

To, PartyId = 1.2.246.777.10.7846981.18.2004.6543

MessageId = 1.2.246.777.10.6280613.18.2004.225.2004.21221

ConversationId = 1.2.246.777.10.6280613.18.2004.225.2004.21221

Sanoma2, viiteaihioiden kuittaus

MessageId = 1.2.246.777.10.7846981.18.2004.6543.2004.102   
ConversationId = 1.2.246.777.10.6280613.18.2004.225.2004.21221

RefToMessageId = 1.2.246.777.10.6280613.18.2004.225.2004.21221

Sanoma3, viitatun sisällön kysely

MessageId = 1.2.246.777.10.7846981.18.2004.6543.2004.103

ConversationId = 1.2.246.777.10.7846981.18.2004.6543.2004.103

Sanoma4, viitattu sisältö, varsinainen CDA-dokumentti

MessageId = 1.2.246.777.10.6280613.18.2004.225.2004.21222

ConversationId = 1.2.246.777.10.7846981.18.2004.6543.2004.103

OID-koodien käyttö MessageId- ja ConversationId-tunnistenumeroissa ei ole pakollista, vaikkakin suositeltavaa. Muutkin yksilöivät generointitavat kelpaavat.

### Service (palvelu)

<element name="Service">  
 <complexType>  
 <simpleContent>  
 <extension base="ar:non-empty-string">  
 <attribute name="type" type="ar:non-empty-string"/>  
 </extension>  
 </simpleContent>  
 </complexType>  
</element>

Elementissä ilmoitetaan palvelun tunnus.

### Action (tapahtuma)

<element name="Action" type="ar:non-empty-string"/>

Action määrittelee varsinaisen tapahtuman määritellyn palvelun (service) alla. Action osoittaa samalla tietovirran päätyypin. Service ja Action liittyvät läheisesti SOAP-Bodyn elementin ja nimiavaruuden nimiin, joilla ilmoitetaan sanomaa käsittelevä metodi ja luokka.

Käytössä olevat Service- ja Aciton -yhdistelmät on erotettu omaksi dokumentikseen ylläpidon yksinkertaistamiseksi.

### MessageData (sanoman tunnistetiedot)

<element name="MessageData">  
 <complexType>  
 <sequence>  
 <element ref="ar:MessageId"/>  
 <element ref="ar:Timestamp"/>  
 </sequence>  
 </complexType>  
</element>

<element name="MessageId" type="ar:non-empty-string"/>  
<element name="Timestamp" type="dateTime"/>

Käytämme tässä elementtejä MessageId ja Timestamp. MessageId on sanoman lähettäneen järjestelmän generoima uniikki sanomatunniste. Timestamp:n tietotyyppi on W3C:n scheman määrittelemä dateTime , joten aikaleima esitetään muodossa: yyyy-mm-ddThh:mm:ss (ISO8601).

## AckRequested

SOAP-headerimme toisella pääelementillä ilmaistaan sanomassa, että lähetyksestä halutaan sovellustason kuittaus.

|  |
| --- |
|  |
| Kuva 6: AckRequested |

Elementti esiintyy tyhjänä:

<ar:AckRequested SOAP-ENV:mustUnderstand="1"/>

## Acknowledgment

<element name="Acknowledgment">  
 <complexType>  
 <sequence>  
 <element ref="ar:Timestamp"/>  
 <element ref="ar:RefToMessageId"/>  
 <element ref="ar:From"/>  
 </sequence>  
 <attribute ref="SOAP-ENV:mustUnderstand"/>  
 </complexType>  
</element>

<element name="RefToMessageId" type="ar:non-empty-string"/>

|  |
| --- |
|  |
| Kuva 7: Acknowledgement |

Tämä elementti esiintyy pelkästään kuittaussanomassa. Kuittaus­headerissä(Acknowledgment) ilmoitetaan kuitattavan (tai hylättävän) sanoman tunnistenumero, kuittausaika (W3C:n dateTime muodossa) ja lähettäjän tiedot.

## HL7FIAcknowledgment

Kotoperäisesti määriteltyä HL7FIAcknowledgment-elementtiä käytetään pelkästään kuittaussanomassa. Sillä ilmoitetaan kuittauksen tyyppi.

<element name=”HL7FIAcknowledgment”>  
 <complexType>  
 <attribute name=”ackstatus” type=”ar:ackstatus.type” use=”required”/>  
 <attribute ref="SOAP-ENV:mustUnderstand"/>  
 </complexType>  
</element>

<simpleType name=”ackstatus.type”>  
 <restriction base=”NMTOKEN”>  
 <enumeration value=”ACK”>  
 <enumeration value=”NAK”>  
 </restriction>  
</simpleType>

|  |
| --- |
|  |
| Kuva 8: HL7FIAcknowledgement |

ACK-tyyppinen kuittaus lähetetään vastauksena onnistuneeseen tiedonsiirtoon. Virhetilanteissa lähetetään NAK-kuittaus. Sen yhteydessä palautetaan tieto palautuksen syistä, mahdolliset tarkemmat virheilmoitukset sekä mahdollisesti alkuperäisen sanoman sisältö.

## HL7FIBodyCount

<xs:element name="HL7FIBodyCount">

<xs:complexType>

<xs:simpleContent>

<xs:extension base="xs:integer">

<xs:attribute ref="SOAP-ENV:mustUnderstand"/>

</xs:extension>

</xs:simpleContent>

</xs:complexType>

</xs:element>

|  |
| --- |
|  |
| Kuva 9: HL7FIBodyCount |

Tämän elementin arvona on sanomaan sisältyvien XML-dokumenttien (esimerkiksi viiteaihioiden tai viitattujen sisältöjen) lukumäärä. Sen arvo on 1 silloin, kun sanoma sisältää yhden XML-dokumentin. Sen arvo voi olla myös suurempi kuin 1 esimerkiksi kun lähetetään nippu viiteaihioita tai vastataan useampia kuin 1 virheellinen viiteaihio.

## SOAP-Headerin elementtien pakollisuus

Kaikkien elementtien käyttämistä suositellaan, vaikka ne eivät olisikaan tarpeellisia juuri kyseisessä käyttötilanteessa aluksi. Myöhemmin voivat tarpeet muuttua, esimerkiksi jos tietoliikenteen reitittämiseen tulee palveluita, jotka tarvitsevat reititystietoja.

Ehdottoman pakollisia elementtejä ovat seuraavat:

* + - * To.PartyId
      * From.PartyId
      * MessageId
      * Timestamp

SOAP headerin schemaa esittävässä kaaviossa pakollisia elementtejä kuvaavissa laatikoissa on ehjä viiva ja ei-pakollisissa elementeissä katkoviiva.

### CPAId – sovellusmäärittelyn tunnus

On tärkeää, että sovellusmäärittelyn tunnus siirretään sanomassa mukana. Käytännössähän kyseinen tunnus on liikennöinnin määrittelevän sovellusoppaan tunnus. Ilman tätä tietoa ei tiedetä, mitä XML-rakenteita pitää osata purkaa vastaanotettavasta sanomasta. Tämä elementti määriteltäneen tulevaisuudessa pakolliseksi.

### Service – palvelu ja action – tapahtuma

Myös näiden tietojen käyttö on tarpeen, sillä näiden elementtien perusteella vastaanottopäässä kukin sanoma ohjataan oikealle käsittelyohjelmalle. Tämä elementti määriteltäneen tulevaisuudessa pakolliseksi.

## Palvelukehys

Varsinainen siirrettävä sisältö sijoitetaan SOAP-ENV:Body elementin sisällä olevaan palvelukehys-elementtiin. SOAP-liikenteessä tämän elementin nimiavaruus kertoo luokan ja elementin nimi kertoo metodin, jolle sisältö ohjataan käsiteltäväksi. Esimerkki:

<SOAP-ENV:Body>

<arb:R2HeadersBody xmlns:arb="urn::refdb-body">

...

siirrettävä sisältö

....

</arb:R2HeadersBody>

</SOAP-ENV:Body>

Kyseisessä esimerkissä sisältö ohjataan luokalle refdb-body ja metodille R2HeadersBody.

Palvelukehykselle ei ole erikseen omaa schemaa. Siirrettävällä sisällöllä voi olla oma schema. Se tulee esitellä siirrettävän XML-dokumentin alussa normaalilla tavalla. Esimerkki CDA dokumentin siirrosta:

<SOAP-ENV:Body>

<arb:R2CDABody xmlns:arb="urn::refdb-body">

<ClinicalDocument  
 xmlns="urn:hl7-org:v3" xmlns:voc="urn:hl7-org:v3/voc"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="urn:hl7-org:v3  
 CDA.ReleaseTwo.CommitteeBallot03.Aug.2004.xsd">

........

CDA dokumentti

........

</ClinicalDocument>

</arb:R2CDABody>

</SOAP-ENV:Body>

Mekanismi on sama CDA R1 ja CDA R2 dokumenteille. Sisältö voi olla muukin XML-dokumentti kuin CDA-dokumentti..

Vastaanottava sovellus voi halutessaan validoida siirretyn XML-dokumentin käyttäen sen alussa esiteltyä scheman tiedoston nimeä. Tällöin vastaanottavassa sovelluksessa on huolehdittava siitä, että schema on saatavilla saman nimisenä tiedostona kuin se on dokumentissa esitelty.

# Vuorovaikutustilanteet ja sanomat

## Yleisperiaatteet

Kuten aikaisemmin todettiin, kahden sovelluksen välisellä vuorovaikutuksella voi olla kaksi käyttötarkoitusta:

* + - * Aineiston lähetys ja vastaanoton kuittaus
      * Aineiston pyyntö ja pyydetyn aineiston palautus

Lähetettävä aineisto voi muodostua yhdestä dokumentista, esimerkiksi lähetteestä, tai se voi sisältää nipun dokumentteja, esimerkiksi nipun viiteaihioita. Vuorovaikutustilanteessa tarvitaan erilaisia sanomia, joilla hallitaan myös varmistuminen perillemenosta sekä virhetilanteiden käsittely.

Tarkoitus on, että erityyppisten sisältöjen siirrossa käytetään samoja periaatteita ja mekanismeja. Avoimet rajapinnat -määritykset laadittiin ensin viitetietojärjestelmää varten. Seuraavaksi rajapintamääritysten käyttöä laajennettiin lähetteiden ja hoitopalautteiden siirtämiseen. OpenCDA -jatkohankkeessa näitä määrityksiä yleistetään kattamaan sovellusten välinen vuorovaikutus yleisellä tasolla.

Näihin vuorovaikutustilanteisiin liittyviä yksityiskohtia kuvataan seuraavassa kappalessa. Sanomat ovat seuraavat:

* + - * Aineiston lähetys
      * Kuittaus hyväksytyn aineiston vastaanotosta
      * Kuittaus virheellisen aineiston vastaanotosta
      * Aineiston pyyntö
      * Vastaus pyyntöön
      * Tiedonsiirtotason kuittaus

## Aineiston lähetys

XML-dokumentit siirretään siten, että SOAP body-osuudessa on yksi tai useampia XML-dokumentteja peräkkäin. XML-dokumenttien lukumäärä kerrotaan SOAP-headerin elementillä *HL7FIBodyCount*. Metodi ja luokka ilmaisevat luokan ja metodin, jolle XML-dokumentit on tarkoitettu. SOAP-Headerin Service ja Action ilmoittavat vastaavan palvelun ja tapahtuman.

<SOAP-ENV:Body>

<arb:metodi xmlns:arb="urn::luokka">

........

<XML-dokumentti>

........

</arb:metodi>

</SOAP-ENV:Body>

## Aineiston vastaanoton kuittaukset

### Kuittaussanomat

Aineiston vastaanoton kuittaaminen on pakollista. Kuittausanomissa *MessageHeaderin Action* arvo on **ACK**. Kuittattava sanoma ilmoitetaan SOAP-headerin elementillä *Acknowledgement,* joka siis esiintyy pelkästään kuittaussanomassa.

Kuittauksen tyyppi ilmoitetaan headerin *HL7FIAcknowledgment* attribuutilla ackstatus, jonka sisältö on seuraava:

* + - * Koko lähetyksen kuittaus, ackstatus=”ACK”
      * Koko lähetyksen hylkäys, ackstatus=”NAK”
      * Joiden XML-dokumenttien hylkäys, ackstatus=”SOME”

ACK-kuittauksen yhteydessä SOAP bodyssa on tyhjä elementti AckBody.

NAK tai SOME-kuittauksen yhteydessä bodyn pääelementti on NAKBody.

Jos kuittauskoodi on ”SOME”, ilmoitetaan SOAP-Bodyssä virheellisen XML-dokumentin id, virhekoodi ja virheen selitys (ja mahdollinen lisäinformaatio eli virheen analyysi) elementillä HL7FIError. Virheellinen XML-dokumentti ”palau­tetaan” heti tämän virhetiedon perässä. Rakenne on toistuvan HL7FIErrSeq elementin sisällä. Toistumia on yhtä monta kuin virheellisiä XML-dokumentteja alkuperäi­sessä lähetyksessä. Lukumäärä ilmoitetaan SOAP-headerin elementillä HL7FIBodyCount.

Jos kuittauskoodi on NAK, on kyseessä koko lähetyksen hylkäys. Tällöin SOAP-bodyssä esiintyy elementti HL7FIErrSeq ja sen sisällä HL7FIError yhden kerran (ilman XML-dokumentteja). Elementin HL7FIError-sisällä oleva HeaderID jätetään myös tyhjäksi (<HeaderId/>). Kaikkia virheellisiä XML-dokumentteja ei siis palauteta. Käytännössä koko lähetyksen hylkäys tehdään, kun ennalta määrätty prosenttiosuus XML-dokumenteista on virheellisiä tai koko lähetys on virheellinen, esim. SOAP-headerissä on vikaa.

<xs:complexType name="AckBody">  
 <xs:sequence/>  
</xs:complexType>

<element name=”NAKBody”>  
 <xs:sequence>  
 <element ref="ar:HL7FIErrSeq" maxOccurs="unbounded"/>  
 <xs:sequence>  
</element>

<element name=”HL7FIErrSeq”>  
 <complexType>  
 <sequence>  
 <element ref=”ar:HL7FIError”/>  
 <any namespace="##other" processContents="lax" minOccurs="0"   
 maxOccurs="unbounded"/>  
 </sequence>  
 </complexType>  
</element>

<element name=”HL7FIError”>  
 <complexType>  
 <sequence>  
 <element ref="ar:HeaderId"/>  
 <element name=”errcode” type=”string”/>  
 <element name=”errTxt” type=”string”/>  
 <element name=”errInfo” type=”string”/>  
 </sequence>  
 </complexType>  
</element>

Alla schemamäärittelyssä HeaderId-elementti vastaa CDA-headerin id element­tiä.

<element name=”HeaderId”>  
 <complexType>  
 <attribute name=”EX” type=”string”>  
 <attribute name=”RT” type=”string”>  
 </complexType>  
</element>

### Onnistuneen vastaanoton kuittaus

Esimerkki kuittaussanomasta, ei virheitä.

<?xml version="1.0" encoding="utf-8"?>

<SOAP-ENV:Envelope xmlns:ar="urn::refdb" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn::refdb hl7fi\_envelope\_V0.1.xsd">

<SOAP-ENV:Header>

<ar:MessageHeader SOAP-ENV:mustUnderstand="1">

<ar:From>

<ar:PartyId>1.2.246.10..vtj.palvelu.18.2002.1</ar:PartyId>

<ar:Role>refdb</ar:Role>

</ar:From>

<ar:To>

<ar:PartyId>1.2.246.10..läh.organisaatio.18.2002.1</ar:PartyId>

<ar:Role>adapter</ar:Role>

</ar:To>

<ar:CPAId>1.2.246.777.11.2004.1</ar:CPAId>

<ar:ConversationId>1</ar:ConversationId>

<ar:Service>refdb</ar:Service>

<ar:Action>R2ACK</ar:Action>

<ar:MessageData>

<ar:MessageId>2</ar:MessageId>

<ar:Timestamp>2003-01-07T08:23:00</ar:Timestamp>

</ar:MessageData>

</ar:MessageHeader>

<ar:Acknowledgment SOAP-ENV:mustUnderstand="1">

<ar:Timestamp>2003-01-07T08:22:00</ar:Timestamp>

<ar:RefToMessageId>1</ar:RefToMessageId>

<ar:From>

<ar:PartyId>1.2.246.10..läh.organisaatio.18.2002.1</ar:PartyId>

<ar:Role>adapter</ar:Role>

</ar:From>

</ar:Acknowledgment>

<ar:HL7FIAcknowledgment SOAP-ENV:mustUnderstand="1" ackstatus="ACK"/>

</SOAP-ENV:Header>

<SOAP-ENV:Body>

<arb:R2AckBody xmlns:arb="urn::refdb-body"/>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

Kuittaus (positiivinen) lähetetään siinä vaiheessa, kun xml-dokumentin on vastaanotettu ja käsitelty. Asynkronisen siirron (http) tietoliikennetason vastaanottokuittaus (negatiivinen tai positiivinen) lähetetään sen sijaan heti.Tietoliikennekäytäntöihin liittyy myös omat alemman tason kuittauksensa, joiden käyttö ja määrittely ei kuulu tähän projektiin. Tällä tarkoitetaan esim. http:n normaaleja virhekoodeja http-kehyksessä.

### Virheilmoitukset ja -koodit

Virhekoodeille ja selityksille käytetään seuraavia arvoja:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Virhe-koodi (errcode) | Virheteksti (errTxt) | Virheen selitys | Käyttö | ackcode |
| 100 | Järjestelmän sisäinen virhe | Vastaanottava järjestelmä ei pystynyt käsittelemään vastaanotettua aineistoa Jos ongelma toistuu, varmista, että järjestelmän yhteyshenkilö saa tiedon ongelmasta.  Tulkinta: pyydetään uudelleenlähetystä | viiteaihiot,  viitatun aineiston kysely,  muut xml-dokumentit | NAK |
| 200 | Viallinen sanoma | Sanoma on virheellinen, eikä sitä pystytty purkamaan. Ilmoitus yhteyshenkilölle. | viiteaihiot,  muut xml-dokumentit | NAK |
| 201 | Lähetys on jo käsitelty | Kyseessä on hukkunut kuittaus tai timeoutin aiheuttama uudelleenlähetys ja nyt uudelleenlähetyksen jälkeen adapteri saa lähetyksen kuitattua. | viiteaihiot,  muut xml-dokumentit | NAK |
| 300 | Viallinen xml-dokumentti | Yksittäinen xml-dokumentti oli virheellinen eikä sitä pystytty purkamaan. Ilmoitus yhteyshenkilölle. | viiteaihiot, muut xml-dokumentit | SOME, NAK |
| 301 | XML-dokumentti on jo käsitelty | Yksittäinen xml-dokumentti on jo käsitelty. | viiteaihiot, muut xml-dokumentit | SOME, NAK |
| 302 | xml-dokumentissa tuntematon koodisto. | Tietty koodisto oli virheellinen (koodiston nimi jos mahdollista elementtiin errInfo). Ilmoitus yhteyshenkilölle. | viiteaihiot, muut xml-dokumentit | SOME, NAK |
| 400 | Virheellinen kyselysanoma | Kyselysanoman purku ei onnistunut kunnolla | viitatun sisällön kysely, muun xml-dokumentin pyyntö | NAK |
| 401 | Pyydettyä aineisto ei ole saatavissa | Hukkunut? | viitatun sisällön kysely, muun xml-dokumentin pyyntö | NAK |
| 500- | toimittaja-kohtainen | Toimittajakohtaiset virhekoodit |  |  |

Virhekoodien jaottelu:

|  |  |
| --- | --- |
| ”100”-sarja | selitys |
| 100-199 | Yleiset virheet |
| 200-299 | xml-dokumenttien koko sanomaan vaikuttavat virheet |
| 300-399 | xml-dokumenttien virheet lähetyssanomassa |
| 400-499 | Pyydetyn aineiston kyselysanomaan liittyvät virheet |
| 500- | Toimittajakohtaiset |

Esimerkki virhesanomasta, yksi virheellinen viiteaihio:

<?xml version="1.0" encoding="utf-8"?>

<SOAP-ENV:Envelope xmlns:ar="urn::refdb" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn::refdb hl7fi\_envelope\_V0.1.xsd">

<SOAP-ENV:Header>

<ar:MessageHeader SOAP-ENV:mustUnderstand="1">

<ar:From>

<ar:PartyId>1.2.246.10..vtj.palvelu.18.2002.1</ar:PartyId>

<ar:Role>refdb</ar:Role>

</ar:From>

<ar:To>

<ar:PartyId>1.2.246.10..läh.organisaatio.18.2002.1</ar:PartyId>

<ar:Role>adapteri</ar:Role>

</ar:To>

<ar:CPAId>1.2.246.777.11.2004.1</ar:CPAId>

<ar:ConversationId>1</ar:ConversationId>

<ar:Service>refdb</ar:Service>

<ar:Action>R2ACK</ar:Action>

<ar:MessageData>

<ar:MessageId>1</ar:MessageId>

<ar:Timestamp>2003-01-07T08:23:30</ar:Timestamp>

</ar:MessageData>

</ar:MessageHeader>

<ar:Acknowledgment SOAP-ENV:mustUnderstand="1">

<ar:Timestamp>2003-01-07T08:23:00</ar:Timestamp>

<ar:RefToMessageId>1</ar:RefToMessageId>

<ar:From>

<ar:PartyId>1.2.246.10..läh.organisaatio.18.2002.1</ar:PartyId>

<ar:Role>adapteri</ar:Role>

</ar:From>

</ar:Acknowledgment>

<ar:HL7FIAcknowledgment SOAP-ENV:mustUnderstand="1" ackstatus="SOME"/>

<ar:HL7FIBodyCount SOAP-ENV:mustUnderstand="1">1</ar:HL7FIBodyCount>

</SOAP-ENV:Header>

<SOAP-ENV:Body>

<arb:R2NAKBody xmlns:arb="urn::refdb-body">

<arb:HL7FIErrSeq>

<arb:HL7FIError>

<arb:HeaderId EX="2002.1230"  
 RT="1.2.246.576.10.2013120.1.8395.11"/>

<arb:errcode>3</arb:errcode>

<arb:errTxt>Viallinen CDA-header</arb:errTxt>

<arb:errInfo>Pakollinen elementti patient\_encounter   
 puuttuu</arb:errInfo>

</arb:HL7FIError>

<ClinicalDocument   
 xmlns="urn:hl7-org:v3" xmlns:voc="urn:hl7-org:v3/voc"   
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"   
 xsi:schemaLocation="urn:hl7-org:v3   
 CDA.ReleaseTwo.CommitteeBallot03.Aug.2004\_Header.xsd"   
 templateId="2.16.840.1.113883.3.27.1776">

<!--

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CDA Header

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-->

<id extension="c266" root="2.16.840.1.113883.3.933"/>

......

### Kuittaus- ja toipumismenettelyt

Xml-dokumenttien lähetyspäässä eli adapterissa seurataan ylläpitäjän toimesta mahdol­lisia virheilmoituksia ja ryhdytään tarvittaviin korjaustoimiin. Tilanteen korjaamista helpottaa, jos virheen syy tekstinä on riittävän kuvaileva ja informatiivinen, joten tässä tapauksessa ylläpidon helppouteen vaikuttaa myös vastaanottavan adapteritoimittajan kuittauksien implementointitapa.

Negatiivisen kuittauksen saaneet xml-dokumentit lähetetään automaattisesti yhden kerran uudestaan, jos kuittausanomassa virhekoodi on 100 (siis sovellustason kuittauksessa, vastaanottotasolla vastaava koodi on 1). Muissa tapauk­sissa virheet selvitetään adapterin ylläpitäjän toimesta. Jos vastausta (sovellusta­son kuittausta) ei tule kahdenkeskisesti sovitun timeoutin kuluessa, aloitetaan kahdenkeskisesti sovittu toipumismenettely. Default-tulkinta timeoutille on, että yritetään yhden kerran uudestaan.

Xml-dokumentteja voi lähettää, vaikka edellisiä ei ole vielä kuitattu. Ajastetun siirron tapauksessa edellisten siirtokertojen kuittaamatto­mat xml-dokumentit lähetetään uudestaan (lopulta, 3 kerran jälkeen, pitää selvittää, miksei kuittauksia tule). Jos yhteys on kokonaan poikki, niin lähetysyrityksiin ryhdytään vasta, kun yhteys on taas päällä. Jos xml-dokumenttien vastaanoton kuittaus ei mene perille, yritetään sitäkin timeoutin jälkeen yhden kerran uudestaan.

Xml-dokumenttien lähettäjä on vastuussa xml-dokumenttien siirrosta, kunnes sovellustason positiivisella kuittauksella ilmoitetaan xml.dokumenttien onnistuneesta käsittelystä. Vastaanottavassa päässä ei tarvitse esim. välittää tilanteesta, jossa havaitaan sano­matunnisteen perusteella sanomia puuttuvan välistä.

## Yksilöidyn xml-dokumentin pyyntö

XML-dokumentti pyydetään siten, että SOAP body-osuudessa on yksi tai useampia XML-dokumentteja yksilöiviä tunnuksia. Pyynnön elementin nimi ja nimiavaruus ilmaisevat metodin ja luokan, jolle XML-dokumenttien pyyntö on tarkoitettu. SOAP-Headerin Service ja Action ilmoittavat vastaavan palvelun ja tapahtuman.

<SOAP-ENV:Body>

<arb:metodi xmlns:arb="urn::luokka">

........

<XML-dokumentin yksilöivä tunnus>

........

</arb:metodi>

</SOAP-ENV:Body>

Oheisessa esimerkissä on CDA R2 dokumentin pyyntö perusjärjestelmäadapterilta.

Avoimet adapterit CDA-rajapintamäärittelyn tasolla 1.1 kysely suoritetaan doku­mentin set\_id-elementin perusteella. SOAP Bodyn pääelementtinä on R2QUERY\_ID, jossa on pelkästään haluttavan aineiston set\_id ja attribuutti Give\_Structured\_Format, jonka avulla kerrotaan , halutaanko viitatusta aineis­tosta rakenteinen (1) vai näyttömuoto (0).

Esimerkki:

<?xml version="1.0" encoding="utf-8"?>

<SOAP-ENV:Envelope xmlns:ar="urn::refdb" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn::refdb hl7fi\_envelope\_V0.1.xsd">

<SOAP-ENV:Header>

<ar:MessageHeader SOAP-ENV:mustUnderstand="1">

<ar:From>

<ar:PartyId>1.2.246.10..vtj.palvelu.18.2002.1</ar:PartyId>

<ar:Role>refdb</ar:Role>

</ar:From>

<ar:To>

<ar:PartyId>1.2.246.10..läh.organisaatio.18.2002.1</ar:PartyId>

<ar:Role>adapter</ar:Role>

</ar:To>

<ar:CPAId>1.2.246.777.11.2004.1</ar:CPAId>

<ar:ConversationId>1</ar:ConversationId>

<ar:Service>refdb</ar:Service>

<ar:Action>R2QUERY</ar:Action>

<ar:MessageData>

<ar:MessageId>1</ar:MessageId>

<ar:Timestamp>2003-01-07T08:23:00</ar:Timestamp>

</ar:MessageData>

</ar:MessageHeader>

</SOAP-ENV:Header>

<SOAP-ENV:Body>

<arb:R2QUERY\_ID xmlns:arb="urn::refdb-body" Give\_Structured\_Format="0">

<arb:setId extension="BB35" root="2.16.840.1.113883.3.933"/>

</arb:R2QUERY\_ID>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

Kyselyn vastauksena on xml-dokumentti loppukäyttäjälle ja dokumentti toimii tavallaan kyselyn kuittauksena.

Jos kyselyä ei pystytä käsittelemään, esim. dokumenttia ei pystytä sillä kertaa generoimaan, palautetaan kyselevään järjestelmään samanlainen kuittaussanoma kuin viiteaihioiden kuittauksessa. Kuittauskoodi on tällöin NAK ja virhekoodi välillä 400-499.

## Xml-dokumenttien pyyntö lomaketunnuksella

Kunnollisessa arkkitehtuurissa henkilötietoja ylläpidetään tietyssä master-sovelluksessa ja useat (lähes kaikki) sovellukset hyödyntävät näitä tietoja. Tämän vuoksi on syytä määritellä myös dokumenttien siirtoympäristöön helppo tapa, jolla tietyn henkilön henkilötietolomake saadaan nähtäväksi (ja liitettäväksi käsiteltävään sairauskertomustietoon). Tarvitaan siis mekanismi, jolla voidaan pyytää tietyn henkilön tietyn lomakkeen uusin versio. Tällaisia lomakkeita ovat henkilötietolomakkeen lisäksi mm. lääkityslista, diagnoosilista ja viimeisimmät laboratoriotulokset.

Tätä tarvetta varten voidaan toteuttaa kysely R2LOMQUERY, jossa kyselyn parametreina annetaan henkilön tunniste (OID id) ja pyydettävien lomakkeiden tunnisteet (OID code):

<xs:element name="R2LOMQUERY">

<xs:complexType>

<xs:sequence>

<xs:element name="patient\_id" type="v3dt:II"/>

<xs:element name="lom\_code" type="v3dt:CD" minOccurs="1"   
 maxOc curs="unbounded"/>

</xs:sequence>

</xs:complexType>

</xs:element>

Potilaan tunnus esitetään samalla tavalla kuin CDA Headerisssa.

<patient>

<id extension="241270-123B" root="1.2.246.573.21.1"/>

</patient>

Lomakkeiden tunnukset esitetään samalla tavalla kuin CDA Headerin localHeader elementissä:

<hl7fi:lomakkeet>

<hl7fi:lomake>

<hl7fi:lomake\_code code="3" codeSystem="1.2.246.537.6.12.2002"

codeSystemName="Lomake" displayName="HEN"/>

</hl7fi:lomake>

<hl7fi:lomake>

<hl7fi:lomake\_code code="20" codeSystem="1.2.246.537.6.12.2002"

codeSystemName="Lomake" displayName="SIS"/>

</hl7fi:lomake>

</hl7fi:lomakkeet>

Toiminnallisuutta toteutettaessa on otettava huomioon mm:

* + - * Palautetaanko vastauksena tulevat lomakkeet yhtenä CDA dokumenttina, joka sisältää kaikki lomakkeet vai useampana CDA dokumenttina, yksi jokaista lomaketta kohti
      * Organisaatioiden välillä tehtäviä kyselysanomia on täydennettävä luovutuspyyntö- ja suostumuslomakkeilla sekä vastauksia vastaavilla luovutuksen saatedokumenteilla

## Xml-dokumentti vastauksena kyselyyn

Vastauksen elementin nimi ja nimiavaruus ilmaisevat metodin ja luokan, jolle vastauksena tuleva XML-dokumentti on tarkoitettu. SOAP-Headerin Service ja Action ilmoittavat vastaavan palvelun ja tapahtuman.

<SOAP-ENV:Body>

<arb:metodi xmlns:arb="urn::luokka">

........

<XML-dokumentti>

........

</arb:metodi>

</SOAP-ENV:Body>

Viitetietokantaympäristössä SOAP-body:ssä on yksi CDA-dokumentti (joka voi sisältää useita lomakkeita), mutta yleisessä tapauksessa XML-dokumentteja voi olla useita. Dokumenttien luku­määrä ilmoitetaan SOAP-headerin elementillä HL7FI*BodyCount*. Xml-doku­mentin sisältävä vastauspaketti toimii kyselyn kuittauksena.

<?xml version="1.0" encoding="utf-8"?>

<SOAP-ENV:Envelope xmlns:ar="urn::refdb" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn::refdb

hl7fi\_envelope\_V0.1.xsd">

<SOAP-ENV:Header>

<ar:MessageHeader SOAP-ENV:mustUnderstand="1">

<ar:From>

<ar:PartyId>1.2.246.10..läh.organisaatio.18.2002.1</ar:PartyId>

<ar:Role>adapter</ar:Role>

</ar:From>

<ar:To>

<ar:PartyId>1.2.246.10..vtj.palvelu.18.2002.1</ar:PartyId>

<ar:Role>refdb</ar:Role>

</ar:To>

<ar:CPAId>1.2.246.777.11.2004.1</ar:CPAId>

<ar:ConversationId>2</ar:ConversationId>

<ar:Service>refdb</ar:Service>

<ar:Action>R2CDA</ar:Action>

<ar:MessageData>

<ar:MessageId>1</ar:MessageId>

<ar:Timestamp>2003-01-07T09:30:00</ar:Timestamp>

</ar:MessageData>

</ar:MessageHeader>

<ar:HL7FIBodyCount SOAP-ENV:mustUnderstand="1">1</ar:HL7FIBodyCount>

</SOAP-ENV:Header>

<SOAP-ENV:Body>

<arb:R2CDABody xmlns:arb="urn::refdb-body">

<ClinicalDocument   
 xmlns="urn:hl7-org:v3"   
 xmlns:voc="urn:hl7-org:v3/voc"   
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"   
 xsi:schemaLocation="urn:hl7-org:v3  
 CDA.ReleaseTwo.CommitteeBallot03.Aug.2004.xsd"   
 templateId="2.16.840.1.113883.3.27.1776">

<id extension="c266" root="2.16.840.1.113883.3.933"/>  
 ......

## Tiedonsiirtoyhteyden toimivuuden testaussanoma

Adaptereita käyttöönotettaessa on hyödyllistä, jos tiedonsiirtoyhteyden toimivuus voidaan todeta vaikka hyötykuormaa ei vielä olisikaan lähetettävänä. Tätä varten ehdotetaan testaussanomaa, johon vastaanottava adapteri lähettää HL7FICommit-sanoman (katso seuraavasta luvusta). Testaussanoman rakenne on seuraava:

<?xml version="1.0" encoding="utf-8"?>

<SOAP-ENV:Envelope xmlns:ar="urn::refdb" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn::refdb

hl7fi\_envelope\_V0.1.xsd">

<SOAP-ENV:Header>

<ar:MessageHeader SOAP-ENV:mustUnderstand="1">

<ar:From>

<ar:PartyId>1.2.246.10..läh.organisaatio.18.2002.1</ar:PartyId>

<ar:Role>refdb</ar:Role>

</ar:From>

<ar:To>

<ar:PartyId>1.2.246.10..vtj.palvelu.18.2002.1</ar:PartyId>

<ar:Role>adapter</ar:Role>

</ar:To>

<ar:CPAId>1.2.246.777.11.2004.1</ar:CPAId>

<ar:ConversationId>1</ar:ConversationId>

<ar:Service>refdb</ar:Service>

<ar:Action>R2RRQ</ar:Action>

<ar:MessageData>

<ar:MessageId>1</ar:MessageId>

<ar:Timestamp>2005-01-07T08:23:00</ar:Timestamp>

</ar:MessageData>

</ar:MessageHeader>

</SOAP-ENV:Header>

<SOAP-ENV:Body/>

</SOAP-ENV:Envelope>

Testaussanomaa käytettäessä Action on R2RRQ. Varsinaista hyötykuormaa ei testaussanomassa tarvita, joten SOAP-ENV:Body on tyhjä elementti tai sisältää metodin ja luokan määrittelevän kehyselementin..

# Siirtomenetelmät

## Siirron luonne

Kuten aikaisemmin todettiin, kahden sovelluksen välisellä vuorovaikutuksella voi olla kaksi käyttötarkoitusta:

* + - * Aineiston lähetys ja vastaanoton kuittaus
      * Aineiston pyyntö ja pyydetyn aineiston palautus

Vuorovaikutustilanteeseen liittyy erilaisia ominaisuuksia, mm. seuraavia:

* + - * Siirto voi olla eräsiirtoa tai yksittäinen xml-dokumentti kerrallaan tapahtuvaa
      * Siirto voi olla synkronista tai asynkronista
      * Siirto voi tapahtua taustalla sovellusten välillä tai reaaliajassa siten, että käyttäjä odottaa siirron valmistumista

Käytännössä siirtoon käytetään http(s):ää. Avoimet Rajapinnat -määrityksissä toisena vaihtoehtona oli ftp(s):n käyttö, mutta sitä ei vielä ole missään toteutettu.

Aineiston pyyntö ja pyydetyn aineiston palautus on toistaiseksi toteutettu synkronisena eli pyyntöön vastataan välittömästi.

Aineiston lähetys ja vastaanoton kuittaus on toteutettu sekä synkronisena että asynkronisena. Synkronisessa siirrossa lähettävä järjestelmä odottaa sovellustason kuittausta jokaisen lähetyksen jälkeen ennen uuden lähetyksen tekemistä. Asynkronisessa siirrossa lähettävä järjestelmä voi lähettää useita sanomia odottamatta sanomaan tulevaa sovellustason kuittausta. Kuitenkin asynkronisessakin siirrossa kunkin sanoman kuittaus on vastaanotettava ja käsiteltävä sekä tarvittaessa ryhdyttävä virheenselvitysmenettelyyn. Asynkronisen siirron etuna synkroniseen siirtoon verrattuna on korkeampi tietoliikenneverkon käyttöaste ja siis aikayksikössä suurempi siirretty määrä. Varjopuolena on poikkeus- ja häiriötilanteiden selvittämisen vaativuus.

## http(s)

Tiedonsiirto tapahtuu käyttämällä POST-metodia. Http sopii erityisesti perusjär­jestelmäkyselyn ja sen vastauksen toteuttamiseen. Tällöin vastaussanoma saadaan POST:in paluuarvon mukana. Http:llä saadaan aikaan synkroninen siirto koko transaktiota ajatellen. Sanomat siirretään http:n avulla siten, että Content-Type on text/xml.

Esimerkki:

POST /AARliikenne HTTP/1.1  
Host: palvelut.sairaala.fi  
Content-Type: text/xml; charset=”utf-8”  
Content-Length: nnnn  
....  
  
……tähän määrittelemämme SOAP-sanoma

Esimerkin Host-muuttujan sisältö on tunniste, jonka on sovittu etukäteen vastaanottajan kanssa

ja jolla vastaanottaja tietää, mistä teknisestä osoitteesta esim. viitteen sisältöä voidaan kysyä.

Tällä on merkitystä kahdessa mielessä: järjestelmien url:t ja host-muuttujat saattavat vaihtua päivi-tysten myötä.

Lisäksi: samassa host:ssa voi olla useita adaptereita. Ne tulee siksi erottaa jollakin muulla tavalla kuin host-muuttujalla.

## Siirtotason kuittaukset

XML-dokumentin pyyntö on luonteeltaan synkroninen tapahtuma, jolloin vastausta odotetaan heti.

Xml-dokumenttien lähetys voi olla synkronisen lisäksi myös asynk­ronista, jolloin varsinainen sovellustason kuittaussanoma lähetetään myö­hemmin. Http-kehyksessä on oma virhekoodinsa, mutta OK-koodikin kertoo vain siirron onnistumisesta, ei varsinaisen vastaanoton onnistumisesta saatikka talletuksesta välivarastoon.

Tarvitsemme siis vastaanottotason kuittausmäärittelyn HL7FICommit edellämaini­tuille synkronisille tapahtumille, sekä onnistunutta että epäonnistunutta vastaanottoa varten.

Oma sovellustason kuittauksesta poikkeava (yksinkertainen ) sanoma HL7FICommit tarvitaan, koska vastaanottovaiheessa sanomaa ei ole vielä purettu ja sovellustason kuittauksessa määritellyt tiedot eivät ole saatavilla.

Epäonnistunutta vastaanottosanomaa käytetään myös synkronisen xml-dokumenttien lähetyksen ”vastaanoton” epäonnistuessa sekä kyselysanoman ”vastaanoton” epäon­nistuessa.

Seuraavassa taulukossa havainnollistetaan vastaanottokuittauksia.

|  |  |  |
| --- | --- | --- |
| Tietovirta | Vastaanotto OK | Vastaanotto ei onnistunut |
| xml-dokumenttien synkroninen siirto | Sovellustasolta tulee kuittaus (joka tosin voi olla myös negatiivinen). Vastaanottokuittausta ei tarvita. | Vastaanottotasolta lähetetään negatiivinen vastaanottokuittaus HL7FICommit, errcode>0 |
| xml-dokumenttien asynkroninen siirto | Sovellustason kuittaus tulee vasta myöhemmin, joten vastaanottotasolta lähetetään positiivinen vastaanottokuittaus HL7FICommit, errcode=0 | Vastaanottotasolta lähetetään negatiivinen vastaanottokuittaus HL7FICommit, errcode>0 |
| xml-dokumenttien kuittaukset asynkronisessa siirrossa | Vastaanottotasolta lähetetään positiivinen vastaanottokuittaus HL7FICommit, errcode=0 | Vastaanottotasolta lähetetään negatiivinen vastaanottokuittaus HL7FICommit, errcode>0 |
| xml-dokumenttien pyyntö (aina synkroninen) | Sovellustasolta tulee vastauksena pyydetty aineisto | Vastaanottotasolta lähetetään negatiivinen vastaanottokuittaus HL7FICommit, errcode>0 |

Edellämainitun taulukon kaikissa tapauksissa täytyy määritellä (yhteyskohtaisesti) **timeout**, jonka on oltava parametroitavissa.

Vastaanottotason kuittaus liittyy aina synkroniseen transaktioon, joten lähettäjä tietää, mistä kuittauksesta on kyse. Kuittaussanomassa ei siis tarvita lähetyksen tunnusta.

Kuittaussanomassa ei ole SOAP-header elementtiä. Implementaatiokohtaisesti se voidaan määritellää kuuluvaksi mukaan sanomaan, jos kuitattava sanoma halutaan yksilöidä messageId:n avulla.  
HL7FICommit:in bodyn rakenne on seuraava:

<xs:element name="CommitBody">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element ref=”errcode”/>  
 <xs:element ref=”errTxt”/>  
 <xs:element ref="errInfo" minOccurs=”0”/>  
 </xs:sequence>  
 </xs:complexType>  
</xs:element>

<xs:element name="errcode" type="xs:string"/>  
<xs:element name="errTxt" type="xs:string"/>  
<xs:element name="errInfo" type="xs:string"/>

Alla esimerkki OK-kuittaussanomasta (esim. kuitataan asynkroninen xml-dokumenttien lähetys vastaanottokuittauksella):

<SOAP-ENV:Envelope xmlns:SOAP-ENV="<http://schemas.xmlsoap.org/soap/envelope/>"   
xmlns:ar="<http://www.hl7.fi/ar2002/refdb>">   
 <SOAP-ENV:Body>  
 <ar:CommitBody>  
 <ar:errcode>0</ar:errcode>  
 <ar:errTxt>Vastaanotto OK</ar:errTxt>  
 <ar:CommitBody/>  
 </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>

|  |  |  |
| --- | --- | --- |
| Virhekoodi (errcode) | Virheteksti (errTxt) | Virheen selitys |
| 0 | Vastaanotto OK |  |
| 1 | Resurssinvarausongelma | Pyydetään uudelleenlähetystä sovitun parametrin retry\_after mukaisen ajan jälkeen. |
| 2 | Sanoma ei ole well formed | XML-rakenne rikki |
| 3 | Sanoma ei ole validi | Ei mennyt parserista läpi |
| .. | .. | .. |
| 500- | Toimittajakohtaiset |  |
| HL7FICommit errcode arvot | | |

(Tulevaisuudessa todellisessa SOAP-ympäristössä vastaanottosanoma hoidetta­neen SOAP:in Fault-sanomalla).

## Yhteenvetokuvat tietovirroista ja niiden kuittauksista

### Asynkroninen lähetys

|  |
| --- |
|  |
| Kuva 10: Asynkroninen lähetys |

1. Lähettävä sovellusadapteri antaa xml-dokumentin (-dokumentit) tietoliikenneadapterille lähetettäväksi
2. Xml-dokumentit sijoitetaan kuljetuskehykseen ja siirretään http:llä vastaanottavalle tietoliikenneadapterille
3. Vastaanottava tietoliikenneadapteri palauttaa synkronisesti kuittaussanoman HL7FICommit, jossa virhekoodi on 0 OK-tapauksessa, virhetapauksessa virhekoodi > 0
4. Vastaanottava tietoliikenneadapteri purkaa xml-dokumentit kuljetuskehyksestä ja antaa ne vastaanottavalle sovellusadapterille, joka käsittelee xml-dokumentit (aikanaan)
5. Vastaanottanut sovellusadapteri palauttaa sovellustason kuittauksen (aikanaan) vastaanottaneelle tietoliikenneadapterille
6. Sovellustason kuittaus siirretään http:llä lähettäneelle tietoliikenneadapterille (aikanaan)
7. Lähettänyt tietoliikenneadapteri palauttaa synkronisesti kuittaussanoman HL7FICommit, jossa virhekoodi on 0 OK-tapauksessa, virhetapauksessa virhekoodi > 0
8. Lähettänyt sovellusadapteri saa tiedon lähettämiensä xml-dokumenttien hyväksymisestä (aikanaan)

### Synkroninen lähetys

|  |
| --- |
|  |
| Kuva 11: Synkroninen lähetys |

1. Lähettävä sovellusadapteri antaa xml-dokumentin (-dokumentit) tietoliikenneadapterille lähetettäväksi
2. Xml-dokumentit siirretään http:llä vastaanottavalle tietoliikenneadapterille
3. Vastaanottava tietoliikenneadapteri palauttaa virhetapauksessa synkronisesti kuittaussa­noman HL7FICommit, jossa virhekoodi > 0
4. Vastaanottava tietoliikenneadapteri purkaa xml-dokumentit kuljetuskehyksestä ja antaa ne vastaanottavalle sovellusadapterille, joka käsittelee xml-dokumentit
5. Vastaanottanut sovellusadapteri palauttaa sovellustason kuittauksen vastaanottaneelle tietoliikenneadapterille
6. Lähettänyt tietoliikenneadapteri saa sovellustason kuittauksen vastaanottaneelta tietoliikenneadapterilta xml-dokumenttilähetyksen paluusanomana.
7. Lähettänyt sovellusadapteri saa tiedon xml-dokumenttien hyväksymisestä

### Aineiston pyyntö (synkroninen)

|  |
| --- |
|  |
| Kuva 12:Aineiston pyyntö (synkroninen) |

1. Pyytävä sovellus pyytää xml-dokumenttia sovellusadapterilta
2. Pyytävä sovellusadapteri siirtää pyynnön http:llä vastaavan sovelluksen tietoliikenneadapterille.
3. Vastaava tietoliikenneadapteri palauttaa virhetapauksessa kuittaussanoman HL7FICommit, jossa virhekoodi > 0
4. Vastaava tietoliikenneadapteri antaa pyynnön vastaavalle sovellusadapterille
5. Vastaava sovellusadapteri palauttaa pyydetyn xml-dokumentin tai sovellustason kuittaussano­man (NAK), jos pyyntö ei onnistunut
6. Pyytävä tietoliikenneadapteri saa pyydetyn xml-dokumentin tai sovellustason kuittauksen vastaavalta sovellusadapterilta kyselyn paluusanomana.
7. Pyytävä sovellusadapteri saa pyydetyn xml-dokumentin tai virheilmoituksen

## Sanomien salaus

Salauksessa käytetään oletusarvoisesti 2-way SSL:ää. VPN on myös mahdollista, jos siitä sovitaan järjestelmätoimittajien kesken. Suojatussa sisä­verkossa salausta ei välttämättä tarvita.

## Xml-dokumenttien version hallinta

Versionhallintaa tehdään seuraavasti:

* + - * Kuljetuskehyksessä on elementti CPAId, joka sisältää kehyksessä kuljetettavan xml-dokumentin määritysdokumentin OID-tunnuksen. Näin kuljetuskehystä käsittelevät sovellukset voivat halutessaan tunnistaa, minkä version mukaista kuljetettava sisältö on.
      * Sovellusten versionhallinnan kannalta on usein käytössä järjestely, jossa sanomien eri versioita käsittelevät eri sovelluskomponentit. Tämä helpottaa sovellusten ylläpitoa ja testausta sekä muutosten hallintaa. Tätä tarkoitusta varten hyödynnetään kuljetuskehyksessä olevia Service- ja Action-muuttujia. Service pysyy samana sekä CDA R1 että CDA R2 sanomille. Action-muuttujaan lisätään etuliite R2, jolla erotetaan toisistaan CDA R1 ja CDA R2 sanomia palvelevat sovelluskomponentit. SOAP-liikennettä käytettäessä siirrettävän sanoman ENV-Body-osassa olevan elementin nimiavaruus osoittaa sovellusluokan ja elementin nimi osoittaa metodin. Näissä käytetään samoja arvoja kuin Service- ja Action muuttujissa. Näin voidaan ohjata sanoma juuri kyseistä versiota käsittelevälle metodille.
      * Siirrettävän CDA R2 dokumentin alussa oleva typeId-elementti kertoo dokumentin täsmällisen määrityksen. Näin voidaan erottaa CDA R2-sanomien alaversiot toisistaan CDA R2 dokumenttia käsittelevän sovelluksen sisällä.
      * Samaa versionhallinnan mekanismia käytetään myös muiden siirrettävien sisältöjen osalta. Toistaiseksi tämä koskee lähetettä ja hoitopalautetta, joista on myös CDA R1 toteutus. Jatkossa tämä versionhallinnan mekanismi koskee kaikkia sisältöjä.

# Tiedonvälitys

HL7 Avoimet Rajapinnat- dokumentissa (Avoin\_Rajapinta\_V1.1.2, 3.2.2003) määritellään täsmällisesti tietojen siirtäminen perusjärjestelmien ja viitetietojärjestelmien välillä. HL7 OpenCDA- dokumentissa (Open\_CDA\_V1.0.doc, 2.2.2004) yleistetään tätä määrittelyä tietojen siirtämiseen yleensä terveydenhuollon tietojärjestelmien välillä eri käyttötarkoituksissa. Perusjärjestelmäadapterin tietosuoja -dokumentissa (Perusjärjestelmäadapterin tietosuoja, Versio 1.0, 27.9.2004) käsitellään SSL-salatun https -protokollan käyttöön liittyviä yksityiskohtia ja käytännön kysymyksiä.

Tämän dokumentin tiedonvälitysosuuden tarkoituksena on määritellä edellä lueteltujen määritysten mukaista tiedonsiirtoa tukeva välityspalvelu. Välityspalvelun tarve on tullut käytännössä tehtyjen toteutusten myötä. Välityspalvelun roolia kuvaa oheinen kuva 1.

|  |  |
| --- | --- |
|  |  |
| Kuva 13: Point-to-point yhteydet ja välityspalvelun kautta toteutettavat yhteydet | |

Point-to-point yhteysmallissa jokaisella tietoliikenteen osapuolella on suora yhteys jokaiseen muuhun tietoliikenteen osapuoleen. Välityspalvelua käyttävässä yhteysmallissa jokaisella tietoliikenteen osapuolella on suora yhteys ainoastaan yhteen osapuoleen eli välityspalveluun. Välityspalvelu puolestaan reitittää sanomat lähettäjältä vastaanottajalle. Tässä dokumentissa kuvattu välityspalvelu toimii ns. siltana eli ei lainkaan käsittele varsinaista sanoman hyötykuormaa vaan ainoastaan sanoman kehykseen sisältyviä reititystietoja.

# Tarve tiedonvälitykselle

## Tietoliikenneadapterit ja sovellusadapterit

Edellä mainituissa määritysdokumenteissa kutsutaan **tietoliikenneadapteriksi** sitä toiminnallisuutta, jolla huolehditaan erilaisten tietosisältöjen lähettämisestä ja vastaanottamisesta tiedonsiirron osapuolten välillä yhdenmukaisella tavalla. **Sovellusadapteriksi** kutsutaan sitä toiminnallisuutta, joka sovelluskohtaisesti kokoaa lähetettävät tietosisällöt sanomaksi tai purkaa vastaanotetut tietosisällöt vastaanottavan sovelluksen käsiteltäväksi. Tarkoitus on, että kertaallen toteutetulla tietoliikenneadapterilla voidaan hoitaa tietoliikenne usean eri sovellusadapterin välillä, jotka voivat sijaita eri tietojärjestelmissä ja eri organisaatioissa.

## Tietoliikenneadapterien parametrointi

Edellä lueteltujen määritysten mukaisia yhteyksiä on toteutettu käyttämällä avoimen internetin yli tehtävää SSL-suojattua tiedonsiirtoa. Tässä dokumentissa kuvattu välityspalvelu perustuu myös avoimen internetin ja SSL-suojattujen yhteyksien käyttöön. Muulla tavalla suojattuja yhteyksiä (esim. VPN yhteyksiä) on myös mahdollista käyttää varsinaisen välityspalvelun toiminnallisuuden muuttumatta. Tiedonsiirron osapuolten tunnistaminen, todentaminen ja tiedonsiirron salaus hoidetaan silloin asianomaisen siirtoyhteyden menetelmin.

Tietoliikenteessä on vakiintunut käytäntö erottaa organisaation sisäverkko palomuureilla organisaatioiden välisestä ulkoverkosta. Sisäverkkoa ja ulkoverkkoa varten on omat palomuurit, joiden väliin jäävää aluetta kutsutaan DMZ-vyöhykkeeksi (Demilitarized Zone, demilitarisoitu vyöhyke). Pääsy tälle vyöhykkeelle on rajoitettua molemmista verkkosegmenteistä. Järjestelyä esittää oheinen kuva.

|  |
| --- |
|  |
| Kuva 14: Tietoliikenneadapterit ja välitysadapteri |

Tietoliikenneadapteri esittää tietoliikenteen yksilöityä osapuolta. Tietoliikenneadapterina käytetään tyypillisesti www-palvelinta ja siihen liittyviä ohjelmistoja. Käytännössä tietoliikenneadapterista voi olla käytössä useampia instansseja eri ympäristöjä varten, esim. testausta ja tuotantoa varten. Kukin instanssi muodostaa tällöin tietoliikenteen yksilöidyn osapuolen. Tietoliikenneosapuoli on nimenomaan looginen osapuoli. Esimerkiksi sama fyysinen palvelin voi hoitaa tietoliikennettä viitejärjestelmää, hoitotakuun seurantaa, E-lausuntoja ja lähete-palauteliikennettä varten. Näitä kutakin käyttötarkoitusta varten voi olla oma OID osapuolitunnus ja sitä vastaava URL.

Jatkossa kun tässä tekstissä viitataan tietoliikenneadapteriin tarkoitetaan nimenomaan yksilöityä tietoliikenteen osapuolta.

Tietoliikenteen osapuoleen liittyvät mm. seuraavat ominaisuustiedot:

* Tietoliikenteen osapuolen yksilöivä OID-tunnus, solmuluokka 18
* URL-osoite
* IP-osoite ja portti
* SSL-varmenne, jolla kolmas osapuoli varmentaa kyseisen tietoliikenteen osapuolen autentisuuden

Kutakin kahden tietoliikenneadapterin välistä yhteyttä perustettaessa on vaihdettava kyseiset tiedot sekä varmenteet. Lisäksi alussa viitattujen määritysdokumenttien mukaisesti on sovittava kahdenvälisestä aikakatkaisun pituudesta, uudelleenyritysten lukumäärästä sekä toipumismenettelystä virhetilanteiden yhteydessä. Lisäksi on kummankin osapuolen avattava palomuuriinsa toista osapuolta varten tarvittavat yhteydet.

## Tiedonvälityspalvelun tarve

Käytännössä siis tietoliikenneyhteyden perustaminen kunkin kahden osapuolen välille vaatii melkoisesti parametrien asettamista ja palomuurien konfigurointia sekä SSL-varmenteiden vaihtamista. Lisäksi kukin yhteys vaatii tuotannon aikaista valvomista ja virhetilanteiden selvittelyä. Tietoliikenteen osapuolten lukumäärän kasvaessa kasvaa kuvan 1 mukaisessa point-to-point mallissa kahdenvälisten yhteyksien lukumäärä voimakkaasti. Jokaiselta osapuolelta vaadittava työmäärä kasvaa tällöin todella suureksi.

Kuvan 1 mukaista tiedonvälityspalvelua käytettäessä on kutakin tietoliikenneadapteria varten tarpeen perustaa vain yksi yhteys tietoliikenteen välitysadapteriin. Tässä mallissa tietoliikenteen perustamisen ja hoitamisen kultakin tietoliikenteen osapuolelta vaatima työmäärä on oleellisesti pienempi. Toisaalta uutena toimijana on välityspalvelun tarjoaja, jonka on siis perustettava yhteydet jokaiseen tietoliikenteen osapuoleen. Kuitenkin kokonaisuutena välityspalvelun käyttö vähentää tarvittavaa perustamisen ja hallinnan työmäärää merkittävästi ja on suositeltava etenemissuunta.

Tämän dokumentin tarkoituksena on siis määritellä välityspalvelun toiminnallisuus siten, että määrityksen pohjalta voivat kaupalliset yritykset tarjota välityspalvelua tietoliikenteen osapuolten käytettäväksi. Välityspalvelua voi valtakunnassa olla tarjolla usealta eri yritykseltä. Tavoitteena on, että välityspalvelun tarjoajat toteuttavat välityspalvelunsa täsmälleen yhteensopivalla tavalla, jolloin asiakkaat voivat vapaasti valita haluamansa välityspalvelun tarjoajan. Mahdollista on myös asetelma, jossa lähettävä osapuoli käyttää yhtä välityspalvelua, vastaanottava osapuoli toista välityspalvelua, ja välityspalvelut ovat keskenään yhteydessä.

## Rajaukset

Kaupalliset tiedonvälityspalvelut voivat tarjota muitakin toimintoja kuin reitityspalveluita. Näitä voivat olla mm. tietoliikenneprotokollan muunnokset ja mahdolliset tietosisällön muunnokset, broadcasting-palvelut jne. Näiden palveluiden määritykset eivät sisälly tähän dokumenttiin.

# Tiedonvälityksen konfigurointi

Avoimet Rajapinnat -määrityksissä kahden sovelluksen välinen tietojenvälitys perustuu tietoliikenteen osapuolten OID-tunnusten käyttöön. SOAP-kehyksessä olevat elementit To ja From sisältävät elementin PartyID, johon osapuolten OID-tunnus sijoitetaan. Esim. organisaation 1567535-0 tietyn järjestelmän sanomaliikenteen osapuolitunnus voisi olla 1.2.246.537.10.1567535.18.2004.12 (vuonna 2004 käyttöön otettu sanomaliikenteen järjestelmä 12). Saman järjestelmän eri ympäristöt saavat kukin oman tunnuksensa. Näin esimerkiksi testiympäristö ja tuotantoympäristö voidaan erottaa toisistaan.

Tietoliikenneadapterit, jotka käyttävät http-protokollaa tiedonsiirtoon, käyttävät tietoliikenteen osoitteita, jotka ovat url-muotoa, esimerkiksi sovellus1.sairaala.fi/tuotanto. Tähän saattaa vielä liittyä porttinumero.

Tietoliikenteen osoitteiden hallintaa varten on kullakin tietoliikenteen osapuolella oltava käytössään reititystaulukko, joka sisältää kaikkien tarpeellisten tietoliikenteen osapuolten OID-tunnukset ja vastaavat url-osoitteet. Kahdenvälisen tietoliikenteen tapauksessa url-osoitteet ovat täsmälleen OID-tunnuksia vastaavat url-osoitteet. Välityspalvelua käytettäessä taulukoiden sisältämät osoitteet muuttuvat oheisen kuvan osoittamalla tavalla.

|  |
| --- |
|  |
| Kuva 15: Reititystaulukot tietoliikenteen päätepisteissä ja välityssolmussa |

Kunkin solmun reititystaulukossa on lueteltu kaikki ne osapuolet, joiden kanssa solmu joutuu tekemisiin. Välityspalvelun reititystaulukossa ovat OID-tunnuksia vastaavat todelliset url-muotoiset tietoliikenneosoitteet. Päätesolmuissa puolestaan on OID-tunnuksia vastaamassa käytettävän välityspalvelun url-osoite. Näin päätesolmut osoittavat sanomansa loogisesti sovellustasolla toiselle päätesolmulle, mutta fyysisesti tietoliikennetasolla välityspalvelulle.

Tarpeen on, että sanomaliikenteen osapuolten OID-tunnukset ja vastaavat tietoliikenteen url-osoitteet kootaan yhteisesti käytettävään taulukkoon. Niiden avulla voi tietoliikenteen välityspalvelujen tarjoaja konfiguroida asiakkailleen reititystaulukot näiden tietoliikenneadaptereiden käyttöön.

Konfigurointitietoihin kannattaa koota muutakin tietoliikenteen hallinnan kannalta tarpeellista tietoa, kuten ip-osoitteet ja portit, organisaatoiden nimet, varmenteiden voimassaoloajat, aikakatkaisun pituus, virheilmoitusten lähetysosoite jne. Konfigurointitiedot on syytä säilyttää luottamuksellisina.

Periaatteessa välityspalvelun ei tule ottaa kantaa siihen, mihin palveluun kyseisen SOAP-kehyksen sisältämä hyötykuorma on tarkoitettu.

# Tiedonvälityksen toiminnallisuus

## Vuorovaikutustilanteet

Avoimet Rajapinnat määrittelee sovellusten väliselle vuorovaikutukselle kaksi tilannetta: Tiedon lähetys ja tiedon pyyntö. Samat vuorovaikutustilanteet on toteutettava myös välityspalvelua käytettäessä. Tavoitteena on välityspalvelun toteuttaminen "läpinäkyvästi", ts. lähettäjän ja vastaanottajan toiminnallisuus säilyy samana riippumatta siitä, käytetäänkö välityspalvelua vai ei.

Tiedonsiirto tapahtuu käyttämällä POST-metodia. Tiedonsiirtoprotokollaksi http sopii erityisesti perusjär­jestelmäkyselyn ja sen vastauksen toteuttamiseen. Tällöin vastaussanoma saadaan POST:in paluuarvon mukana. Myös tiedon lähettämiseen käytetään http POST-metodia. Tässä tapauksessa tarvitaan siirtotason kuittaussanoma HL7FICommit (jäljempänä commit-sanoma) sekä sovellustason ACK-kuittaussanoma, jossa HL7FIAcknowledgment-elementin attribuutilla ackstatus (ACK, NAK, SOME) ilmoitetaan siirron onnistuminen tai epäonnistuminen. Commit-sanoma on aina synkroninen vastaus POST-metodiin, joten erillistä sanomatunnusta messageId ei tarvita. Katso tarkemmin luvusta 5.3.

## Asynkroninen ja synkroninen tiedonsiirto

Avoimet rajapinnat määrittelee mahdolliseksi käyttää sekä asynkronista että synkronista tiedonsiirtoa. Välityspalvelun käyttäminen saattaa tuoda lähettäjän ja vastaanottajan väliseen vuorovaikutukseen oman lisäviiveensä. Synkronisen tiedonsiirron käyttö välityspalvelun yhteydessä ei ole suositeltavaa, koska välittäjä joutuu pitämään lähettäjän ja vastaanottajan välistä yhteyttä auki ja siihen tarvittavia resurssejaan varattuina. Lisäksi lähettäjä saattaa joutua odottamaan pitkänkin ajan vastaanottajalta tulevaa commit-kuittausta ennekuin pääsee jatkamaan seuraavan lähetyksen valmistelua. Lähetettäessä tiedonsiirto on siis asynkronista siten, että välittäjä kuittaa tiedonsiirron aina commit-kuittauksella lähettäjälle ennenkuin tekee edelleenlähetyksen vastaanottajalle. Tiedon pyyntö voi olla asynkronista tai synkronista riippuen käyttötarkoituksesta ja vastausaikavaatimuksista. Tässä dokumentissa kuvataan pyynnöstä kumpikin tilanne.

## Aineiston lähetys

Oheinen kuva esittää aineiston lähetyksen asetelmaa ja toiminnallisuutta.

|  |
| --- |
|  |
| Kuva 16: Aineiston lähetyksen asetelma ja toiminnallisuus |

Lähettävän sovelluksen (sovellusadapterin) tehtävänä on koota lähetettävä aineisto, lähettää se vastaanottavalle sovellukselle (sovellusadapterille) ja huolehtia aineiston perillemenosta vastaanottavan sovelluksen lähettämien kuittausten avulla. Vastaanottava sovellus voi ilmoittaa aineistossa havaitsemistaan virheistä lähettämällä asianmukaiset virheilmoitukset vastaanottokuittausten mukana. Mikäli lähettäjä ei saa kuittauksia sovitun ajan sisällä, tekee se uudelleen yrityksiä sovitun lukumäärän, jonka jälkeen ilmoittaa virheestä tietoliikenteen toimivuudesta vastaavalle valvojalle.

## Aineiston lähetys yksityiskohtaisesti

1. Lähettävä sovellusadapteri antaa xml-dokumentin (-dokumentit) tietoliikenneadapterille lähetettäväksi. Määritys ei suoranaisesti ota kantaa siihen, tuottaako sovellusadapteri vai tietoliikenneadapteri lähetettävän sisällön xml-muotoon, loogisesti se on sovellusadapterin tehtävä. Samoin sovellusadapterin tehtävä on kertoa vastaanottava sovellus tietoliikenneadapterille, esim. OID-tunnuksen avulla.
2. Lähettävä tietoliikenneadapteri sijoittaa sisällön SOAP-kehykseen, varustaa kehyksen tarvittavilla tiedoilla, mm. vastaanottajan OID-tunnuksella ja välityspalvelun url-osoitteella ja lähettää sanoman http(s) protokollalla välittävälle tietoliikenneadapterille. Tietoliikenne on salattua.
3. Välittäjän tietoliikenneadapteri palauttaa synkronisesti kuittaussanoman HL7FICommit, jossa virhekoodi on 0 OK tapauksessa, virhetapauksessa virhekoodi > 0.
4. Välittäjän tietoliikenneadapteri hakee vastaanottajan OID-tunnusta vastaavan url-osoitteen reititystaulukosta ja lähettää sanoman http(s) protokollalla vastaanottavalle tietoliikenneadapterille. Tietoliikenne on salattua. Mikäli vastaanottaja on tuntematon tai lähetys ei onnistu palauttaa välittäjän tietoliikenneadapteri sovellustason virheilmoituksen kuittaussanomassa.
5. Vastaanottava tietoliikenneadapteri palauttaa synkronisesti kuittaussanoman HL7FICommit, jossa virhekoodi on 0 OK-tapauksessa, virhetapauksessa virhekoodi > 0
6. Vastaanottava tietoliikenneadapteri purkaa xml-dokumentit kuljetuskehyksestä ja antaa ne vastaanottavalle sovellusadapterille, joka käsittelee xml-dokumentit (aikanaan)
7. Vastaanottanut sovellusadapteri palauttaa sovellustason kuittauksen (aikanaan) vastaanottaneelle tietoliikenneadapterille
8. Vastaanottava tietoliikenneadapteri sijoittaa kuittauksen SOAP-kehykseen, varustaa kehyksen tarvittavilla tiedoilla, mm. lähettäjän OID-tunnuksella ja välityspalvelun url-osoitteella ja lähettää sanoman http(s) protokollalla välittäjän tietoliikenneadapterille.
9. Välittäjän tietoliikenneadapteri palauttaa synkronisesti kuittaussanoman HL7FICommit, jossa virhekoodi on 0 OK tapauksessa, virhetapauksessa virhekoodi > 0.
10. Välittäjän tietoliikenneadapteri hakee lähettäjän OID-tunnusta vastaavan url-osoitteen reititystaulukosta ja lähettää sanoman http(s) protokollalla lähettäneelle tietoliikenneadapterille. Tietoliikenne on salattua. Mikäli lähettäjä on tuntematon tai lähetys ei onnistu palauttaa välittäjän tietoliikenneadapteri sovellustason virheilmoituksen kuittaussanomassa vastaanottavalle tietoliikenneadapterille.
11. Lähettänyt tietoliikenneadapteri palauttaa synkronisesti kuittaussanoman HL7FICommit, jossa virhekoodi on 0 OK-tapauksessa, virhetapauksessa virhekoodi > 0
12. Lähettänyt sovellusadapteri saa tiedon lähettämiensä xml-dokumenttien hyväksymisestä (aikanaan)

## Aineiston pyyntö

Pyytävän sovelluksen (sovellusadapterin) tehtävänä on laatia pyyntösanoma, joka yksilöi pyydetyt dokumentit. Pyytävä sovellus lähettää pyynnön aineistoa hallitsevalle sovellukselle, joka palauttaa pyydetyt dokumentit. Dokumenttien palautus toimii tavallaan sovellustason kuittauksena. Mikäli pyydettyjä dokumentteja ei löydy, palauttaa sovellus asiaa koskevan virheilmoituksen. Mikäli pyytäjä ei saa mitään vastausta sovitun ajan sisällä, tekee se uudelleen yrityksiä sovitun lukumäärän, jonka jälkeen se ilmoittaa häiriöstä tietoliikenteen toimivuudesta vastaavalle valvojalle.

Pyyntö voi sisältää dokumentit yksilöivien tunnusten lisäksi erilaisia hakuehtoja, joihin aineistoa hallitseva sovellus palauttaa hakuehtoihin täsmäävät dokumentit. Dokumenttien pyytämistapa ei vaikuta tietoliikenteen toiminnallisuuteen.

Tiedon pyyntö voi olla asynkronista tai synkronista riippuen käyttötarkoituksesta ja vastausaikavaatimuksista.

### Aineiston asynkroninen pyyntö yksityiskohtaisesti

Oheinen kuva esittää välityspalvelun yli tapahtuvaa aineiston asynkronisen pyynnön asetelmaa ja toiminnallisuutta. Tarve saattaa liittyä esimerkiksi potilaskertomusten noutoon arkistosta ajoissa ennen ajanvarausvastaanotolle saapumista.

|  |
| --- |
|  |
| Kuva 17: Aineiston asynkronisen pyynnön asetelma ja toiminnallisuus |

1. Pyytävä sovellusadapteri antaa xml-dokumentin (-dokumenttien) yksilöivät tunnukset tai hakuehdot tietoliikenneadapterille lähetettäväksi aineistoa hallitsevalle sovellukselle. Määritys ei suoranaisesti ota kantaa siihen, tuottaako sovellusadapteri vai tietoliikenneadapteri pyynnön sisällön xml-muotoon, loogisesti se on sovellusadapterin tehtävä. Samoin sovellusadapterin tehtävä on kertoa aineistoa hallitseva sovellus tietoliikenneadapterille, esim. OID-tunnuksen avulla.
2. Pyytävä tietoliikenneadapteri sijoittaa sisällön SOAP-kehykseen, varustaa kehyksen tarvittavilla tiedoilla, mm. aineistoa hallitsevan sovelluksen OID-tunnuksella ja välityspalvelun url-osoitteella ja lähettää sanoman http(s) protokollalla välittävälle tietoliikenneadapterille. Tietoliikenne on salattua.
3. Välittäjän tietoliikenneadapteri palauttaa synkronisesti kuittaussanoman HL7FICommit, jossa virhekoodi on 0 OK tapauksessa, virhetapauksessa virhekoodi > 0.
4. Välittäjän tietoliikenneadapteri hakee aineistoa hallitsevan sovelluksen OID-tunnusta vastaavan url-osoitteen reititystaulukosta ja lähettää sanoman http(s) protokollalla palauttavalle tietoliikenneadapterille. Tietoliikenne on salattua. Mikäli vastaanottaja on tuntematon tai lähetys ei onnistu palauttaa välittäjän tietoliikenneadapteri sovellustason virheilmoituksen kuittaussanomassa.
5. Palauttava tietoliikenneadapteri palauttaa synkronisesti kuittaussanoman HL7FICommit, jossa virhekoodi on 0 OK-tapauksessa, virhetapauksessa virhekoodi > 0
6. Palauttava tietoliikenneadapteri purkaa xml-dokumenttien tunnukset tai hakuehdot kuljetuskehyksestä ja antaa ne palauttavalle sovellusadapterille, joka suorittaa dokumenttien haun aineistoa hallitsevan sovelluksen avulla (aikanaan, kuitenkin mahdollisimman nopeasti).
7. Palauttava sovellusadapteri palauttaa löytyneet dokumentit palauttavalle tietoliikenneadapterille. Mikäli aineisto säilytetään tietokannassa muussa kuin XML-muodossa tekee sovellusadapteri muunnoksen XML-muotoon.
8. Palauttava tietoliikenneadapteri sijoittaa dokumentit SOAP-kehykseen, varustaa kehyksen tarvittavilla tiedoilla, mm. pyytäjän OID-tunnuksella ja välityspalvelun url-osoitteella ja lähettää sanoman http(s) protokollalla välittäjän tietoliikenneadapterille.
9. Välittäjän tietoliikenneadapteri palauttaa synkronisesti kuittaussanoman HL7FICommit, jossa virhekoodi on 0 OK tapauksessa, virhetapauksessa virhekoodi > 0.
10. Välittäjän tietoliikenneadapteri hakee pyytäjän OID-tunnusta vastaavan url-osoitteen reititystaulukosta ja lähettää sanoman http(s) protokollalla aineistoa tietoliikenneadapterille. Tietoliikenne on salattua. Mikäli pyytäjä on tuntematon tai lähetys ei onnistu palauttaa välittäjän tietoliikenneadapteri sovellustason virheilmoituksen kuittaussanomassa vastaanottavalle tietoliikenneadapterille.
11. Pyytänyt tietoliikenneadapteri palauttaa synkronisesti kuittaussanoman HL7FICommit, jossa virhekoodi on 0 OK-tapauksessa, virhetapauksessa virhekoodi > 0
12. Pyytänyt sovellusadapteri saa pyyntöön löytyneet dokumentit ja voi aloittaa niiden jatkokäsittelyn.

### Aineiston pyyntö synkronisesti

Oheinen kuva esittää välityspalvelun yli tapahtuvaa aineiston synkronisen pyynnön asetelmaa ja toiminnallisuutta. Tarve saattaa liittyä esimerkiksi päivystyspotilasta koskevien tietojen katseluun toisen organisaation perusjärjestelmästä.

|  |
| --- |
|  |
| Kuva 18: Aineiston synkronisen pyynnön asetelma ja toiminnallisuus |

1. Pyytävä sovellusadapteri antaa xml-dokumentin (-dokumenttien) yksilöivät tunnukset tai hakuehdot tietoliikenneadapterille lähetettäväksi aineistoa hallitsevalle sovellukselle. Määritys ei suoranaisesti ota kantaa siihen, tuottaako sovellusadapteri vai tietoliikenneadapteri pyynnön sisällön xml-muotoon, loogisesti se on sovellusadapterin tehtävä. Samoin sovellusadapterin tehtävä on kertoa aineistoa hallitseva sovellus tietoliikenneadapterille, esim. OID-tunnuksen avulla.
2. Pyytävä tietoliikenneadapteri sijoittaa sisällön SOAP-kehykseen, varustaa kehyksen tarvittavilla tiedoilla, mm. aineistoa hallitsevan sovelluksen OID-tunnuksella ja välityspalvelun url-osoitteella ja lähettää sanoman http(s) protokollalla välittävälle tietoliikenneadapterille. Tietoliikenne on salattua. Tietoliikenneadapteri jää odottamaan vastausta.
3. Välittäjän tietoliikenneadapteri hakee aineistoa hallitsevan sovelluksen OID-tunnusta vastaavan url-osoitteen reititystaulukosta ja lähettää sanoman http(s) protokollalla palauttavalle tietoliikenneadapterille. Tietoliikenne on salattua. Mikäli vastaanottaja on tuntematon tai lähetys ei onnistu palauttaa välittäjän tietoliikenneadapteri sovellustason virheilmoituksen kuittaussanomassa. Välittävä tietoliikenneadapteri jää odottamaan vastausta.
4. Palauttava tietoliikenneadapteri purkaa xml-dokumenttien tunnukset tai hakuehdot kuljetuskehyksestä ja antaa ne palauttavalle sovellusadapterille, joka suorittaa dokumenttien haun aineistoa hallitsevan sovelluksen avulla mahdollisimman nopeasti.
5. Palauttava sovellusadapteri palauttaa löytyneet dokumentit palauttavalle tietoliikenneadapterille. Mikäli aineisto säilytetään tietokannassa muussa kuin XML-muodossa tekee sovellusadapteri muunnoksen XML-muotoon.
6. Palauttava tietoliikenneadapteri sijoittaa dokumentit SOAP-kehykseen, varustaa kehyksen tarvittavilla tiedoilla, mm. pyytäjän OID-tunnuksella ja välityspalvelun url-osoitteella ja vastaa sanoman http(s) protokollalla välittäjän tietoliikenneadapterille, joka odottaa vastausta.
7. Välittäjän tietoliikenneadapteri vastaa sanoman http(s) protokollalla aineistoa pyytäneelle tietoliikenneadapterille, joka odottaa vastausta. Tietoliikenne on salattua.
8. Pyytänyt sovellusadapteri saa pyyntöön löytyneet dokumentit ja voi aloittaa niiden jatkokäsittelyn.

Edellä esitetty "täysin" synkroninen menettely sisältää sen riskin, että se saattaa sitoa tiedonsiirtokanavan pitkäksi aikaa. Tilanteessa, jossa on paljon yht´aikaisiä käyttäjiä, joilla kullakin on tiukat vastausaikavaatimuksensa tämä ei ole toivottava ilmiö. Toiminnallisesti parempi ratkaisu on edellä kuvattu asynkroninen pyyntö siten täydennettynä, että pyyntösanomaan on lisätty prioriteettitunnus, jonka perusteella palauttava sovellusadapteri tietää vastata pyyntöön mahdollisimman nopeasti.

## Kuittausten käsittelyn yksityiskohtia

Sovellustason kuittauksiin liittyy mm. seuraavia yksityiskohtia, jotka pätevät sekä aineiston lähetykseen että myös aineiston pyyntöön:

* Lähettäjä yksilöi lähettämänsä sanomat SOAP-headerissa olevan messageId-elementin avulla. Lähettäjä odottaa sovellustason kuittauksia samalle messageId:lle. Tästä syystä on tärkeätä, että välityspalvelu ei muuta messageId:tä missään tilanteessa. Samoin vastaanottavan tietoliikenneadapterin tulee käyttää sovellustason kuittaukselle aina samaa messageId:tä. Myös uudelleenlähetystilanteessa, jossa lähettäjä ei saa sovitun ajan kuluessa sovellustason kuittausta tulee uudelleen lähetys tehdä samalla messageId:llä. Vastaavasti välityspalvelu tekee uudelleenlähetykset samalla messageId:llä.
* Vastaanottajan on tarpeen varautua duplikaattisanomien esiintymiseen. Tähän vastaanottaja käyttää messageId:tä. Välityspalvelua käytettäessä saattaa esiintyä tilanne, että lähetyssanoma on välitetty onnistuneesti vastaanottajalle, joka on sen kuitannut onnistuneesti välityspalvelimelle. Tämä ei jostain syystä saa kuittausta lähettäjälle, joka aikakatkaisun jälkeen uudelleenlähettää kyseisen sanoman ja joka siis tulee toiseen kertaan vastaanottajalle. Vastaanottaja lähettää tässä tapauksessa normaalin kuittaussanoman uudestaan.

## Virheilmoitukset ja -koodit

ACK-sanoman (Sovellustason kuittaus) ja Commit-sanoman (siirtotason kuittaus) virhekoodeille ja selityksille otetaan käyttöön seuraavia uusia arvoja:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Virhe-koodi (errcode) | Virheteksti (errTxt) | Virheen selitys | Käyttö | ackcode |
| 150 | Tuntematon vastaanottaja | Välityspalvelu ei löydä reititystaulukostaan SOAP-kehyksessä olevaa To.PartyId:n OID-tunnusta vastaavaa url-osoitetta | kaikki sanomat | NAK |
| 151 | Tuntematon lähettäjä | Välityspalvelu ei löydä reititystaulukostaan SOAP-kehyksessä olevaa From.PartyId:n OID-tunnusta vastaavaa url-osoitetta | kaikki sanomat | NAK |
| 152 | Vastaanottaja ei vastaa | Välityspalvelu ei saa yhteyttä vastaanottajaan sovitun ajan kuluessa | kaikki sanomat | NAK |

## Menettely erilaisissa poikkeustilanteissa

Oheiseen taulukkoon on koottu eri yhteysväleillä esiintyvät mahdolliset ongelmatilanteet ja menettelyt niissä.

|  | Tilanne | Mahdollisia syitä | Menettely |
| --- | --- | --- | --- |
| **0** |  | Normaali tilanne. | Lähettäjä lähettää aineiston, Välittäjä kuittaa tietoliikennetasolla (Commit) vastaanottaneensa aineiston ja lähettää sen edelleen Vastaanottajalle, joka kuittaa vastaanoton tietoliikennetasolla.  Vastaanottaja lähettää sovellustason kuittauksen Välittäjälle, joka kuittaa sen tietoliikennetasolla ja lähettää edelleen sovellustason kuittauksen Lähettäjälle, joka kuittaa sen tietoliikennetason kuittauksella.  Sovellustason kuittaus saattaa sisältää aineistoon liittyvän virheilmoituksen virheestä, joka on tullut ilmi vasta kun vastaanottanut sovellus on käsitellyt saamaansa aineistoa. Lähettäjän tulee voida vastaanottaa sovellustason kuittauksessa tulevat virheilmoitukset ja aineiston korjauksen jälkeen lähettää virheellisiksi merkityt tietueet uudestaan. Käytännössä tämä vaatii käyttäjän puuttumista havaittujen virheiden selvittämiseen ja korjaamiseen. |
| **1** |  | Välittäjä alhaalla  Verkko-ongelmia | 1) Lähettäjä yrittää uudelleen  2) Lähettäjä lähettää tiedonsiirtoyhteyden toimivuuden testaussanoman (R2RRQ) ja tarkistaa, että verkko toimii  3) Lähettäjä ottaa yhteyden välittäjän tukipalveluun |
| **2** |  | Välittäjä toteaa sanomassa virheen  - muotovirhe  - tuntematon vastaanottaja | 1) Välittäjä palauttaa virhekoodin Commit-kuittauksessa  2) Lähettäjä ja välittäjä tarkistavat lähetetyn viestin ja konfiguraatiotiedot virheilmoituksen mukaisesti  3) Osapuolet tekevät tarpeelliset korjaukset ja Lähettäjä lähettää sanoman uudelleen |
| **3** |  | Vastaanottaja alhaalla  Verkko-ongelmia | 1) Välittäjä yrittää uudelleen N kertaa  2) Lähettäjä lähettää tiedonsiirtoyhteyden toimivuuden testaussanoman (R2RRQ) ja tarkistaa, että verkko toimii  3) Välittäjä ottaa yhteyden vastaanottajan tukipalveluun  4) Lähettäjä tekee aikanaan uudelleen lähetyksen |
| **4** |  | Vastaanottaja toteaa sanomassa muotovirheen heti  Vastaanottaja toteaa duplikaatin heti | 1) Lähettäjä, Välittäjä ja Vastaanottaja selvittävät virhetilanteen  2) Ongelman syy korjataan ja lähettäjä lähettää sisällön uudestaan.  Vastaanottaja ohittaa duplikaattisanoman.  Vastaanottaja palauttaa virhekoodin Commit-kuittauksessa välittäjälle  Välittäjä muodostaa kyseiselle messageId:lle sovellustason ACK-kuittaussanoman (ackstatus=NAK) ja sijoittaa siihen vastaanottajan palauttaman virhekoodin ja virheen selostuksen sekä lähettää sovellustason kuittauksen Lähettäjälle |
| **4.1** |  | Vastaanottaja toteaa virheen myöhemmin | Vastaanottaja on purkanut sanoman ja esimerkiksi tietokantaan viennissä huomaa sisällössä virheitä  Tällöin vastaanottaja lähettää sovellustason ACK-kuittaus­sanomassa (NAK, SOME) virhekoodin ja virheen selostustuksen |
| **5** |  | Välittäjä alhaalla  Verkko-ongelmia | 1) Vastaanottaja yrittää uudelleen  2) Lähettäjä lähettää tiedonsiirtoyhteyden toimivuuden testaussanoman (R2RRQ) ja tarkistaa, että verkko toimii  3) Vastaanottaja ottaa yhteyden välittäjän tukipalveluun  4) Lähettäjä tekee aikanaan uudelleen lähetyksen, mikäli kuittaussanoma ei mene aikaisemmin perille |
| **6** |  | Välittäjä toteaa sanomassa virheen  - muotovirhe  - tuntematon vastaanottaja | 1) Välittäjä palauttaa virhekoodin Commit-kuittauksessa  2) Vastaanottaja ja Välittäjä tarkistavat lähetetyn viestin ja konfiguraatiotiedot virheilmoituksen mukaisesti  3) Osapuolet tekevät tarpeelliset korjaukset ja Vastaanottaja lähettää kuittaussanoman uudelleen  4) Lähettäjä tekee aikanaan uudelleen lähetyksen, mikäli kuittaussanoma ei mene aikaisemmin perille |
| **7** |  | Lähettäjä alhaalla  Verkko-ongelmat | 1) Välittäjä yrittää uudelleen N kertaa  2) Lähettäjä lähettää tiedonsiirtoyhteyden toimivuuden testaussanoman (R2RRQ) ja tarkistaa, että verkko toimii  3) Välittäjä ottaa yhteyden vastaanottajan tukipalveluun  4) Lähettäjä tekee aikanaan uudelleen lähetyksen, mikäli kuittaussanoma ei mene aikaisemmin perille |
|  |  |  |  |

Yleisiä kommentteja:

* Montako kertaa kukin osapuoli yrittää uudelleen?
* Kuinka pitkä on kullakin osapuolella aikaväli, jonka jälkeen uudelleenyritys tehdään?

# Muita toteutusnäkökohtia

Ohessa on joitakin näkökohtia toteutuksia ajatellen:

* Välityspalvelu saattaa palvella samanaikaisesti useita tiedonsiirron osapuolia. Erityisen huolellisesti on suunniteltava siirtokapasiteetin tarve ja käyttö siten, ettei meneillään oleva siirto häiritse esimerkiksi samaan aikaan tehtäviä kyselyitä. Kiireellisten pyyntöjen priorisointiin saatetaan tarvita lähetysten tai pyyntöjen kiireellisyysluokitus. Toistaiseksi sitä ei ole määrityksissä mukana, mutta asiaan voidaan palata.
* Välityspalvelu saattaa pitää siirrettävää aineistoa muistissaan jonkin aikaa ennekuin se lähtee eteenpäin. Erityisen huolellisesti pitää suojata väliaikaisesti talletettu aineisto, koska se on hetkellisesti salaamatonta kahden SSL-salatun siirron välillä. Välityspalvelin saattaa salakirjoitaa aineiston, jos se joudutaan välivarastoimaan pidemmäksi ajaksi.
* Välityspalvelulla saattaa olla tarve ilmoittaa siihen liitetyille tietoliikenteen osapuolille mahdollisista katkoista palvelussa, häiriöistä tai muista mahdollisista asioista. Tällaisia ilmoitussanomia ei kuitenkaan ole toistaiseksi sovittu.
* Kunkin tietoliikenneadapterin on tarpeen ylläpitää lokitietoja lähetetyistä sanomista ja niihin saaduista kuittauksista sekä muista huomionarvoisista asioista. Lokitietojen avulla voidaan selvittää virhetilanteita sekä toistaalta seurata tietoliikenteen toimivuutta ja volyymejä.
* Tietojensiiron ketjussa Lähettäjän ja Vastaanottajan välillä saattaa olla useampiakin välityspalvelimia. Toimintaperiaate on edellä kuvatun mukainen, eli kukin välityspalvelu toteuttaa välityspalvelimen tehtävät samalla tavalla. Reititystaulukoiden perusteella kukin välityspalvelu tietää, mihin sanomat ja kuittaukset lähetetään.

# PULL-malli

Välityspalvelu voi myös toimia ns. PULL-periaatteella, eli Vastaanottaja käy noutamassa välityspalvelulta sinne kertyneet sanomat. Myös PULL-mallissa on yksityiskohtaisesti sovittava SOAP-kehyksen parametrit, vuorovaikutustilanteiden käsittely sekä erilaiset virhe- ja poikkeustilanteet ja niistä toipumismenettelyt. Niiden määrittely on tämän dokumentin ulkopuolella.