

Lappeenrannan teknillinen yliopisto  
Tietotekniikan osasto

## Sähköisten reseptien hallinta

Diplomityön aihe on hyväksytty Tietotekniikan osaston osastoneuvostossa 12.4.2006

Tarkastaja Juha Puustjärvi

Ohjaaja Leena Puustjärvi

26.5.2006

Heidi Eklund-Karvonen

Hallituskatu 17 A 11

45100 Kouvola

040-522 1416

## **TIIVISTELMÄ**

Lappeenrannan teknillinen yliopisto

Tietotekniikan osasto

Heidi Eklund-Karvonen

### **Sähköisten reseptien hallinta**

Diplomityö

2006

98 sivua, 17 kuvaa, 3 taulukkoa ja 3 liitettä

Tarkastajat: Professori Juha Puustjärvi

Proviisori Leena Puustjärvi

Hakusanat: Sähköinen resepti, XML, RDF, RDF Schema, kyselyt

Keywords: Electronic prescription, XML, RDF, RDF schema, queries

Sähköinen reseptijärjestelmä nykyisessä muodossaan sai alkunsa Sosiaali- ja terveysministeriön aloitteesta ja Kansaneläkelaitoksen toimesta vuoden 2001 aikana. Käytössä oleva kansallinen reseptipalvelin mahdollistaa potilaiden oikeudet muun muassa apteekin vapaaseen valintaan ja lääkkeiden suorakorvausmenetelmään. Sähköinen resepti pyritään integroimaan olemassa olevien potilasrekisterijärjestelmien yhteyteen, joten semanttisen webin tuomat teknologiat kuten Web-palvelut ja XML-pohjaiset kielet ovat tärkeässä asemassa reseptien hallinnassa.

Tässä työssä pyritään esittämään sähköisen reseptijärjestelmän toimintaa ja sen mahdollistavia jo olemassa olevia tekniikoita. Esille tuodaan myös uusia lisäarvoa tuovia tekniikoita, kuten RDF(S)-kuvauskielet, jotka mahdollistavat muun muassa resepteihin kohdistuvat kyselyt ja täten uudet terveyspalvelut sekä potilaille että lääkäreille. Oleellista on tietenkin myös tietoturva, sillä organisaatioiden välillä resepteissä liikkuu arkaluontoisia tietoja. Turvallisuutta pyritään edistämään muun muassa sähköisillä allekirjoituksilla ja älykorttien avulla tapahtuvalla järjestelmään tunnistautumisella.

Jotta sähköinen resepti toimisi tulevaisuudessakin halutulla tavalla ja mahdollistaisi lisäarvopalveluita, on siinä panostettava sisällön lisäksi sen kuvaamiseen. Sisällön ja sen suhteiden tarkka määrittely auttaa muun muassa reseptikyselyjen tekemisessä, mikä onkin olennainen osa potilasturvallisuuden ylläpitoa. Kuvauskielten ja ontologioiden käyttö voi myös auttaa lääkehoidon määrittämisessä useiden erilaisten ja jatkuvasti lisääntyvien lääkkeiden viidakossa, kun lääkkeitä voidaan etsiä tietokannoista reseptin kirjoituksen yhteydessä.

## **ABSTRACT**

Lappeenranta University of Technology

Department of Information Technology

Heidi Eklund-Karvonen

### **Electronic prescription management**

Thesis for the Degree of Master of Science in Technology

2006

98 pages, 17 figures, 3 tables and 3 appendices

Examiners: Professor Juha Puustjärvi

M.Sc.(Pharm.) Leena Puustjärvi

Keywords:

Electronic prescription, XML, RDF, RDF schema, queries

In its current form, the Electronic Prescription System in Finland started in 2001 with an initiative made by the Ministry of Social Affairs and Health that was carried out by the Social Insurance Institution. The system uses a centralized national prescription server that enables the patient's rights to choose his pharmacy and the direct compensation of prescription drugs. The electronic prescription will be integrated in the existing patient record systems, which places the technologies of the semantic web, such as XML and Web Services, in an important position in managing electronic prescriptions.

This thesis aims at portraying the electronic prescription system and the existing techniques it uses. Also new valuable techniques, like the RDF(S), that make queries and other health related services for both patients and doctors possible, are introduced. Information security plays an important role here as well, since, for example, personal data is transferred from one organization to another. The information security issues are battled with, for example, digital signatures and identification and authentication with smartcards.

In order for the electronic prescription to work as planned and enable additional health services, emphasis must be put on describing the data as well as the contents. Careful definition of the prescription content and its relations will help in, for instance, making specific queries, which is an essential part of patient security. The usage of description languages and ontologies might also help in determining the proper medication for a patient when drugs can be searched while writing the prescription.

# SISÄLTÖ

## SISÄLLYSLUETTELO

## KUVALUETTELO

## TAULUKKOLUETTELO

## LYHENTEET

## SELITYKSET

1	JOHDANTO .....	1
2	JOHDATUS RESEPTEIHIN.....	4
3	SÄHKÖINEN RESEPTI.....	8
3.1	Toimintamalli Suomessa.....	9
3.1.1	Vaatimuksia ja oletuksia .....	9
3.1.2	Keskitetty tietokantamalli .....	11
3.1.3	Sähköisen reseptin käsittely .....	13
3.2	Standardointi sähköisessä reseptissä.....	17
3.2.1	Health Level 7 .....	17
3.2.2	International Organization for Standardization.....	19
3.2.3	ITSEC ja Common Criteria - kriteeristöt.....	20
4	SÄHKÖISEN RESEPTIN TEKNIKOITA.....	22
4.1	Sähköisen reseptin järjestelmät Suomessa.....	22
4.1.1	Elres.....	23
4.1.2	Effica eResepti .....	23
4.1.3	Pegasos.....	24
4.1.4	HealthNet .....	26
4.1.5	Salix ja Linnea.....	27
4.2	XML - metakieli.....	28
4.2.1	DTD - määrittely .....	31
4.2.2	XML Schema .....	32
4.3	RDF - kuvauskieli .....	34
4.3.1	RDF Schema .....	35

4.3.2	OWL - ontologiakieli .....	37
4.4	Älykortti ja sähköinen resepti .....	38
4.5	Digitaalinen allekirjoitus .....	41
4.5.1	Tiiviste .....	43
4.5.2	XML Allekirjoitus.....	44
4.6	Web-palvelut .....	47
4.6.1	Palvelukeskeinen arkkitehtuuri .....	51
4.6.2	SOAP - protokolla.....	54
4.6.3	WSDL - kieli .....	58
4.7	Liiketoimintamallit.....	60
4.7.1	BPEL4WS-määrittelykieli .....	61
4.7.2	BPML-mallinnuskieli.....	61
4.8	Muita tekniikoita .....	62
4.8.1	VPN - tekniikka .....	62
4.8.2	SSL - tietoturvaprotokolla.....	63
4.8.3	EDI ja EDIFACT .....	64
4.8.4	VLAN - verkkotekniikka .....	65
5	SEMANTTISEN WEBIN TEKNIKOIDEN HYÖDYNTÄMINEN .....	67
5.1	XML:n käyttö sähköisessä reseptissä.....	67
5.1.1	XML Schema .....	71
5.1.2	XPath - polkulausekkeet .....	76
5.2	XML-pohjaisten kielten käyttö sähköisen reseptin hallinnassa .....	79
5.2.1	RDF – kuvauskielen hyödyntäminen .....	79
5.2.2	RDFS - kaaviotekniikka.....	81
5.2.3	OWL – ontologiakielen hyödyntäminen.....	84
5.2.4	RQL - kyselykieli.....	86
6	XML-TEKNIKOIDEN TOTEUTUS .....	89
6.1	Protégé - ontologiaeditori.....	89
6.2	Muita sovelluksia .....	94
7	YHTEENVETO .....	96

## LIITTEET

- 1 LAINSÄÄDÄNTÖ SUOMESSA
- 2 PILOTOINTI
- 3 SÄHKÖINEN RESEPTI MAAILMALLA

## **Kuvaluettelo**

- Kuva 1 Kelan hyväksymä reseptilomake ja siihen laitettavat merkinnät [Poliklinikka.net 2004], s. 5
- Kuva 2 Toimintakaavio sähköisen reseptin käsittelystä [Koponen-Piironen et al. 2001], s. 15
- Kuva 3 CDA R2 dokumentin periaatteellinen rakenne [HL7 2006], s. 18
- Kuva 4 Pegasos-järjestelmän arkkitehtuuri [Kilpivuori 2006], s. 25
- Kuva 5 Luokkien hierarkia [Antoniou & Harmelen 2004 s. 82], s. 36
- Kuva 6 Sirukortti ja SIM-kortti [Setec 2006; Wikipedia 2006], s. 39
- Kuva 7 Radioaalloilla toimiva ja lukijan vaatima sirukortti [Gemplus], s. 40
- Kuva 8 Digitaalisen allekirjoituksen muodostaminen [ABA 2006], s. 42
- Kuva 9 Digitaalisen allekirjoituksen tarkastaminen [ABA 2006], s. 43
- Kuva 10 Web-palvelun käynnistäminen [Booth et al. 2004], s. 48
- Kuva 11 Web-palvelun arkkitehtuuria kuvaava pino [Booth et al. 2004], s. 50
- Kuva 12 SOA:n erilaisia toteutusvaihtoehtoja [Endrei et al. 2004 s.28], s. 52
- Kuva 13 Kaavio RDF:n ja RDFS:n kerroksista ja suhteista toisiinsa [Antoniou & Harmelen 2004 s. 83-90], s. 81
- Kuva 14 Esimerkkireseptijärjestelmän luokat ja niiden suhteet, s. 90
- Kuva 15 Protégén käyttöliittymä ja esimerkkijärjestelmään talletetut luokat, s. 91
- Kuva 16 Protégén kyselyt-välilehti ja esimerkkikysely, s. 92
- Kuva 17 Protégén kyselyikkuna graafisia kyselyihin, s. 94

## **Taulukkuuettelo**

- Taulukko 1 W3C:n määrytykset XML-allekirjoituksen kanssa käytettäville algoritmeille [Bartel et al. 2002], s. 46
- Taulukko 2 Esimerkkireseptijärjestelmän luokat instansseineen, s. 90
- Taulukko 3 Protégélle esitettyjä kysymyksiä ja vastauksia, s. 92

## Lyhenteet

ABS	Acrylonitrile Butadiene Styrene
ANSI	American National Standards Institute
ASN.1	Abstract Syntax Notation One
ATC	Anatomical-Therapeutical-Chemical –lääkeluokka
CCOW	Clinical Context Object Workgroup
CDA	Clinical Document Architecture
CPU	Central Processing Unit
CSS	Cascading Style sheet
DB2	DataBase2
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
DTD	Document Type Definition
EDI	Electronic Data Interchange
EDIFACT	EDI For Administration, Commerce and Transport
EEPROM	Electronically Erasable Programmable Read Only Memory
EiRes	Elektroninen Resepti
EMV	Europay, Master Card, Visa
EPS	Electronic Prescription System
ETP	Electronic Transfer of Prescriptions
FTP	File Transfer Protocol
HHS	United States Department of Health and Human Services
HL7	Health Level Seven
HMAC	keyed Hash Message Authentication Code
HST	Henkilön Sähköinen Tunnistaminen
HTML	HyperText Markup Language
HTTP	HyperText Markup Language
HUS	Helsingin ja Uudenmaan Sairaanhoidopiiri
HYKS	Helsingin Yliopistollinen KeskusSairaala



ICT	Information and Communication Technology
IP	Internet Protocol
IPSec	Internet Protocol Security
ISO	International Organization for Standardization
ITSEC	Information Technology Security Evaluation Criteria
J2EE	Java 2 Platform, Enterprise Edition
MAC	Message Authentication Code; Medium Access Control
MD5	Message Digest 5
NHS	National Health Service (Englanti)
NIST	National Institute of Standards and Technology
OID	Object Identifier
OWL	Web ontology Language
PCDATA	Parserable Character DATA
PCT	Primary Care Trust
PIN	Personal Identification Number
PKI	Public Key Infrastructure
PP	Protection Profile
PPA	Prescription Pricing Authority
PPTP	Point-to-Point Tunneling Protocol
PRA	Patient Record Architecture
PVC	PolyVinyl Chloride, Polyvinylikloridi
RAM	Random Access Memory
RDF	Resource Description Framework
RDFS	RDF Schema
RDQL	RDF Data Query Language
RIM	Reference Information Model
ROM	Read Only Memory
RQL	RDF Query Language
RSA	Rivest, Shamir, Adleman
SeRQL	Sesame RDF Query Language

SGML	Standard Generalized Markup Language
SHA	Secure Hash Algorithm
SIM	Subscriber Identity Module
SMTP	Simple Mail transfer Protocol
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SSL	Secure Sockets Layer
ST	Security Target
SV-numero	SairausVakuutus-numero
SWIM	Subscriber/WAP Identity Module
TCSEC	Trusted Computer Security Evaluation Criteria, "Orange Book"
TEO	Terveysturvituksen Oikeusturvakeskus
TOE	Target Of Evaluation
TPDU	Transport Protocol Data Unit
UDDI	Universal Description, Discovery and Integration
URI	Universal Resource Identifier
URL	Universal Resource Locator
USB	Universal Serial Bus
UTF	Unicode Transformation Format
VANS	Value-Added Network Service
VLAN	Virtual Local Area Network
VNR	Pohjoismainen tuotenumero lääkkeille (VareNummeR)
VPN	Virtual Private Network
W3C	World Wide Web Consortium
WSD	Web Service Description
WSDL	Web Services Description Language
XAdES	XML Advanced Electronic Signatures
XML	eXtensible Markup Language
XPath	XML Path Language

XSD	XML Schema Definition
XSL	eXtensible Style Language
XSLT	eXtensible Style Language Transformations

# 1 JOHDANTO

Olemme vahvasti siirtymässä kohti tietoyhteiskuntaa, niin kuin olemme olleet jo useamman vuoden ajan. Tietotekniikka ja sen tuomat laitteet ja sovellukset kehittyvät nopeasti ja yhä varmemmiksi. Tietoturva on yhä useamman mielessä jokapäiväisissä töissä ja siihen panostetaan resurssien ja tarpeen mukaan enenevässä määrin. Tietoa siirretään vieläkin paljon paperin välityksellä, vaikka tiedonsiirto langallisia ja langattomia teitä pitkin laitteiden välillä ilman paperia valtaa alaa enemmän joka hetki. Esimerkiksi kirjastot ovat siirtymässä myöhästymismuistutusten lähettämiseen sähköpostitse tai tekstiviestillä perinteisen paperisen muistutuksen sijaan ja pankeissa on jo pitkään toiminut verkkopankkiin toimitettava sähköinen tiliote, vaikka myös paperivaihtoehto on yhä olemassa. Omat tiedot voidaan niin kirjaston kuin pankin tunnuksilla tarkastaa Internetin välityksellä; pankissa omilla verkkopankkitunnuksilla voidaan muun muassa varata aika pankkiin lainaneuvotteluun tai tilata uusi pankkikortti ja kirjaston kortilla sekä salasanalla voidaan muun muassa uusia lainoja tai varata kirjoja eri kirjastoista. Kun kerran myös kirjastot ja pankit monien muiden lisäksi käyttävät hyödyksi tietoverkkoa, niin mikseivät terveyspalvelutkin?

Terveyskeskuksissa, lääkäriasemilla, apteekkeissa ja muissa terveydenhuollon toimipisteissä on pikkuhiljaa otettu enemmän tietotekniikkaa käyttöön. Eri toimipisteet kuitenkin käyttävät useita eri järjestelmiä ja vaikka potilasrekisterit on useimmissa paikoissa siirretty tietokantoihin, niitä ei välttämättä voida siirtää toimipisteiden tai edes osastojen välillä. Eri sairaanhoitopiireissä on kuitenkin hiljattain aloitettu siirtyminen yhteisiin tietojärjestelmiin. Tällaisia on esimerkiksi Kymenlaakson ja Etelä-Karjalan sairaanhoitopiireissä, joissa toimii yhteinen Kaapo-niminen tietojärjestelmä sekä Pirkanmaan sairaanhoitopiirissä, jossa eri kuntien järjestelmät on yhdistetty YT Tiedon aluetietojärjestelmän avulla. Eri järjestelmien yhdistäminen tai kokonaan uuteen järjestelmään siirtyminen vie aikaa ja kuluttaa sekä henkilöstön että hallinnon voimavaroja. Uusi järjestelmä kun toimii sulavasti vasta, kun kaikki osaavat sitä käyttää. Yhtenäinen tietojärjestelmä kuitenkin vähentää lääkäreiden ja sairaanhoitajien työtä, kun esimerkiksi vieraspaikkakuntalaisten potilastietoja ei tarvitse

erikseen tilata toisesta sairaalasta, vaan ne saadaan käsittelyyn tietokoneella saman tien. Tämä myös lisää potilaiden hyvinvointia ainakin osittain vähentämällä odotusaikoja. [Peltonen 2006 s.24; YT Tieto 2006]

Aikaisempaa tutkimusta sähköisestä reseptistä on tehty useissa maissa järjestelmien pilotoinnin sekä käyttöönoton yhteydessä. Myös tutkijat eri yliopistoissa ovat tehneet reseptijärjestelmiin, reseptiin liittyviin tekniikoihin tai potilaan ja lääkärin oikeuksiin ja hyötyihin kohdistuvia tutkimuksia. Muun muassa Ball, Chadwick ja Mundy Salfordin yliopistosta ovat tehneet paljon tutkimusta sähköisistä resepteistä ja niiden turvallisuudesta [Ball et al. 2003; Chadwick & Mundy 2003 & 2004]. Sirukorttipohjaista sähköisen reseptin järjestelmää ovat tutkineet esimerkiksi Yang et al. [Yang et al. 2004] ja Suomen reseptijärjestelmän pilotoinnin yhteydessä tutkimuksen ovat julkaisseet ainakin Hyppönen et al. [Hyppönen et al. 2005]. Lisäksi metakieliä ja niiden käyttöä on tutkittu eri tarkoituksissa. XML (eXtensible Markup Language) ja RDF (Resource Description Framework) ovat olleet tutkimuskohteena muun muassa Hart & Deckerillä [Hart & Decker 2005] sekä Kleinilla [Klein 2002], joilla aiheina olivat verkkoon kohdistuvien RDF-kyselyjen optimoidut kyselyt ja XML, RDF ja niiden suhteet. Kyselykielistä RQL:ää (RDF Query Language) ovat tutkineet esimerkiksi Karvounarakis et al. vuonna 2002 [Karvounarakis et al. 2002].

Tässä diplomityössä on tarkoitus antaa kattava yleiskatsaus sähköisestä reseptistä. Työssä kerrotaan miten sähköinen reseptijärjestelmä toimii ja mitä sen kehittämisessä tulisi ottaa huomioon. Lisäksi kuvataan siihen liittyviä erilaisia tekniikoita, niiden käyttöä ja lainsäädäntöä sekä mahdollisia standardeja, jotka koskevat itse sähköistä reseptiä tai siihen liittyviä tekniikoita. Työssä pyritään esittämään myös mahdollisia uusia ratkaisuja, joilla voidaan parantaa tietoturvaa tai mahdollistaa uusia terveyspalveluita sekä potilaille että lääkäreille. Lopuksi onkin tarkoitus keskittyä sähköisen reseptin tietosisällön kuvaamiseen käyttämällä XML- ja RDF/S-tekniikoita sekä ontologioita. Tietosisällön esittäminen metakielellä tuo esille uusia käyttökonteksteja terveydenhuoltopalveluille ja sähköiselle reseptille ja nämä luovat lisäarvoa sekä potilaalle että lääkettä määräävälle lääkärille.

Tällaisia ovat muun muassa resepteihin kohdistuvat kyselyt, joiden avulla lääkärit voivat selvittää potilaiden aikaisempaa lääkitystä nykyisen tueksi tai viranomaiset voisivat selvittää mahdollisia väärinkäytöksiä.

Diplomityössä keskitytään reseptin tekniseen toimintaan ja Suomessa käytössä oleviin tekniikoihin. Lisäksi tutkitaan meta- ja kuvauskielten tuomia hyötyjä sähköisen reseptin käyttöön. Laajan aihealueen rajaamiseksi muun muassa liiketoimintamallit ja niiden kuvaaminen sekä muissa maissa käytössä olevia tekniikoita on selvitetty vain lyhyesti. Työ koostuu viidestä osiosta, joista ensimmäisessä selvitetään sähköisen reseptin toimintamallia Suomessa ja sen standardointia. Toisessa osiossa kuvataan sähköiseen reseptiin liittyviä tekniikoita ja niiden käyttöä ja kolmannessa selvitetään kuinka semanttisen webin tekniikoita voitaisiin paremmin hyödyntää sähköisessä reseptissä ja sen toiminnassa. Lopuksi sähköisen reseptin hallintaa pyritään mallintamaan ja testaamaan vapaan lähdekoodin omaavalla Protégé-ohjelmalla ja esitetään yhteenveto käsitellyistä asioista ja saaduista tuloksista.

## 2 JOHDATUS RESEPTIIN

Tyypillisellä lääkärikäynnillä potilas kertoo ensin lääkärille vaivansa, ja lääkäri tekee vaivasta diagnoosin. Lääkäri voi määrätä potilaalle tutkimuksia, joista hän kirjoittaa lähetteen esimerkiksi laboratorioon tai röntgeniin ja joiden jälkeen hän päättää potilaan hoidosta. Lääkehoidon valitessaan lääkäri ottaa esille Kelan hyväksymän reseptilomakkeen (kuva 1) tai kirjoittaa reseptin tietokoneella ja tulostaa sen sitten potilaalle. Sosiaali- ja terveysministeriön säädöksen (726/2003) mukaan tämän lääkemääräyksen tulee sisältää seuraavat tiedot:

- Potilaan nimi, syntymäaika tai henkilötunnus ja alle 12-vuotiaan paino
- Lääkkeestä lääkeaineen nimi tai sen myyntinimike, lääkkeen vahvuus ja määrä, lääkemuoto, lääkityksen kestoaika, mahdollinen lääkevaihtoa kieltävä merkintä sekä apteekissa valmistettavan lääkkeen koostumus ja määrä (kuva 1, 1.)
- Lääkkeen annostus- ja käyttöohje sekä lääkityksen tyyppi (tarvittaessa vai säännöllisesti) ja käyttötarkoitus, ellei sen poisjättämiselle ole perusteita (kuva 1, 2.)
- Lääkemääräyksen antamipaikkakunta ja päivämäärä, lääkkeen määränneen henkilön allekirjoitus sekä nimileimasin, josta tulee ilmetä määrääjän sairaskäytösnumero, oppiarvo ja erikoisala (kuva 1, 13.)
- Merkintä ”Sic” mikäli lääkemääräys tai ex-tempore lääkevalmiste ylittää valmisteyhteenvedossa ilmoitetun tai myyntiluvallisen annostusohjeen. [726/2003]

Reseptin saatuaan potilas valitsee mieleisensä apteekin ja menee sinne noutamaan lääkettä. Farmaseutti ottaa reseptin vastaan ja pyytää potilaan Kela-korttia, jolla varmistetaan potilaalle kuuluvat korvaukset lääkkeestä. Farmaseutti siirtää reseptissä olevat tiedot omaan järjestelmäänsä ja merkitsee reseptiin seuraavat merkinnät:

- Potilaan ostama lääkemäärä (kuva 1, 3.)
- Reseptille jäänyt vielä saamatta oleva lääkemäärä (kuva 1, 6.)

- Proviisorin tai farmaseutin nimikirjaimet (kuva 1, 5.)
- Lääkkeen toimittaneen apteekin leima ja lääkkeen toimittamispäivämäärä (kuva 1, 4.)
- Apteekin reseptinumero (kuva 1, 7.)
- Reseptin tyyppi; A = alkuperäinen, U = uusittu, P = puhelinresepti, I = lääkettä saa myös ilman reseptiä (kuva 1, 8.)
- Lääkkeen normaalihinta ennen Kelan lääkekorvausta (kuva 1, 9.)
- Lääkkeen toimituspäivä ja – määrä (kuva 1, 10.)
- Lääkevalmisteen pohjoismaisen tuotenumero (kuva 1, 11.)
- Kelan korvausryhmä; O = peruskorvaus, Y = alempi erityiskorvaus, K = ylempi erityiskorvaus, EK = ei Kela-korvausta [Poliklinikka.net 2004; Kela 2006]

Osa 1 RESEPTI

Potilaan nimi: *Anna Allergja* Syntymäaika: *2/12/47* Alle 12-vuotiaan paimo

Kysymykseessä on: *hoidon* Työntekijän ja vakuutusyhtiön nimi (täytetään työtataturmistapöytäkirjassa)

1 Rec: *Lääke, vahvuus, määrä tai hoidon kestoaika, annostus ja käyttötarkoitus*

1. *Chinocort puhkalevy* 7. *Hinnoinn. m. 216,32* 8.

*N<sup>o</sup> V 5.* 12. *Rx: 53377 U* 9. *C 216,32*

2. *2 annosta* 10. *Tein/E-p 1-200 DOS*

*aamit. Allergiat.* 11. *€ RSTR/215257*

2 Rec

Päikkä ja aika

Lääkärin allekirjoitus: *[Signature]* 13. *SV-leima tai nimen selvitys ja SV-numero*

Tämän lomakkeen täytettiin kokonaan lääkäri- ja apteekki-

TOIMITTAMISTA KOSKEVAT MERKINNÄT EI SAA PEITTÄÄ

3. *X* Otetaan 2 *ANNA APTEEKKI* 4. *2 & 04 99*

5. *15. 01 99* 6. *06. 09 1999* *ROJUVAARIN APTEEKKI*

7. *06. 09 1999* *OULU*

8. *OULU*

9. *OULU*

10. *OULU*

11. *OULU*

12. *OULU*

13. *OULU*

14. *OULU*

15. *OULU*

16. *OULU*

17. *OULU*

18. *OULU*

19. *OULU*

20. *OULU*

21. *OULU*

22. *OULU*

23. *OULU*

24. *OULU*

25. *OULU*

26. *OULU*

27. *OULU*

28. *OULU*

29. *OULU*

30. *OULU*

Kuva 1 Kelan hyväksymä reseptilomake ja siihen laitettavat merkinnät [Poliklinikka.net 2004]



Vuonna 2005 Suomessa määrättiin yli 26,7 miljoonaa sairausvakuutuskorvaukseen oikeuttavaa reseptiä, joista puolet kirjoitettiin terveyskeskuksissa. Terveyskeskuslääkärien määräämiä reseptejä toimitettiin apteekkeista yli 13 miljoonalla reseptillä. Vaikka terveyskeskuksissa toimivien lääkärin määräämät lääkkeet olivat pääsääntöisesti edullisempia kuin keskivertolääkäri määräämät lääkkeet, kaikkien korvattujen lääkkeiden kokonaiskustannukset olivat noin 1,6 miljardia euroa. Jokaisen reseptin kohdalla lääkäri ensin kirjoittaa lääkemääräyksen joko käsin tai tietokoneella ja merkitsee sitten lääkkeet oman toimipisteensä potilasrekisteriin eli tietojärjestelmään. Potilas ottaa reseptin vastaan ja vie sen valitsemaansa apteekkiin, jossa farmaseutti ottaa reseptin ja kirjoittaa sen tiedot omaan tietojärjestelmäänsä. Apteekin tietojärjestelmässä reseptiä säilytetään 13 kuukautta ja lisäksi apteekin on pidettävä resepteistä paperimuotoista reseptipäiväkirjaa, jota säilytetään viisi vuotta. Näiden lisäksi alkuperäistä reseptiä vaativien lääkkeiden ja huumausaineiden reseptejä on säilytettävä apteekissa 10 vuotta. [Klaukka et al. 2005; Koponen-Piiroinen et al. 2001]

Suomessa sairausvakuutuslain mukaan sairaanhoitona voidaan korvata lääkärin määräämä lääke, vaikean sairauden hoitoon vaadittava lääkärin määräämä kliininen ravintovalmiste sekä apteekista hankittava lääkärin määräämä ihotaudin hoitoon tarvittava perusvoide. Korvausluokkia on kolme; peruskorvaus, joka korvaa 42 prosenttia lääkkeen hinnasta, alempi erityiskorvaus, joka korvaa 72 prosenttia lääkkeen hinnasta, ja ylempi erityiskorvaus, joka korvaa 100 prosenttia lääkkeen hinnasta, mutta jokaista lääkettä kohti on kolmen euron omavastuu. Vuodesta 1970 on ollut toiminnassa suorakorvausmenettely, jossa apteekki voi myöntää sairausvakuutuksen mukaisen korvauksen suoraan lääkkeen hinnasta. Jokainen apteekki laatii tietyin väliajoin suorakorvausmenettelyn mukaisen tilityksen Kelalle, joka tekee lääkekorvausratkaisut tilitysten perusteella. [Kela 2006; Koponen-Piiroinen et al. 2001]

Yhden ainoan reseptin määräämiseen, toimittamiseen ja korvaamiseen vaaditaan siis paljon työtä, josta suuri osa on edellisen toistamista. Lisäksi lääkkeiden määrä kasvaa joka vuosi, ja samalla lääkkeiden yhteisvaikutukset ja mahdolliset reaktiot niihin muuttuvat ja

lisääntyvät. Lääkärin on varmistettava, ettei lääke reagoi minkään potilaan aiemman lääkkeen kanssa tai ettei se sisällä jotakin hänelle sopimatonta lääkeainetta. Potilaille määrätään lääkkeitä usein usean eri terveydenhuolto-organisaation toimesta, joten kaikkea lääkitystä ei voida tarkastaa sen hetkisen lääkäinvastaanoton potilasrekisteristä. Osa potilaista ei välttämättä muista mitä lääkkeitä heillä on tai mihin vaivaan ne heille on määrätty, mikä lisää työmäärää entisestään. Potilaat eivät myöskään välttämättä muista ottaa muiden lääkkeiden reseptejä mukaan vastaanotolle, tai osa niistä saattaa olla kateissa. Nopeasti varsinkin käsin kirjoitetut reseptit saattavat olla epäselviä ja apteekkihenkilökunnan on soitettava lääkkeen määränneellä lääkärille varmistuakseen lääkkeen esimerkiksi lääkkeen määrästä tai annostuksesta. [Koponen-Piiroinen et al. 2001]

Työn helpottamiseksi, saman työn toistamisen ja odotusaikojen vähentämiseksi, sekä potilaiden hyvinvoinnin edistämiseksi onkin kehitteillä ollut jo pitkään sähköinen resepti. Sähköisen reseptin perusideana on, että lääkäri kirjoittaa lääkemääräyksen tietokoneella suoraan tietojärjestelmään, jossa hän voi mahdollisesti samalla tarkastaa muun muassa lääkkeiden yhteis- tai sivuvaikutuksia ja hintoja. Lääkäri allekirjoittaa reseptin digitaalisesti ja lähettää sen yhteiseen reseptitietokantaan. Potilaalle reseptistä annetaan potilasohje, jossa kerrotaan esimerkiksi lääkkeen nimi, annostus ja käyttö. Potilaan tullessa apteekkiin, farmaseutti hakee reseptin tietokannasta omaan järjestelmäänsä ja toimittaa lääkkeen potilaalle. Kela näkee toimitetut reseptit tietokannasta ja voi niiden perusteella suorittaa lääkekorvaukset apteekkeille. Kerran kirjoitettu resepti siis kiertää yhtenäisten järjestelmien välillä helpottaen lääkäreiden, farmaseuttien ja potilaiden työtä. Kaikki ei kuitenkaan ole aivan niin yksinkertaista kuin miltä se kuulostaa. Reseptit on toimitettava tietokantaan ja sieltä apteekkiin rikkomatta potilaan tietosuojaa ja yksityisyyttä, joten se vaatii muiden muassa tiukkaa tietoturvaa, lääkäreiden toimilupien varmistamista, henkilöstön koulutusta ja tietojärjestelmiin kirjautumista. [Koponen-Piiroinen et al. 2001]

### 3 SÄHKÖINEN RESEPTI

Sähköisellä reseptillä on monta nimeä. Suomessa se tunnetaan perinteisesti *sähköisenä reseptinä* tai *sähköisenä lääkemääräyksenä*, sekä *elektronisena reseptinä* ja lyhyemmin *e-reseptinä*. Myös termiä *ElRes* (Elektroninen Resepti) saatetaan käyttää joissain tilanteissa, mutta se viittaa lähinnä Mediweb Oy:n tuottamaan sähköistä reseptiä käsittelevään ohjelmistoon. Kansainvälisemmin tunnettuja ovat *electronic prescription*, *e-prescription* ja *ePrescription*, jotka kaikki kirjoitusasustaan huolimatta voidaan suomentaa sähköiseksi reseptiksi. Lisäksi *ETP* (Electronic Transfer of Prescriptions) ja *EPS* (Electronic Prescription System) eli reseptien sähköinen siirto ja sähköinen reseptijärjestelmä ovat maailmanlaajuisessa käytössä. Useista nimistään huolimatta peruseräite sähköisessä reseptissä on sama: Lääkäri kirjoittaa reseptin ja lähettää sen sähköisessä muodossa suoraan tai tietokannan kautta apteekkiin, josta potilas voi noutaa lääkkeen ilman paperista reseptilomaketta. [ElRes; Hyppönen et al. 2005]

Sähköinen resepti määrittellään lääkemääräyksen sisältämän lääketieteellisen informaation sähköiseksi kuljetukseksi ja prosessoinniksi kaikkien reseptiin liittyvien osapuolten järjestelmien ja/tai komponenttien kautta. Suomessa sähköistä reseptiä on testattu useita kertoja jo 1980-luvulta lähtien. Useimmiten sähköisen reseptin aiempaa kehittämistä on johtanut jokin kaupallinen järjestö tai yritys, mutta myös Kansaneläkelaitos on kokeillut toimikorttipohjaista sähköistä reseptiä 1990-luvun alussa. Suurin Suomessa järjestetty sähköisen reseptin kehittämis- ja pilotointihanke käynnistyi vuonna 2001 sosiaali- ja terveysministeriön esityksestä ja Kansaneläkelaitoksen toimesta. Projektin tarkoituksena oli selvittää millainen sähköisen reseptin tulisi olla, miten lainsäädäntö Suomessa koskee sähköistä reseptiä ja miten se vaikuttaa sähköisen reseptin toteutumiseen. Lisäksi selvitettiin sähköisen reseptin vaatimia toimintamalleja ja tietotekniikkaa, sekä suoritettiin pilotointia. Mukana yhteistyössä olivat Kansaneläkelaitos, Lääkelaitos, Suomen Apteekkariliitto, Yliopiston Apteekki, Suomalainen lääkärisseura Duodecim, Kymenlaakson

sekä Helsingin ja Uudenmaan sairaanhoitopiirit, Suomen lääkäriliitto sekä sosiaali- ja terveystieteiden ministeriö. [Koponen-Piiroinen et al., 2001]

### **3.1 Toimintamalli Suomessa**

Suomessa sähköisen reseptin idea lähti liikkeelle oletuksesta, että niin sanottu turha työ jäisi reseptien kirjaamisessa ja siirtämisessä organisaatiosta toiseen pois kokonaan ja reseptin kirjoittamisessa voitaisiin hyödyntää aikaisemmin potilasrekisteriin merkittyjä lääketietoja. Projektin tavoitteena oli, että sähköinen resepti vastaisi toiminnoiltaan perinteistä paperireseptiä, eikä sen alussa ollut tarkoituskaan korvata normaalia lääkemääräystä. Potilaalla tuli vielä pysyä oikeus valita perinteinen paperinen resepti ja oli erittäin tärkeää saada potilaalta kirjallinen suostumus tietojen välittämiseen tietoverkossa, sillä sähköisessä reseptissä potilaalle jää ainoastaan kirjallinen potilasohje eikä itse reseptiä lainkaan. Pilotoinnin aikana kuitenkin huomattiin suostumuksen toisinaan hidastuttavan lääkärin toimintaa, kun hänen oli ennen reseptin kirjoittamista varmistettava potilaan suostumus sähköiseen reseptiin. Suostumus on siis hyvä pyytää potilaalta jo ennen lääkärin vastaanottoa, esimerkiksi ilmoittautumisen yhteydessä, jolloin reseptistä voidaan kertoa potilaalle paremmin. Mikäli sähköinen resepti syrjäyttää paperireseptin kokonaan, tulee suostumuskäytännöstäkin helpompaa, sillä se on annettava vain kerran ja tähän voidaan vaikuttaa myös muuttamalla lainsäädäntöä. [Hyppönen 2005; Koponen-Piiroinen et al. 2001]

#### **3.1.1 Vaatimuksia ja oletuksia**

Sähköisen reseptin ollessa tarkoitus toimia aivan kuten paperiversiokin, oli olennaista, että potilas voi yhä valita apteekin, josta hän lääkkeensä noutaa. Ei myöskään voida tällöin olettaa, että potilas tarvitsisi uusia teknisiä välineitä sähköistä reseptiä käytettäessä. Mikäli kansalaisvarmenteet ja HST-kortti (Henkilön Sähköinen Tunnistaminen) yleistyvät, voidaan näitä toki hyödyntää myöhemmin myös sähköisen reseptin käytössä. Projekti pyrki kuitenkin rakentamaan sähköisen reseptijärjestelmän käyttäen olemassa olevia tekniikoita

ja kehittämällä niitä. Lisäkustannuksia uusien tekniikoiden tai sovellusten muodossa pyrittiin välttämään sekä esiselvityksessä että pilotoinnissa. Vaatimuksista ja oletuksista huolimatta järjestelmän tuli olla aukoton, tehokas, turvallinen ja joustava, ja järjestelmä pyrittiin kehittämään valtakunnallisesti yhtenäiseksi ja selkeäksi. Usean eri järjestelmän käyttäminen saattaisi johtaa muun muassa apteekkien vaikeuksiin noutaa reseptejä ja Kansaneläkelaitoksen korvauksien maksamisen hidastumiseen ja hankaloitumiseen. Tätä ei tietenkään voitu sallia, sillä järjestelmän oli tarkoitus helpottaa reseptien määräämistä ja hallintaa, lääkkeiden toimittamista sekä ylläpitää potilaiden oikeuksia. Turvallisuus oli yhtä lailla yksi olennaisimmista kriteereistä, sillä resepteissä liikkuu potilaan kannalta arkaluontoisia tietoja, kuten henkilötunnus. Muun muassa tämän takia projektin esiselvitys oli sitä mieltä, että järjestelmän hallinnoinnin tulisi olla virkamiesluontoista. Lääkkeiden myyntitiedot tai tietyn henkilön ostamat lääkkeet saattaisivat olla arvokasta tietoa väärissä käsissä, joten myös kaupallinen hyöty reseptitiedoista tulisi estää. [Koponen-Piironen et al. 2001]

Niin kuin jo aiemmin mainittiin, sähköinen resepti perustuu vahvasti sen paperiseen esikuvaan. Myös tämä piirre toi suunnitteluun haasteita ja lisävaateita. Ensinnäkin 1970-luvulta asti voimassa ollut sairausvakuutuslain mukainen suorakorvausmenetelmä tulisi olla mahdollista myös sähköisessä reseptissä, sillä sen puuttuminen pakottaisi potilaat hakemaan korvauksia suoraan Kansaneläkelaitokselta, mikä taas olisi sähköisen reseptin tavoitteiden vastaista. Lisäksi sähköisen reseptin tulisi mahdollistaa reseptin uusimien, mikä paperiversiossa voidaan tehdä kolmesti. Sähköistä reseptiä ei sinällään kuitenkaan voida uusia, sillä kerran kirjoitettuun reseptiin ei voida tehdä muutoksia, vaan uusi muutettu resepti korvaa aina vanhemman. Käytännössä reseptin uusiminen on silti mahdollista, sillä lääkäri voi hakea vanhan reseptin uuden lääkemääräyksen pohjaksi ja tallentaa tämän sitten uutena reseptinä. Tällöin potilas saa uuden määräyksen lääkkeestä ja voi noutaa sen haluamastaan apteekista, ja reseptitietokannassa näkyvät sekä alkuperäinen resepti että niin sanottu uusittu resepti. Samalla periaatteella voidaan määrätä useampaa lääkettä yhtä aikaa, sillä vaikka paperireseptillä voidaan määrätä kahta lääkevalmistetta

samanaikaisesti, voi yhdellä sähköisellä reseptillä määrätä vain yhtä lääketta. [Koponen-Piironen et al. 2001]

Myös lääkkeen määrääjälle ja potilaalle joudutaan asettamaan vaateita. Aivan kuten paperireseptiäkään, ei sähköistä reseptiä voi kirjoittaa kuka tahansa. Lääkkeitä voivat Suomessa määrätä laillistetut lääkärit, hammaslääkärit, eläinlääkärit ja eläinlääketieteen kandidaatit sekä lääketieteen kandidaatit ollessaan tietyssä virassa (ja vain hoitamilleen potilaille) ja toimiluvan saaneet ulkomaalaiset lääkärit ja hammaslääkärit. Terveystuollon oikeusturvakeskus sekä maa- ja metsäministeriö pitävät yllä tietokantoja, jotka sisältävät tiedot kaikista laillistetuista lääkäreistä, hammaslääkäreistä ja eläinlääkäreistä. Tästä tietokannasta voidaan ennen reseptijärjestelmän käyttöä luotettavasti ja sähköisesti tarkastaa, että lääkkeen määrääjällä on siihen oikeus. Myös potilaan tunnistaminen on olennaista, sillä lääkemääräykset ovat henkilökohtaisia. Sähköinen resepti voidaan kirjoittaa potilaalle, jolla on Suomessa annettu henkilötunnus. Suurimmassa osassa eri toimipisteitä reseptijärjestelmä ja potilasrekisteri toimivat yhdessä, joten henkilötunnuksen lisäämisen reseptiin ei aiheuta erityistoimia. Ainoastaan vastasyntyneille lapsille ja ulkomaalaisille joudutaan kirjoittamaan paperinen resepti, sillä heillä ei ole (vielä) henkilötunnusta. [Koponen-Piironen et al. 2001]

### **3.1.2 Keskitetty tietokantamalli**

Suomessa erilaisista teknisistä malleista valittiin käyttöön ja pilotointiin keskitetty tietokantaratkaisu. Siinä kaikki kirjoitetut reseptit allekirjoitetaan ensin sähköisesti ja sen jälkeen talletetaan valtakunnalliseen tietokantaan. Tietokantaa voivat käyttää lääkärit, apteekit ja Kansaneläkelaitos. Lääkärit siis voivat määrätä lääkkeen, kirjoittaa reseptin, allekirjoittaa sen ja lähettää sen tietokantaan. Tämän jälkeen potilaan valitsema apteekki voi sähköisen tunnistautumisen jälkeen hakea reseptin tietokannasta ja toimittaa lääkkeen potilaalle. Toimittamisesta jää tietokantaan merkintä ja Kansaneläkelaitos voi sen perusteella suorittaa sairausvakuutuskorvauksen. Tässä ratkaisussa jokaisella toimijalla on oltava standardoitu reseptin tietosisältö, rajapinta ja sähköinen allekirjoitus, joten usean eri

järjestelmän ei pitäisi aiheuttaa vaikeuksia. Potilaan oikeus valita haluamansa apteekki säilyy, kun resepti voidaan noutaa yhteisestä tietokannasta, ja samalla mahdollistuvat lääkkeen osatoimitukset. Potilaan turvallisuus parantuu lisäksi, jos lääkäri voi katsoa tietokannasta mitä lääkkeitä potilaalla tällä hetkellä on, tai mitä hänellä on aikaisemmin ollut. Yhteistä tietokantaa käyttämällä voidaan lisäksi estää reseptien monistaminen, sillä jokaisesta yhteydestä ja muutoksesta tietokantaan jää merkintä. [Hyppönen 2005; Koponen-Piironen et al. 2001]

Mahdollisena ongelmana ratkaisua valitessa nähtiin järjestelmän raskas ylläpidettävyys, sillä pelkästään vuonna 2005 reseptejä määrättiin miltei 27 miljoonaa kappaletta ja sähköisten reseptijärjestelmien yleistyessä ne kaikki on mahdutettava tietokantaan. Lisäksi reseptitietokannan ylläpitäjällä on vastuu kaikista tietojen luovuttamisista ja kaikista sähköiseen reseptiin liittyvistä toiminnoista, kuten potilaan aikaisempien reseptien tarkastelusta lääkkeen määräämistä varten. Ongelmista huolimatta ratkaisu nähtiin Suomelle parhaana ja toimivimpana vaihtoehtona ja Kansaneläkelaitos valittiin ylläpitämään reseptitietokantaa. Tietokanta koostuu XML-muotoisista (eXtensible Markup Language) resepteistä eli sanomatiedostoista ja DB2-muodossa (IBM:n tietokantaohjelmiston muoto, DataBase2) olevista indeksitiedostoista. Lisäksi tietokannassa on lokitiedosto, johon tulevat merkinnät kaikista tietokannan sanomista. [Hyppönen 2005; Koponen-Piironen et al. 2001]

Eri terveydenhuolto-organisaatiot käyttävät eri tietojärjestelmiä, mutta sähköisen reseptin on mahdollistettava myös lääkkeen määrääminen toisella paikkakunnalla kuin sen noutaminen. Lisäksi tiedonvälitys organisaatioiden välillä tapahtuu Internetin yli, joten tarvitaan jokin turvallinen ja alustariippumaton kommunikaatiomekanismi. Tästä johtuen kaikki sanomaliikenne organisaatioiden ja tietokannan välillä on XML-muotoista ja kommunikaatiokanavana toimii SOAP (Simple Object Access Protocol), joka avulla asiakkaat ja palvelut voivat vaihtaa XML-viestejä. Reseptien tietosisältö, eli sanomiin XML-muotoon muutettava tieto sisältää potilasta koskevat tiedot, lääketiedot, lisätiedot, annostuksen, lääkärin tiedot, lääkärinpalkkion, apteekin korjausmerkinnät ja

toimittamismerkinnät. Jokainen nämä tiedot sisältävä resepti on alkuperäinen resepti ja sisältää vain yhden lääkkeen. [Hyppönen 2005; Koponen-Piiroinen et al. 2001]

### **3.1.3 Sähköisen reseptin käsittely**

Potilaan tullessa lääkärin vastaanotolle, hän ilmoittautuu potilastoimistossa, jossa häneltä pyydetään kirjallinen suostumus sähköisen reseptijärjestelmän käyttämiselle (kuva 2, 1). Tämä tarkoittaa, että potilas hyväksyy hänen tietojensa ja lääkemääräyksen tietojen kulkevan sähköisesti Kansaneläkelaitoksen tietokantaan, josta apteekki voi ne noutaa lääkettä toimittaessaan ja jonka mukaan Kansaneläkelaitos voi myöntää sairausvakuutuslain mukaisen korvauksen. Potilas voi myös antaa erillisen suostumuksensa lääkärille, jotta tämä voi tarkastaa tietokannasta potilaan aikaisempia lääkemääräyksiä hoidon niin vaatiessa. Molemmat suostumukset talletetaan vastaanotolla ja niistä lisätään merkintä potilasrekisteriin. Potilas voi milloin tahansa niin halutessaan peruuttaa suostumuksensa tai pyytää paperisen reseptin. [Koponen-Piiroinen et al. 2001]

Lääkärin päätettyä hoidosta, hän kirjoittaa potilaalle reseptin, jossa hän voi hyödyntää tietokannassa olevia aiempia reseptejä. Resepti kirjoitetaan potilasrekisteriin liitettyllä reseptijärjestelmällä tai erillisellä ohjelmistolla, joka on liitetty lääketietokantaan lääkkeen valitsemisen helpottamiseksi (kuva 2, 2). Kaikki sähköisiä reseptejä tuottavat järjestelmät muodostavat reseptin, joka on rakenteeltaan, tietosisällöltään ja koodaukseltaan yhtenäinen, jotta sekaannuksilta eri järjestelmien ja organisaatioiden välillä vältytään. Lääkäri vahvistaa reseptin EY-direktiivin (1999/93/EY) mukaisella sähköisellä allekirjoituksella ja lähettää sen reseptitietokantaan salattuna odottamaan lääkkeen toimittamista (kuva 2, 4). Lääkärin oikeus kirjoittaa reseptiä tarkastetaan reseptitietokannan käyttöoikeusrekisteristä, jotta vain laillistetut lääkärit, hammaslääkärit ja eläinlääkärit voivat määrätä lääkkeitä. Reseptitietokanta ilmoittaa lääkärille tallennuksen onnistuneen ja reseptinkirjoitusohjelmasta tulostetaan potilaalle potilasohje (kuva 2, 3). Tämä vastaa sisällöltään paperista reseptiä, mutta on ulkoasultaan kuitenkin erilainen eikä sillä pelkästään saa lääkettä apteekista. Pilotin esiselvityksessä todetaan myös, että potilaalla on

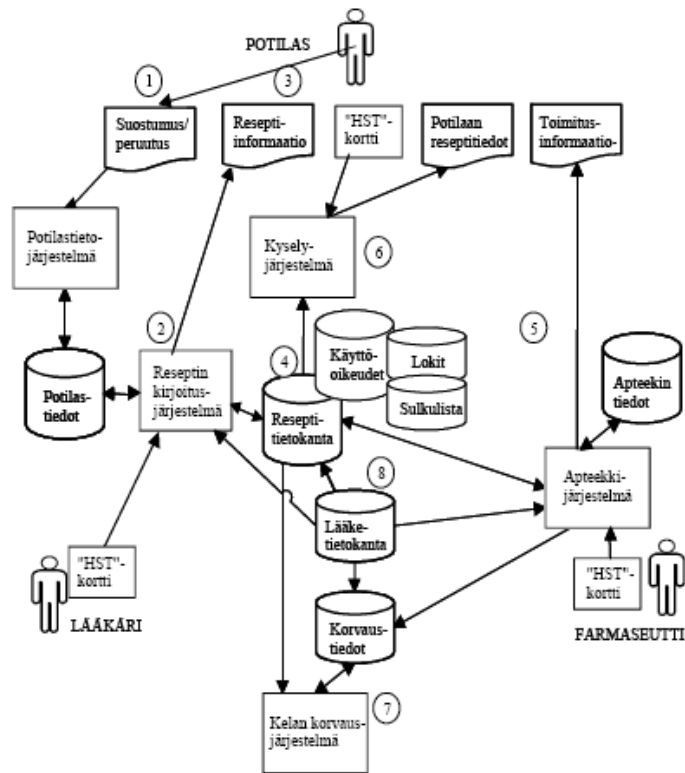


oikeus käyttää reseptitietojen salaamisessa omaa PIN-tunnustaan (Personal Identification Number), jolloin kukaan ei näe reseptin tietoja ilman kyseessä olevaa tunnusta. [Koponen-Piironen et al. 2001]

Aivan kuin paperireseptiä käytettäessäkin, potilas voi valita haluamansa apteekin ja noutaa lääkkeensä sieltä. Apteekissa potilaalta pyydetään Kela-kortti, jolla potilas tunnistetaan ja jolla varmistetaan oikeus sairausvakuutuslain mukaiseen korvaukseen. Farmaseutti käyttää sähköistä tunnistusta kirjautuessaan järjestelmään, jolloin hänen oikeutensa hakea reseptejä tietokannasta tarkastetaan, aivan niin kuin lääkärinkin kohdalla (kuva 2, 5). Tunnistautumisen jälkeen farmaseutti voi hakea potilaan reseptin tai mahdollisesti useita reseptejä, jos potilas ei osaa yksilöidä tiettyä reseptiä. Mikäli potilas on halunnut salata reseptin lääkärin vastaanotolla omalla PIN-tunnuksellaan, pyydetään häntä avaamaan salaus samalla tunnuksella. Reseptin toimitustiedot tallennetaan reseptitietokantaan farmaseutin allekirjoitettua ne sähköisesti. Potilaalle toimitetaan lääke sekä haluttaessa kirjallinen informaatio toimituksesta. Reseptin voi noutaa potilaan puolesta myös valtuutettu. Tällöin on hyvä, jos valtuutetulla on mukanaan lääkkeen potilasohje ja potilaan Kela-kortti, jotta lääke ja potilaan omavastuu hinta voidaan kohdistaa oikein. Myös apteekissa voidaan antaa kirjallinen suostumus muiden reseptitietojen tarkasteluun. Tämä voi auttaa lääkkeiden käytön rationalisoinnissa ja parantaa lääketurvallisuutta, kun myös farmaseutti tai proviisori voi tarkastaa päällekkäisen lääkityksen yhteisvaikutuksia. [Koponen-Piironen et al. 2001]

Niin halutessaan potilaalla on aina oikeus tarkastaa omat reseptitietonsa rekisterinpitäjän luona, eli Kansaneläkelaitoksella (kuva 2, 6). Projektin alussa apteekit toimittavat suorakorvauksista Kansaneläkelaitokselle tilityksen vanhan tavan mukaisesti, ja tämä voi tarkastaa reseptin alkuperäisyyden reseptitietokannasta (kuva 2, 7). Tavoitteena kuitenkin on, että reseptijärjestelmän yleistyessä Kansaneläkelaitos voi tarkastaa suorakorvausmenetelmällä potilaalle korvatut reseptit suoraan tietokannasta ja maksaa korvaukset kullekin apteekille ilman erillistä tilitystä. Paperiresepteillä haetuista lääkkeistä apteekki joutuu joka tapauksessa tekemään normaalin kirjallisen tilityksen ja potilaan

hakiessa korvausta suoraan Kansaneläkelaitoksen konttorista, saa hän apteekista selvityksen ja kuitin sähköisen reseptin käytöstä ja lääkeostosta. Yhtä lailla potilas voi saada tarkan erittelyn ostoistaan esimerkiksi vakuutusyhtiötä varten. [Koponen-Piironen et al. 2001]



Kuva 2 Toimintakaavio sähköisen reseptin käsittelystä [Koponen-Piironen et al. 2001]

Kaiken toiminnan takana sekä lääkäreillä että apteekilla on lääketietokanta, jossa pidetään ajantasaiset tiedot kaikista Suomessa myynnissä olevista lääkkeistä. Lääkäri näkee lääketietokannasta valittavat lääkkeet, niiden annoskoot ja vahvuudet, mikä helpottaa lääkkeen ja pakkauksen valitsemista potilaalle. Tämä helpottaa myös apteekin toimintaa, kun heidän ei tarvitse varmistaa pakkaukset erikseen lääkäriltä. Sähköisen reseptin yhteydessä lääketietokannasta nähdään lääkevalmisteen nimi, tunniste, vahvuus, lääkemuoto, pakkaukset, lääkkeen status, myyntiluvan sisältämät ehdot ja lääkkeen korvaustiedot. Myöhemmin olisi hyödyllistä saada lääketietokantaan myös lääkkeiden hinnat helpottamaan lääkkeen valintaa ja potilasta. Yhteisen lääketietokannan käyttö on

merkittävää toiminnan joustavuuden ja potilasturvallisuuden kannalta. Poikkeavien tietojen kohdalla apteekin on aina otettava yhteys valmisteen määränneeseen lääkäriin, mikä viivästyttää lääkkeen toimittamista potilaalle. Näin ollen lääketietokannan on hyvä olla helposti ja maksuttomasti saatavilla, jotta sen käyttöönotto ja päivittäminen on mahdollisimman tiivistä. [Koponen-Piironen et al. 2001]

Itse reseptitietokanta on valtakunnallinen ja keskitetty, ja siitä vastaa Kansaneläkelaitos. Sillä on siis vastuu ylläpitää sekä rekisterissä olevien potilaiden oikeuksia että yleistä tietoturvaa. Tietokannan on oltava erittäin skaalautumiskykyinen, sillä kun järjestelmästä tulee kansallinen, on sen kestävä kymmeniä miljoonia reseptejä vuodessa ja useita yhtäaikaista toimintoja joka sekunti. Kansaneläkelaitoksen onkin varauduttava tuottamaan järjestelmään varmuutta ja jatkuva käyttäjätukea, sillä reseptitietokannan kaatuessa tai yhteyden siihen katketessa, ei reseptejä voida toimittaa lainkaan ilman tällaisen tilanteen varalle suunniteltua varokeinoa. [Koponen-Piironen et al. 2001]

Reseptitietokanta sisältää varsinaisia reseptitietoja, käyttäjä- ja käyttöoikeusrekisterin, lokitiedostot, sulkulistan ja arkiston. Reseptitiedoissa ovat reseptin yksilöllinen tunnistus, potilaan tiedot, tiedot lääkkeistä ja niiden annostuksista, mahdolliset lisätiedot, tiedot reseptin kirjoittajasta, lääkärinpalkkiotiedot (jos lääke on määrätty potilaalle muualla kuin vastaanotolla), toimitustiedot, toimituksen status, korjaustiedot ja muutoshistoria. Toimituksen statuksella merkitään onko lääkemäärä toimitettu kokonaisuudessaan, vai onko potilaalla vielä hakematta siitä osa. Käyttäjä- ja käyttöoikeusrekisteristä tarkastetaan, ketkä saavat kirjoittaa reseptejä, ketkä hakea niitä ja ketkä käsitellä niitä. Rekisterissä on terveydenhuollon oikeusturvakeskuksen ylläpitämä lista laillistetuista lääkäreistä ja hammaslääkäreistä sekä maa- ja metsätalousministeriön ylläpitämä lista eläinlääkäreistä ja eläinlääketieteen kandidaateista. Lisäksi siellä on toimipaikkakohtainen lista apteekin henkilökunnasta sekä rekisterin ylläpitäjän tiedot. Lokitiedostoihin merkitään aina reseptin katselu- ja käsittelytoimista merkintä. Lokissa näkyy kuka toiminnon on tehnyt, milloin se on tehty ja mitä on tehty. Sulkulistalla on tiedot mahdollisista rajoituksista, esimerkiksi

lääkärit, joiden toimilupa on evätty ja arkistoissa säilytetään erityissäädösten perusteella muita reseptejä kauemmin säilytettävät reseptitiedot. [Koponen-Piironen et al. 2001]

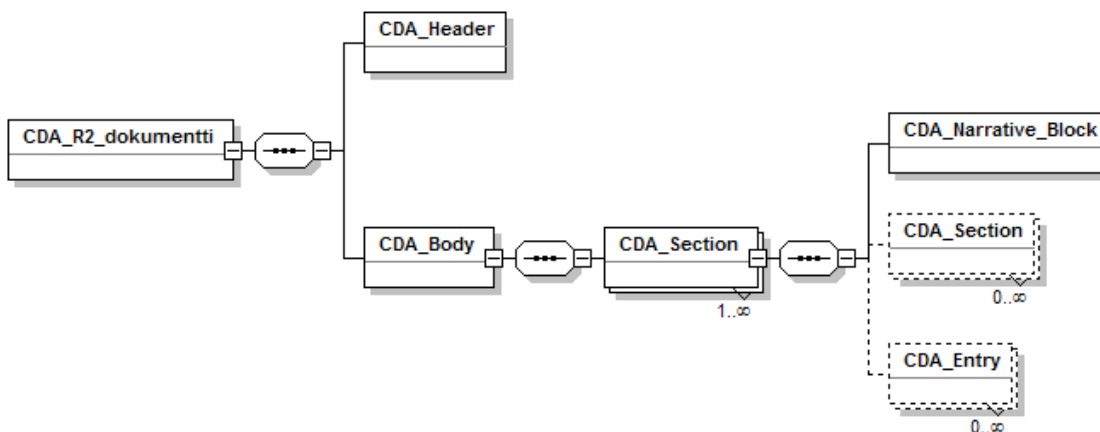
## **3.2 Standardointi sähköisessä reseptissä**

Sähköiselle reseptille ei vielä ole montaakaan standardia, vaikkakin sen mahdollistaville tekniikoille niitä on olemassa. Kansainvälinen HL7 (Health Level Seven) pyrkii tuomaan terveydenhuollon alalle viestienvälitykseen ja eri dokumentteihin standardeja. ISO (International Organization for Standardization) puolestaan standardoi tekniikoita ja tietoturva puolelle on saatu yhteisiä suuntauksia sekä kriteeristöjen että ISO:n standardien avulla. ISO on lisäksi kehittämässä standardia itse sähköiselle reseptille, mutta se on vasta alkuvaiheissa. Suomessa sähköisen reseptin rakentaminen on pyritty tekemään niin, että se vastaa voimassa olevia sekä tulevia standardeja mahdollisuuksien mukaan. Tämä on tietenkin tärkeää ottaen huomioon erilaisten järjestelmien määrän sairaanhoitopiireissä, apteekkeissa ja muissa terveydenhuollon organisaatioissa. Seuraavassa on lyhyt esittely sähköisen reseptin pilotissakin huomioon otetuista standardeista.

### **3.2.1 Health Level 7**

HL7 eli Health Level Seven on terveydenhuollon piirissä toimiva standardointiorganisaatio, jonka pääalaa on kliininen ja hallinnollinen tieto. HL7 tuottamat standardit (tai spesifikaatiot ja protokollat) ovat ANSI:n (American National Standards Institute) valtuuttamia ja itse organisaatio on voittoa tavoittelematon useiden muiden standardointiorganisaatioiden lailla. HL7 on kansainvälinen organisaatio ja sen jäseninä on useita tuottajia, kauppiaita, asiantuntijoita ja hallitusryhmiä, jotka muodostavat teknisiin komiteoihin ja erikoisryhmiin jaetun työryhmän. HL7 on ollut mukana kehittämässä XML:ää vuodesta 1996 osana sen omia standardeja, kuten HL7 V2.3.1 viestit ja CDA (Clinical Document Architecture). Sen tunnetuimmat standardit ovatkin juuri CDA ja viestitysstandardi (The Messaging Standard) sekä RIM (Reference Information Model) ja CCOW (Clinical Context Object Workgroup). [HL7 2006]

Suomessa sähköisen reseptin pilotin esiselvityksessä määritettiin, että reseptinkirjoitusohjelmien tulisi pystyä lähettämään ja käsittelemään XML-muotoisia viestejä ja määrittämään näiden tietosisältö HL7:n CDA määrittelyjen mukaisesti. CDA, joka aiemmin tunnettiin paremmin nimellä PRA (Patient Record Architecture), on ANSI:n standardiksi hyväksymä tiedonsiirtomalli kliinisille dokumenteille. CDA käyttää hyödykseen XML:ää, HL7:n RIM-mallia ja koodattuja sanastoja, jolloin CDA dokumentit ovat sekä ihmisten että koneiden luettavissa. Ne on siis helppo jäsentää, käsitellä sähköisesti ja saada esille. CDA muotoiset dokumentit voidaan näyttää XML:ää ymmärtävissä selaimissa tai jopa langattomissa laitteissa kuten matkapuhelimissa. Uusin versio CDA:sta on vuonna 2005 julkaistu R2 (Release 2.0), jonka dokumentin periaatteellista rakennetta esitetään alla olevassa kuvassa 3. [HL7 2006]



Kuva 3 CDA R2 dokumentin periaatteellinen rakenne [HL7 2006]

CDA määrittelee kliinisten dokumenttien syntaksin sekä viitekehuksesta semantiikan. CDA käyttää XML:ää ja voi sisältää mitä tahansa tietoa, mutta se voi toki käyttää muutakin kuin XML-pohjaista rakennetta. Koska CDA hyödyntää XML:ää, voidaan sitä käyttää miltei millä tahansa sovelluksella tai säilyttää varastossa, joka tukee XML:ää. CDA standardin avulla pyritään lisäämään semanttista yhteensopivuutta niin, että tietty informaatio voidaan

esittää missä tahansa pisteessä. Esimerkiksi sähköisen reseptin tapauksessa eri käyttöjärjestelmässä olevassa reseptinkirjoitusohjelmistossa kirjoitettu resepti voidaan lähettää reseptitietokantaan ja sieltä apteekin järjestelmään, josta se voidaan lukea ongelmitta. CDA on jaettu kolmeen eri tasoon, joiden mukaan säädellään automaattisten prosessien määrää. Taso yksi ei tee mitään oletuksia vaan toimittaa yksinkertaisesti standardin otsikon metadatan ja rungon muodossa, jossa käyttäjät sen ymmärtävät. Taso 2 tarkoittaa, että runko on XML muodossa ja eri lohkot on koodattu. Taso kolme käsittää molempien edellisten tasojen oletukset ja tiettyä koodattua informaatiota lohkoissa. Käytännössä nämä kolme tasoa voivat toimia hyvinkin sekaisin, vaikka jokainen niistä tarjoaakin tavan asettaa koodaukselle tiettyjä rajoja. [HL7 2006]

### **3.2.2 International Organization for Standardization**

International Organization for Standardization (ISO) on 157 maan kansallista standardointijärjestöä käsittävä standardeja tuottava verkosto, jonka toimintaa ohjaa Genevessä Sveitsissä sijaitseva kanslia. Sähköisen reseptin osalta ISO:n standardeista merkittäviä ovat ISO/IEC 17799 (Information technology - Security techniques - Code of practice for information security management), 27799 (Health informatics - Security management in health using ISO/IEC 17799), 13335 (Information technology - Security techniques - Management of information and communications technology security) ja 7816 (Identification cards -- Integrated circuit cards). [ISO 2006]

ISO/IEC 17799:2005 luo yleiset periaatteet organisaation tietoturvan hallinnolle sekä sen ylläpitämiselle ja kehitykselle. Standardi käsittää muun muassa tietoturvapolitiikan muodostamisen, henkilöstöhallinnon tietoturvan, fyysisen ja ympäristön tietoturvan, pääsynhallinnan ja määräysten noudattamisen. Sen avulla voidaan tuottaa organisaatiolle tehokas tietoturvahallinto ja luoda luottamusta sekä organisaation sisällä että sen kanssa asioiville. ISO/DIS 27799 on vasta kehitteillä, mutta se on tarkoitus määrittää terveydenhuollon informatiikkaa käyttämällä hyödyksi standardia 17799. ISO/IEC 13335:n uusin versio on vuodelta 2004 ja se määrittää erilaisia tietoturvatekniikoita, joiden avulla

voidaan ymmärtää ja rakentaa ICT - (Information and Communications Technology) tietoturva organisaatiolle. ISO/IEC 7816 on henkilökortteja ja sirukortteja varten kehitetty standardi, joka määrittää esimerkiksi korttien fyysisiä mittoja, kontaktipintoja, käytettäviä signaaleja ja protokollia sekä tiedontallennusta ja – siirtoa. [ISO 2006]

### **3.2.3 ITSEC ja Common Criteria - kriteeristöt**

Sirukorttien turvallisuusominaisuuksia ei vielä toistaiseksi ole standardoitu, mutta niissä voidaan käyttää hyödyksi kansainvälisiä kriteeristöjä, kuten ITSEC:ä (Information Technology Security Evaluation Criteria) ja Common Criteriaa. ITSEC on Euroopan yhteisön (Ranskan, Saksan Alankomaiden ja Englannin) luoma yhteinen ja nykyisin kansainvälinen kriteeristö tietoturvalle. Se pohjautuu maiden omiin aikaisempiin kriteeristöihin sekä amerikkalaiseen TCSEC:iin (Trusted Computer System Evaluation Criteria, ”Orange Book”). ITSEC on jaettu kolmeen tasoon; turvallisuuspäämäärään, turvallisuutta toimeenpaneviin tehtäviin sekä turvamekanismeihin. Turvallisuuspäämäärä selvittää, miksi tietty turvatoimenpide halutaan, turvatoimenpiteen kertovat mitä toimia on ja turvamekanismit kertovat miten turvatoimenpiteitä käytetään. Kriteeristö kertoo esimerkiksi miten ja millainen tietoturvapoliittikka tulisi laatia, miten tietoturvapoliittikka voidaan käytännössä noudattaa ja miten eri ohjelmistot, dokumentit ja laitteet tulisi valmistaa. [ITSEC 1991]

Common Criteria on kansainväliseen käyttöön kehitetty tietoturvakriteeristö, joka pohjautuu eurooppalaisiin ja pohjoisamerikkalaisiin kriteeristöihin (ITSEC, TCSEC ja CTCPEC) yhtenäistämällä näiden eroja. Sen ensimmäinen versio ilmestyi vuonna 1996 ja vuonna 1998 ilmestynyt toinen versio on nykyisin myös ISO:n standardi (ISO 15408). Common Criterionin pääkäsitteitä ovat arvioinnin kohde TOE (Target of Evaluation), turvaprofiili PP (Protection Profile) ja tietoturvakohde ST (Security Target). TOE on siis se osa tuotteesta tai järjestelmästä, jota arvioidaan ja siihen kohdistuvat uhat, tavoitteet, vaatimukset ja turvallisuustoiminnot muodostavat tietoturvakohteen, jota käytetään arvioinnin pohjana. PP puolestaan on käsite, jonka avulla voidaan luoda toteutuksesta

riippumattomat tietoturva-vaatimukset halutuille tuotteille tai järjestelmille. Tietoturvapolitiikka voidaan muodostaa arvioimalla TOE:ta kriteeristössä olevien komponenttien ja tasojen avulla. [Common Criteria 2006]



## **4 SÄHKÖISEN RESEPTIN TEKNIKOITA**

Sähköinen resepti voidaan toteuttaa usealla eri tavalla, niin tekniikaltaan kuin käytännöltäänkin. Esimerkiksi sähköinen resepti voidaan toimittaa lääkäriltä erilliseen keskusvarastoon, josta apteekki noutaa sen, se voidaan lähettää sähköpostin tyyliin suoraan haluttuun apteekkiin tai se voidaan tallentaa potilaalle mukaan sähköiseen älykorttiin. Koska sähköinen resepti käsittelee potilaan henkilökohtaisia tietoja, jotka useimmissa maissa ovat jollain lailla suojattuja, on siirrettävä tieto pystyttävä pitämään salassa muilta. Tätä varten reseptit on pystyttävä salaamaan ja pitämään muuttumattomina, ja yhteydet, joilla niitä lähetetään, on pyrittävä pitämään koskemattomina ja turvassa ulkopuolisilta. Lisäksi on pidettävä huolta, että vain auktorisoidut henkilöt voivat käsitellä ja lukea reseptejä. Kaikki tämä johtaa siihen, että sähköisen reseptin järjestelmää suunniteltaessa on otettava huomioon yletön määrä eri rajoituksia ja vaateita, ja sovitettava ne yhteen. Jo pelkkä käytettävän tekniikan valinta on vaikea, sillä erilaisia toteutuksia voi olla useita. Seuraavassa on esitelty eräitä ehkä yleisimpiä toteutusmalleja sähköiselle reseptille ja niitä tekniikoita, joita toteutusmalleja varten tarvitaan.

### **4.1 Sähköisen reseptin järjestelmät Suomessa**

Suomessa on eri sairaanhoitopiireillä käytössä eri valmistajien potilastietojärjestelmiä, joihin sähköisen reseptin järjestelmät on tarkoitus liittää. Sähköisen reseptin pilotissa toimineilla sairaanhoitopiireillä on jokaisella eri järjestelmä, ja lisäksi apteekeilla on kaksi omaa järjestelmää. Helsingin ja Uudenmaan sairaanhoitopiirissä on käytössä Mediwebin selainpohjainen Elres 4.1 – järjestelmä, Kymenlaaksossa käytetään TietoEnatorin suunnittelemaa Efficia – järjestelmää, Turussa toimii Novo Groupin Pegasos-järjestelmä ja Joensuussa Doctorexin HealthNet. Apteekeissa ovat käytössä Linnea- ja Salix-järjestelmät. Haasteena pilotille ja sähköisen reseptin kansalliselle toiminnalle ovatkin erilaiset käytössä olevat järjestelmät ja niiden integroiminen yhteensopiviksi apteekkien järjestelmien kanssa. Toiminnan kannalta on oleellista, että olipa kyseessä mikä tahansa terveydenhuollon

organisaatio ja mikä tahansa sen käyttämä järjestelmä, reseptit toimitetaan varmasti ja turvallisesti Kansaneläkelaitoksen reseptitietokantaan ja sieltä apteekkiin.

#### **4.1.1 Elres**

Elres (Elektroninen Resepti) on suomalaisen vuonna 1996 perustetun Mediweb Oy:n tuottama sähköisten reseptien kirjoittamiseen tarkoitettu sovellus, jolla voidaan kirjoittaa, lähettää ja toimittaa määrätyt lääkkeet turvallisesti. Järjestelmän, niin kuin sähköisen reseptin yleensäkin, on tarkoitus estää kirjoitusvirheistä aiheutuvat toimitusvirheet, reseptien väärentäminen ja muut väärinkäytökset. Keskitetyn järjestelmän ansiosta päästään näkemään potilaiden aikaisempi lääkitys ja toimitus sujuu helposti, vaikka lääkkeet haettaisiin osissa tai resepti uusittaisiin. Myös elres käyttää lääkäreiden ja farmaseuttien tunnistamiseen älykorttia tai vastaavaa turvatokenia ja kaikki toimitukset allekirjoitetaan sähköisesti. Lisänä elresin paikallinen sähköinen potilastietojärjestelmä voi vastaanottaa ilmoituksia toimitetuista lääkkeistä, mikäli lääkärit haluavat valvoa tätä. [Elres 2006]

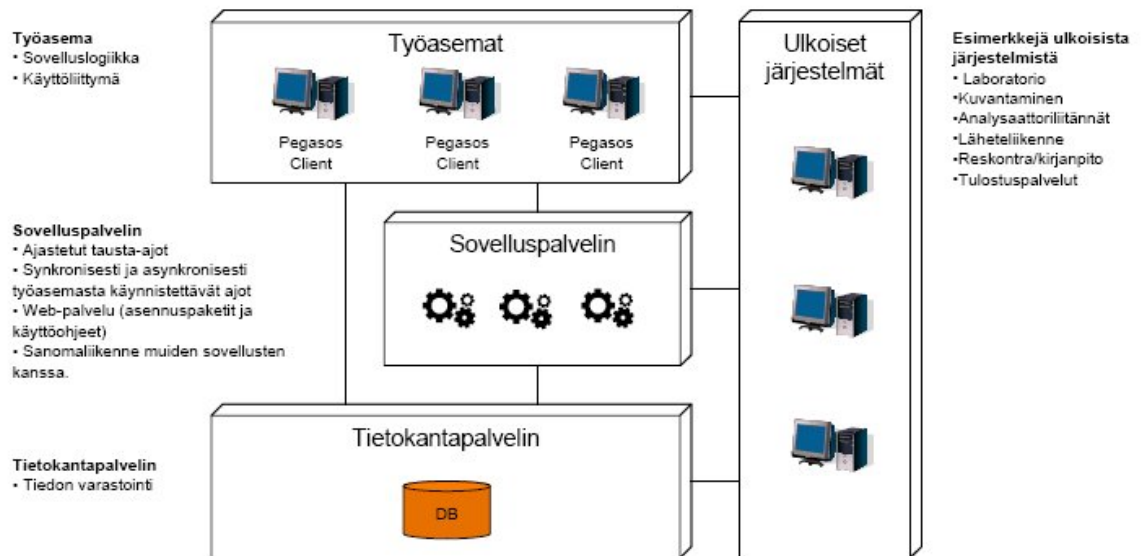
#### **4.1.2 Effica eResepti**

Effica on myös potilastietojärjestelmä, joka on tarkoitettu sairaalakäyttöön. Koska Effica on ollut käytössä jo ennen sähköisen reseptin pilottia, on siihen toteutettu erillisenä osana eResepti-niminen sähköisen reseptin kirjoitusjärjestelmä. Kirjoitusjärjestelmään päästään potilaskertomuksen reseptilomakkeelta ja vastaavasti lääkityskyselyjä voidaan tehdä potilaskertomuksen lääkelehdeltä. EResepti-ominaisuutta voidaan käyttää ainoastaan potilaskertomuksen kanssa ja potilas, jolle sähköinen resepti halutaan kirjoittaa, on oltava valittuna eReseptille pääsemiseksi. Ennen jokaista reseptin kirjoitusta tai lääkityshistorian tarkastamista järjestelmä kysyy potilaan suostumuksen. Näin eri lääkärit voivat varmistaa, että potilas on antanut suostumuksensa sähköisen reseptin käyttöön, tai he voivat pyytää suostumuksen mikäli sitä ei ole aiemmin annettu. [TietoEnator 2006]

Muutenkin Effican eResepti näyttää toimivan aivan pilotin esiselvityksen määräämällä tavalla. Jokainen määrätty lääke tulee omalle reseptilleen ja jokainen niistä on allekirjoitettava erikseen. Allekirjoitus tapahtuu syöttämällä lääkärin HST-kortti (Henkilön Sähköinen Tunnistaminen) järjestelmän lukijaan ja syöttämällä lääkärin ja kortin PIN-tunnus. Vaikka jokainen lääkemääräys on allekirjoitettava erikseen, ne voidaan kuitenkin lähettää reseptipalvelimelle yhtä aikaa. Reseptin lähetyksen jälkeen näytölle ilmestyy lähetettyjen reseptien tilanne, josta voidaan tarkastaa ovatko reseptit lähteneet palvelimelle. Mikäli lähetyksessä on tapahtunut virhe, resepti voidaan lähettää uudelleen ja mikäli lähetyksessä on toiminut moitteetta, voidaan jokaisesta reseptistä (eli lääkkeestä) tulostaa potilaalle potilasohje. [TietoEnator 2006]

#### **4.1.3 Pegasos**

Pegasos on sosiaali- ja terveystietojärjestelmä, joka on tarkoitettu sairaaloille, terveyskeskuksille ja sosiaalitoimen yksiköille. Se toimii osissa niin, että eri organisaatiot voivat ottaa käyttöön vain tarvitsemansa moduulit ja laajentaa järjestelmää tarvittaessa. Tietojen tallennus tapahtuu keskitettyyn tietokantaan, jossa tiedot pidetään turvassa ja muuttumattomina tietokantaohjelmiston avulla. Järjestelmään kuuluvat muiden muassa asiakas- ja potilaskertomus, ajanvaraus, laskutus, tilastointi, laboratorio- ja röntgenjärjestelmä, e-posti, raportointi sekä sähköinen asiointi. Asiakas- ja potilaskertomuksissa voidaan käyttäjien oikeuksia rajata niin, että ne ovat tietosuojalain mukaisia ja laboratorio- ja röntgenjärjestelmä on rakennettu niin, että tarvittaessa voidaan ottaa käyttöön HL7-yhteys muihin laitoksiin ja näin välittää tuloksia tai pyyntöjä myös organisaation ulkopuolelle. [Pegasos 2005]



Kuva 4 Pegasos-järjestelmän arkkitehtuuri [Kilpivuori 2006]

Pegasos on käytännössä perinteinen asiakas-palvelin –ratkaisu, jossa työasemaohjelmistoon voidaan liittää useita erilaisia yhteensopivia moduuleita. Tällaisia ovat muun muassa edellä mainitut potilaskertomus, osastohoito ja laboratoriojärjestelmä. Järjestelmä on pyritty rakentamaan niin, että se on käyttäjän näkökulmasta hyvin toimiva useiden eri toimintojen kokonaisuus. Järjestelmää voidaan käyttää hyödyksi muuhunkin kuin vain paperisen potilaskertomuksen korvaamiseen; se pystyy hallitsemaan rakenteellisen tiedon sekä valtavan määrän ohjattuja toimintoja ja sitä voidaan käyttää toiminnan ohjauksessa sekä asiantuntijajärjestelmänä. Kuvassa 4 havainnollistetaan Pegasos-järjestelmän arkkitehtuuria. Sanomaliikenteen ja ajastettujen toimintojen toiminnasta vastaa sovelluspalvelin, johon voidaan työasemien lisäksi liittää myös ulkoisia järjestelmiä. Lisäksi työasemat voidaan liittää ulkoisiin palveluihin, kuten esimerkiksi digisanelulaitteisiin. Pegasoksen tietokantaa käytetään tietojen varastointiin. Siihen ovat yhteydessä sekä työasemat että sovelluspalvelin ja ulkoiset järjestelmät. [Kilpivuori 2006]

Sähköisen reseptin pilotin yhteydessä toteutettuihin järjestelmiin saatiin sanomaliikenne-rajapinnan määrittely ja reseptikeskuksen toiminnallisuus MediWeb Oy:ltä, joka oli projektin puolesta valittu tuottamaan nämä. Myös pilotin selainjärjestelmä oli MediWebiltä, ja tämä tarjosikin selainpohjaista tallennusta myös muiden pilottien järjestelmien käyttöön. VM-Datalla kehitettiin kuitenkin sanomaliikenne-rajapinta Pegasoksen reseptien tallennuksen yhteyteen, sillä lääkärin olisi joka tapauksessa kirjattava tiedot potilasrekisteriin ja selainpohjaisesta tallentamisesta olisi aiheutunut lisätyötä. Näin ollen sähköisen reseptin kirjoittaminen Pegasoksella on yhä miltei samanlaista kuin paperisen atk-reseptin kirjoittaminen. Ainoa ero on, että reseptin tulostamisen tilalla resepti lähetetään reseptitietokantaan sähköisesti, jolloin sovellus kysyy lääkärin PIN-koodia (salainen avain) sähköistä allekirjoitusta varten. Reseptin tilalle tulostuu potilasohje, joka annetaan potilaalle mukaan, ja jota lääkärin ei tarvitse allekirjoittaa. [Kilpivuori 2006]

Potilaan PIN-koodia ei Pegasoksen reseptinkirjoitusjärjestelmään ainakaan pilottivaiheessa panostettu. VM-data totesi, ettei se toisi kokonaistavoitteen kannalta mitään uutta, eikä sitä kannattanut muutaman ääreistapauksen takia toteuttaa. Tulevaisuudessa asiaa pohditaan uudestaan, mikäli se tulee ajankohtaiseksi. VM-Datan Pegasos-järjestelmää ja sen reseptinkirjoitusosaa on testattu useaan otteeseen, viimeksi keväällä 2006. Sovellukset ovat täysin valmiita toimintaan, mutta reseptikeskuksen ja apteekkijärjestelmien ongelmien takia sitä ei ole päästy kokeilemaan kokonaisvaltaisesti. Sähköisen reseptin projektin jatkumisen siirryttyä Kansaneläkelaitoksen ja Accenturen hoiviin toivotaan, että järjestelmää päästään piakkoin koettelemaan tositoimissa. [Kilpivuori 2006]

#### **4.1.4 HealthNet**

HealthNet on Doctorexin kehittämä ja nykyisin TietoEnatorin ylläpitämä erikoissairaanhoidon potilaskertomusjärjestelmä, jossa on lähete-palautetoiminnallisuus. Sähköisen reseptin pilotissa järjestelmä on käytössä Pohjois-Karjalan sairaanhoitopiirissä, jossa Windows 2000/XP-työasemille on asennettu win32 client-sovellus. Verkossa on erillinen tietokantapalvelin, johon työasemat tallentavat tietonsa. Pilotissa jokaisella

HealthNetin käyttäjällä tuli olla lääkärin oikeudet ja oikeus lähettää reseptejä. Lisäksi reseptin kirjoitusta varten potilaalta tuli olla suostumus, joka oli kirjattu HealthNetiin. Lääkkeet voitiin valita PharmaPointin tuottamasta lääketietokannasta, josta lisättiin reseptiin kaikki lääkkeen tarvittavat tiedot. Reseptin muuttamiseen XML-muotoon sekä allekirjoitukseen käytettiin Avain Technologiesin allekirjoituspalvelinta, joka myös hoiti SOAP-kuoren muodostamisen ja reseptin lähettämisen reseptitietokantaan. [Multanen 2006]

Järjestelmä toimintaperiaate on seuraavanlainen. Ensin potilaskertomusjärjestelmästä valitaan potilas, jolle resepti halutaan kirjoittaa sekä reseptinkirjoitusosio. Lääke valitaan Pharmapointin lääketietojärjestelmästä ja lisätään tälle vahvuus, määrä ja Signatum. Potilaalta saatu suostumus näkyy merkittynä resepti-ikkunassa ja kun resepti on valmis, se voidaan lähettää painikkeella. Sähköistä allekirjoitusta varten lääkäri syöttää allekirjoitusikkunaan oman PIN-koodinsa ja resepti lähetetään paikalliselle allekirjoituspalvelimelle (XSign). Allekirjoituksen oikeellisuus tarkastetaan palvelimella, jonka jälkeen resepti lähetetään Kansaneläkelaitoksen reseptitietokantaan ja lääkärille lähetetään kiittäus onnistuneesta tai epäonnistuneesta sanomasta. Mikäli lähetys onnistuu, tulostetaan automaattisesti potilasohje. [Tikkanen 2004 s. 4-11]

#### **4.1.5 Salix ja Linnea**

Salix ja Linnea ovat apteekkeissa käytössä olevia järjestelmiä, jotka on tarkoitettu reseptinkäsittelyyn. Salix on Pharmadata Oy:n toimittama tietojärjestelmä, jolla voidaan hoitaa muun muassa reseptinkäsittely, varastonvalvonta, kassa toiminnot ja laskutus. Pharmadatan omistaa Suomen apteekkariliitto, joten järjestelmä toimii sen esittämien vaatimusten mukaan. Linnea puolestaan on tietotekniikkatalo Receptum Oy:n suunnittelema järjestelmä. Siinä on valittavana erilaisia tuotteita, kuten apteekkiohjelma, varastonvalvonta, kassajärjestelmä ja varmennepalvelu, aivan kuten Salixissakin. [Salix 2001; Linnea 2006]

## 4.2 XML - metakieli

Extensible Markup Language, eli XML, on yksinkertainen ja joustava merkkauškieli, joka alun perin tarkoitettiin suurimittaiseen sähköiseen julkaisutoimintaan, mutta toimii nykyisin myös tärkeässä roolissa kommunikoinnissa verkon välityksellä. XML on johdettu SGML-kielestä (Standard Generalized Markup Language, ISO 8879), jolla järjestetään ja merkataan elementtejä dokumenteissa ja se on metakieli, jonka avulla kuvataan tietoa tiedosta. Sillä siis kerrotaan, mitä tietoa on tallennettu, ei miten se esitetään. Toisin sanoen dokumentin sisältö ja ulkoasu on erotettu toisistaan. Ulkoasukin voidaan erikseen määritellä, mutta siihen käytetään tyylikieliä. Näitä ovat HTML:stä (HyperText Markup Language) tuttu CSS (Cascading Style Sheet) sekä XML:n oma XSL (eXtensible Style Language). [Bray et al. 2004; Tuikka & Kanala 2001 s.3-17]

XML dokumenttia voidaan käsitellä vaikkapa tekstieditorissa, aivan niin kuin HTML dokumenttiakin. Tunnisteet eli tagit voidaan nimetä itse, ja kuka tahansa pystyykin suhteellisen helposti lukemaan XML dokumenttia, jos tagit ovat järkevästi ja kuvailevasti nimettyjä. Helppolukuisuudesta huolimatta XML rakentuu niin, että se on myös koneen ymmärtämässä muodossa. Jokainen XML dokumentti voidaan jakaa rakenteeltaan kahteen osaan, loogiseen ja fyysiseen. XML dokumentin sisältö koostuu erilaisista elementeistä, ja looginen rakenne määrittelee nämä elementit, niiden järjestyksen ja hierarkian. Looginen rakenne edellyttää dokumentin olevan hyvin muodostettu (well-formed), mikä tarkoittaa, että sen on oltava XML-kieliopin mukainen ja täytettävä XML-määrittelyn (W3C Recommendation) mukaiset ehdot. Lisäksi sekä loogisen että fyysisen rakenteen on oltava sisennetty oikein. Fyysisesti dokumentin rakenne koostuu pienistä yksiköistä, eli entiteeteistä (entity), jotka yksilöidään nimellä ja joilla kaikilla on jotain sisältöä. [Bray et al. 2004; Tuikka & Kanala 2001 s.3-17]

XML rakentuu siis elementeistä, attribuuteista ja entiteeteistä. Dokumentit alkavat juuri-elementeistä ja kaikki muut elementit ovat tämän sisällä tasapainossa. Jokaisella elementillä on alkutunniste `<elementti>` ja lopputunniste `</elementti>` ja näiden sisälle merkitään elementtien sisältö. Mikäli elementillä ei ole sisältöä, se on tyhjä, ja voidaan

merkitä joko alku- ja lopputunnisteella tai tyhjän elementin tunnisteella <elementti/>. Kun elementeille halutaan antaa lisätietoa, käytetään attribuutteja. Attribuutit merkitään alkutunnisteen sisään ja niiden arvo lisätään nimen perään lainausmerkeissä (<elementti attribuutti= "arvo">), sillä yhdellä attribuutilla voi olla useampi arvo. Attribuutit ovat aina nimi-arvo-pareja ja useat niistä voivat toimia myös omina elementteinä, joten niiden käyttö onkin täysin käyttäjän valittavissa. Suurissa dokumenteissa attribuuttien käyttö voi olla käytännöllistä, sillä ne säästävät tilaa vähentämällä merkkien määrää. Seuraavassa on esimerkki samasta tiedosta ensin attribuutteja käyttämällä ja sitten elementtien avulla. [Antoniou & Harmelen 2004 s. 23-26; Bray et al. 2004; Tuikka & Kanala 2001 s.3-17]

```
<resepti reseptinro="007" potilas="James Bond" pvm="7.7.2006">
  <laake laakenro="001" kappale="10"/>
  <laake laakenro="002" kappale="20"/>
</resepti>
```

```
<resepti>
  <reseptinro>007</reseptinro>
  <potilas>James Bond</potilas>
  <pvm>7.7.2006</pvm>
  <laake>
    <laakenro>001</laakenro>
    <kpl>10</kpl>
  </laake>
  <laake>
    <laakenro>002</laakenro>
    <kpl>20</kpl>
  </laake>
</resepti>
```

Entiteetit muodistavat XML dokumentin fyysisen rakenteen. Ne määritellään erikseen käyttämällä DTD- (Document Type Definition) tai XML Schema- määrittelyä ja kuvaavat muun muassa eri dokumentin osia, kuvia tai symboleja. XML dokumentissa ei voida



käyttää erikoismerkkejä, kuten skandinaavisia ä-, ö- ja å-kirjaimia tai kreikkalaisia kirjaimia. Nämä erikoismerkit voidaan muodostaa DTD:ssä määriteltyjen entiteettien avulla. DTD dokumenteissa määritetään myös koko XML dokumentin rakenne. Nykyisin DTD dokumentteja korvaamaan on kehitetty uudempi lähestymistapa XML Schema, joka pikkuhiljaa syrjäyttää DTD:t. [Antoniou & Harmelen 2004 s. 23-26; Bray et al. 2004; Tuikka & Kanala 2001 s.3-17]

Jokainen XML dokumentti alkaa käytössä olevan XML-version kertovalla esittelyosalla `<?xml version="1.1"?>`. Samassa voidaan mainita käytössä oleva merkkikooditus (esimerkiksi UTF-8, UTF-16 tai ISO 8859-1), mikä ei ole pakollista, mutta on kuitenkin suotavaa ja hyvän koodaustavan mukaista. Jos käytössä rakennekuvaus eli DTD tai XML Schema dokumentti, tulee se määrittää heti esittely osan jälkeen, ja tämän jälkeen merkitään mahdolliset käytössä olevat tyylimäärittelyt eli CSS- tai XSL-tyylitiedostot. Mikäli dokumenttiin halutaan lisätä tekstiä, jota parseri ei huomioi, voidaan käyttää kommenttia muodossa `<!--tämä on kommentti -->`. Alla esitetään XML dokumentin rakennetta ja käytettäviä elementtejä attribuutteineen. [Antoniou & Harmelen 2004 s. 23-26; Bray et al. 2004; Tuikka & Kanala 2001 s.3-17]

```
<!--tämä on esittelyosa-->
<?xml version="1.0" encoding="UTF-16"?>
<!DOCTYPE rakenne SYSTEM "rakenne.dtd">
<?xml:stylesheet type="text/css" href="tyylitiedosto.css">

<!--tämä on XML-sisältöä-->
<juurielementti>
  <potilas sotu="120345-678A">
    <nimi>
      <etunimi> Marja </etunimi>
      <sukunimi> Piiras </sukunimi>
    </nimi>
    <puhelinnumero> (123) 1234567</puhelinnumero>
  </potilas>
</juurielementti>
```

#### 4.2.1 DTD - määrittely

Document Type Definition määrittelyssä todetaan mitä tageja ja attribuutteja käytetään XML dokumentin sisällössä, missä niitä voidaan käyttää ja miten ne toimivat yhdessä. DTD voi olla joko ulkoinen (external DTD) tai sisäinen (internal DTD). Sisäinen DTD sijoitetaan XML dokumenttiin rakennemäärittelyn yhteyteen hakasulkujen sisään (`<!DOCTYPE rakenne [...]>`) ja ulkoinen DTD on oma erillinen tiedosto. Ulkoisten DTD:ien käyttöä suositellaan, sillä silloin yhtä DTD:tä voidaan käyttää useampaan XML dokumenttiin ja näin vältetään asioiden toisto ja saadaan dokumenteista johdonmukaisia. [Antoniou & Harmelen 2004 s. 32; Tuikka & Kanala 2001 s.23-29]

```
<!ELEMENT henkilo (nimi,puhelinumero)>
<!ELEMENT nimi (#PCDATA)>
<!ELEMENT puhelinnumero (#PCDATA)>
```

Yllä olevassa esimerkissä on tyypillinen esimerkki DTD:stä. Ensimmäisenä kerrotaan, että XML dokumentissa voidaan käyttää elementtejä `henkilo`, `nimi` ja `puhelinumero`. `Henkilo` – elementti sisältää `nimi`- ja `puhelinumero`-elementit esitetyssä järjestyksessä ja näillä elementeillä voi olla mitä tahansa sisältöä. DTD:llä on vain yksi atominen tyyppi elementeille ja se on `#PCDATA` (Parserable Character DATA). Jos elementti ei sisällä mitään tietoa, voidaan se määritellä DTD:ssä tyypillä `EMPTY`. Attribuuttien kanssa DTD:hen täytyy lisätä attribuuttityyppi ja –arvo sekä mahdollinen kardinaliteettioperaattori (cardinality operator), joita on kolme; `?` tarkoittaa, että elementti esiintyy nolla tai yksi kertaa, `*` tarkoittaa, että elementti esiintyy nolla tai useamman kerran ja `+` tarkoittaa, että elementti esiintyy yhden tai useamman kerran. Jos operaattori jätetään kokonaan pois, tarkoittaa se, että elementti esiintyy täsmälleen yhden kerran. [Antoniou & Harmelen 2004 s. 32-33; Tuikka & Kanala 2001 s.23-29]

Seuraavassa esimerkki DTD:stä, jossa esitetään sähköisen reseptin kirjoitusta. Esimerkki on kuvitteellinen, sillä käytännössä yhdellä reseptillä voidaan määrätä vain yhtä lääkettä, mutta esimerkissä näkyvät hyvin DTD:n eri toiminnot. Tunniste `<!ATTLIST` kertoo edelliseen

elementtiin liittyvät attribuutit ja <!ENTITY> määrittää entiteetit. [Antoniou & Harmelen 2004 s. 33-36; Tuikka & Kanala 2001 s.23-29]

```
<!DOCTYPE RESEPTINKIRJOITUS [  
<!ELEMENT RESEPTINKIRJOITUS (RESEPTI+)>  
<!ELEMENT RESEPTI (POTILAS , SOTU , LAAKE , VAHVUUS , KESTO)>  
<!ELEMENT POTILAS (#PCDATA)>  
<!ELEMENT SOTU (#PCDATA)>  
<!ELEMENT LAAKE (#PCDATA)>  
<!ELEMENT VAHVUUS (#PCDATA)>  
<!ELEMENT KESTO (#PCDATA)>  
<!ATTLIST RESEPTI LAAKARI CDATA #REQUIRED>  
<!ATTLIST RESEPTI OHJE CDATA #REQUIRED>  
<!ATTLIST RESEPTI KAYTTOTARKOITUS CDATA #IMPLIED>  
<!ATTLIST RESEPTI SIC CDATA #IMPLIED>  
<!ENTITY RESEPTINKIRJOITUS "Effica 1.0">  
<!ENTITY LAAKARI "Pekka Pilleri">  
<!ENTITY LAITOS "Saimaan Sairaala">  
>
```

#### 4.2.2 XML Schema

Toinen XML dokumentin rakenteen kuvaamiseen käytettävä tapa on XML Schema. Sen avulla rakennekuvauksista saadaan tarkempia ja määrittelyyn voidaan sisällyttää muutakin tietoa kuin rakennekuvauksia, kuten tietotyyppejä (int, string, date) ja nimiavaruuksia. Erittäin hyödyllinen se on verkossa siirrettävän tiedon kuvaamisessa ja tarkistuksessa ja näin ollen sitä käytetään paljon muun muassa sähköisessä kaupankäynnissä ja tietokantatietojen siirrossa. XML Schemaan viitataan myös nimellä XSD (XML Schema Definition) eli XML skeemamäärittely. Tästä tuleekin jokaisen rivin alkuun tunniste `xsd:`, sillä XML Schema dokumentti on myös XML dokumentti ja ilman tunnistetta ne voidaan sekoittaa toisiinsa. Juurielementtinä toimii `<schema>`, ja sen sisällä viitataan käytössä olevaan nimiavaruuteen `xmlns` -attribuutilla. Seuraavassa on esimerkki XML Schema dokumentista. [Antoniou & Harmelen 2004 s. 37-43; Tuikka & Kanala 2001 s.23, 29-33]

```

<xsd:schema xmlns:xsd=http://www.w3c.org/2000/10/XMLSchema version="1.0">
<xsd:element name="elementtiResepti" type="attribuuttiLaake">

<xsd:complexType name="reseptiNimi">
  <xsd:element name="laakeNimi" type="laakeTyyppi">
</xsd:complexType>

<xsd:complexType name="reseptiNimi">
  <xsd:element name="laakeElementti1" type="xsd:string"/>
  <xsd:element name="laakeElementti2" type="xsd:string"/>
  <xsd:element name="kuva1" type="xsd:empty"/>
</xsd:complexType>

</xsd:schema>

```

Elementti reseptiNimi on tyypiltään complexType, mikä tarkoittaa, että sen sisällä on muita elementtejä ja attribuutteja. Lisäksi voidaan määrittellä simpleType- elementtejä, jotka voivat sisältää ainoastaan numero ja merkkietoa. Osa simple-tyypeistä on määritelty valmiiksi, jotta niitä ei tarvitse enää erikseen määrittellä. Näitä ovat esimerkiksi string, boolean, time ja date. Alaelementeillä ja pääelementeillä voi olla sama nimi kuten esimerkissä, mutta tällöin on muistettava, että ne kuitenkin ovat eri asioiden nimiä. [Antoniou & Harmelen 2004 s. 37-43; Tuikka & Kanala 2001 s.23, 29-33]

Monipuolisemman XML Scheman käyttö on yleistynyt DTD:n tilalla pikkuhiljaa ja sen odotetaan vielä syrjäyttävän DTD:t kokonaan. Tämä johtuu siitä, että XML Schema on avoin ja laajennettavissa myös tuleville lisäyksille, se on monipuolisempi ja toimivampi kuin DTD, tukee tietotyyppisiä ja nimiavaruuksia ja se kirjoitetaan XML:llä. XML Scheman käyttö vähentää myös virheitä lähetettävissä tiedoissa ja näin parantaa turvallisuutta esimerkiksi sähköisessä kaupankäynnissä ja sähköisessä reseptissä. Mahdollisia virheitä voivat olla esimerkiksi päivämäärät. Päiväys "02-09-2006" voi joissain maissa olla 2.syyskuuta ja toisissa 9.helmikuuta. XML Schemassa päiväys määritellään aina muodossa vuosi-kuukausi-päivä (yyyy-mm-dd; <date type="date">2006-02-

09</date>), jolloin sekä lähettäjä että vastaanottaja saavat selville saman päivämäärän. [Antoniou & Harmelen 2004 s. 37-43; Tuikka & Kanala 2001 s.23, 29-33]

### 4.3 RDF - kuvauskieli

XML on käytännöllinen ja hyvin yleistynyt tapa kuvata ja välittää tietoa sovellusten välillä verkossa. Se ei kuitenkaan tarjoa mitään tapaa kuvata tiedon semantiikkaa eli sen tarkoitusta. XML ei esimerkiksi kerro mitään tarkkaa tapaa sisäistää dokumenttien tietoja, vaan tämä on aina käyttäjän mielen mukainen. Otetaan esimerkiksi lause ”Antti Angiina käyttää aspiriinia.”. Alla näkyvät eri vaihtoehdot, miten lause voidaan esittää XML:llä:

```
<laake name="Aspiriini">
  <potilas>Antti Angiina</potilas>
</laake>
```

```
<potilas name="Antti Angiina">
  <käyttää>Aspiriini</käyttää>
</potilas>
```

```
<Resepti>
  <potilas>Antti Angiina</potilas>
  <laake>Aspiriini</laake>
</Resepti>
```

Tämän ongelman avuksi tulee Resource Description Framework (RDF), tietomalli, jolla voidaan kuvata myös tiedon merkitystä. RDF rakentuu lauseista (statement), joissa on objekti, attribuutti ja arvo (object-attribute-value triple). RDF toimii samalla syntaksilla kuin XML:kin, mutta sitä voidaan käyttää myös muiden syntaxien kanssa. RDF:ssä käyttäjät voivat määrittää oman terminologiansa RDFS:n eli RDF Scheman avulla. Vaikka nimi on hyvin samantyylinen kuin XML:n Schema, ei näillä kuitenkaan ole vastaavanlaista yhteyttä; XML Schema määrittää XML:n rakennetta kun taas RDF Schema määrittää sanaston, jota käytetään RDF:n tietomalleissa. [Antoniou & Harmelen 2004 s. 61-64]

RDF koostuu kolmesta pääkonseptista; resurssista (resource), ominaisuudesta (property) ja lauseesta (statement). Resurssit ovat eräänlaisia objekteja eli tapahtuman kohteita. Ne voivat olla mitä tahansa kirjailijasta kyselyyn, joilla jokaisella on tunniste, URI (Universal Resource Identifier). Ominaisuudet ovat erityislaatuisia resursseja, sillä ne kuvaavat resurssien välisiä yhteyksiä (ikä, otsikko, säveltänyt). Kuten resursseilla, myös ominaisuuksilla on URI, joka käytännössä on usein URL (Universal Resource Locator). Lauseet puolestaan ovat objekti-attribuutti-arvo –kolmikkoja, jotka kertovat resurssien ominaisuuksia. Tällainen lause voi olla esimerkiksi ”Anni Apteekki omistaa verkkosivun [www.apteekki.net](http://www.apteekki.net).” ja sen objekti-attribuutti-arvo-kolmikko on (”Anni Apteekki”, <http://www.mydomain.org/site-owner>, <http://www.apteekki.net>). Alla on edellä mainitulle lauseelle kirjoitettu RDF. Ylimmällä rivillä määritellään, että käytössä on XML. Tämä on mainittava aina kun halutaan kirjoittaa RDF dokumentti, joka käyttää XML syntaksia. [Antoniou & Harmelen 2004 s. 61-66]

```
<?xml version="1.0" encoding="UTF-16">
<rdf:RDF xmlns:rdf="http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
  xmlns:mydomain="http://www.mydomain.org/my-rdf-ns">

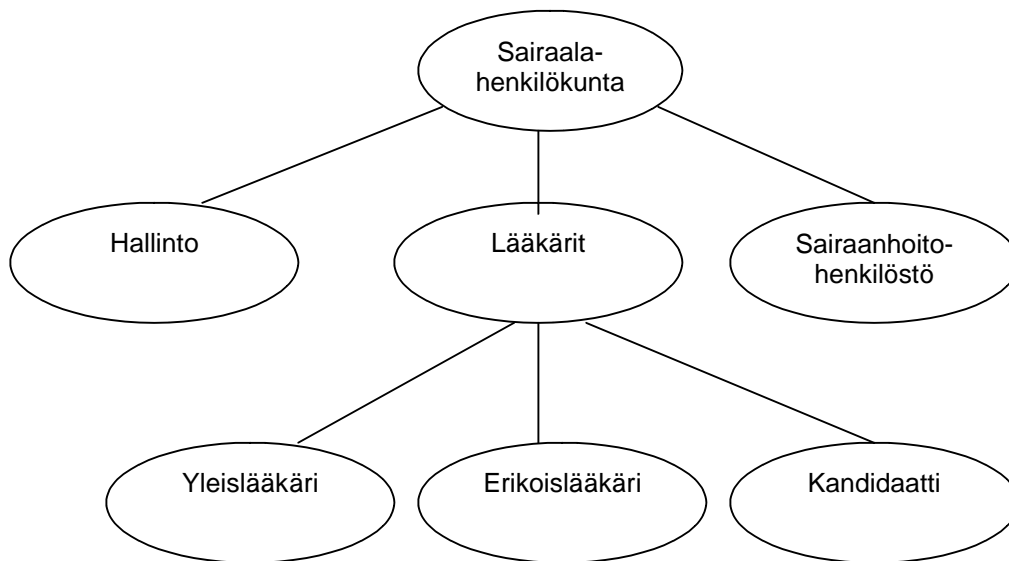
  <rdf:Description rdf:about=http://www.apteekki.net>
    <mydomain:site-owner>Anni Apteekki</mydomain:site-owner>
  </rdf:Description>
</rdf:RDF>
```

### 4.3.1 RDF Schema

RDF:n etuna on, että se antaa käyttäjien kuvata tietoja käyttämällä omia sanastojaan. Esimerkiksi XML dokumentin tietoja voidaan kuvata käyttämällä samaa syntaksia, jolloin käyttäjän ei tarvitse opetella uutta kieltä tätä varten. Koska RDF on yleismaailmallinen kieli, se ei tee minkäänlaisia oletuksia sovelluksista tai ympäristöistä, joissa sitä käytetään. Tämän takia tarvitaan RDF Schema, jonka avulla jokainen käyttäjä voi määrittää itse oman terminologiansa ja käyttöympäristön semantiikan. RDF:ssä määritetään resursseja, eli tiettyjä yksittäisiä objekteja. RDF Schemassa pyritään määrittämään luokkia (class), jotka

kuvaavat näitä objektityyppejä. Luokkia voidaan ajatella eräänlaisena elementtijoukkona ja jokainen objekti, joka kuuluu luokkaan, on sen luokan instanssi (instance). Luokkien avulla määrätään rajoituksia sille, mitä skeemaa käyttävässä RDF dokumentissa voidaan esittää. Samoin luokkien ominaisuuksien halutaan periytyvän alempiin luokkiin, mikä onkin yksi RDF:n perusominaisuus. [Antoniou & Harmelen 2004 s. 80-81]

Otetaan esimerkiksi sairaala, jossa työskentelee henkilökunnan jäseniä. Nämä jäsenet voidaan edelleen luokitella hallintohenkilöstöksi, lääkäreiksi ja sairaanhoitohenkilöstöksi, joista lääkärit voidaan edelleen jakaa yleislääkäreiksi, erikoislääkäreiksi ja lääketieteen kandidaateiksi (kuva 5). Jos määritetään, että potilaita voivat hoitaa ainoastaan lääkärit ja ennalta on määritetty Timo Tohtori yleislääkäreiksi, niin edeltävän rajoituksen mukaan, hän ei voi hoitaa potilaita. Tämän takia halutaan tietyt ominaisuudet periyttää luokasta toiseen. Tässä tapauksessa halutaan, että yleislääkärit, erikoislääkärit ja kandidaatit perivät pätevyyden hoitaa potilaita lääkärit-luokalta, sillä he ovat osa tätä luokkaa. Näin ollen RDF Schema määrittää uudelleen lauseen ”on aliluokka” semantiikkaa, niin että sen ymmärtäminen ei ole kiinni sovelluksesta vaan se on kaikkien RDF:ää käyttävien ohjelmistojen käytössä. RDF schema onkin eräänlainen alkeellinen ontologiakieli. [Antoniou & Harmelen 2004 s. 81-82]



Kuva 5 Luokkien hierarkia [Antoniou & Harmelen 2004 s. 82]

### 4.3.2 OWL - ontologiakieli

Semanttisen verkon toimintaa varten sekä XML, RDF että RDF Schema ovat vielä rajoittuneita semantiikan kuvaajia. Jotta tieto saadaan koneiden ymmärtämään muotoon, otetaan käyttöön Web Ontology Language (OWL), joka kuvaa hierarkkista järjestelmää. OWL on siis ontologiakieli, joka sallii käyttäjien kirjoittaa tarkkoja ja muodollisia käsitelmalleja, joilla on vähintäänkin hyvin määritelty syntaksi, muodollinen semantiikka, mukautuvat lausekkeet, tehokas päättelytuki ja tarvittava ilmaisuvoima. Hyvin määritelty syntaksi on oleellista tiedolle, jota koneet käsittelevät. Muodollinen semantiikka kuvaa tarkasti tiedon tarkoitusta, millä pyritään siihen, ettei siinä jää tilaa ihmisten eikä koneiden tulkinnalle tietoja käsitellessä. [Antoniou & Harmelen 2004 s. 109-112]

OWL perustuu RDF:ään ja RDF Schemaan, joten se käyttää samaa XML-pohjaista syntaksia. Muitakin syntaktisia versioita on kehitetty, sillä RDF:n syntaksi on suhteellisen hankalaa lukea. OWL dokumentteja kutsutaan myös OWL ontologioiksi, ja ne ovat kaikki samalla RDF dokumentteja. Täten molempien juurielementti on sama `rdf:RDF` ja ontologian alussa ilmenevät väitteet kootaan yhteen `owl:Ontology` elementin alle. Mikäli ontologiaan kuuluu muitakin ontologioita, lisätään `owl:Ontology` elementin alle `owl:imports`, jossa mainitaan kaikki muut ontologiat. Käytännöllisyyden vuoksi OWL jaetaan kolmeen alikieleen; OWL Full, OWL DL (Description Logic) ja OWL Lite. OWL Full käsittää koko kielen ja sallii yhdistelmiä RDF:n ja RDF Scheman kanssa. OWL Full on siis yhteensopiva RDF:n kanssa semantiikaltaan ja syntaksiltaan niin, että jokainen laillinen RDF dokumentti on myös laillinen OWL dokumentti. OWL DL rajoittaa OWL:n ja RDF:n muodostimien käyttöä. OWL Lite on tarkoitettu erittäin kevyeen käyttöön ja siitä on riisuttu muun muassa hajanaisuus lauseet ja mielivaltainen kardinaliteetti. Rajoituksista huolimatta OWL Liten etuna on, että se on helpompi ymmärtää ja helpompi toteuttaa. [Antoniou & Harmelen 2004 s. 109-114]



#### 4.4 Älykortti ja sähköinen resepti

Saksassa sähköinen resepti toimii elektronisen terveystietokortin avulla, ja myös Suomessa älykorttia käytetään lääkäreiden tunnistamiseen sähköisen reseptin järjestelmissä. Sähköistä korttia voidaan helposti käyttää tunnistamiseen, etenkin PKI-järjestelmissä (Public Key Infrastructure), sillä sitä voidaan käyttää avainlaitteena. Käytännössä kortille on upotettu pieni tietokone ja sen käyttöjärjestelmä sekä pieni siru, jonka avulla korttia voidaan lukea. Tämän takia korttia kutsutaankin älykortiksi. Sirulle voidaan tallentaa tietoja ja niitä voidaan lukea siitä, aivan kuten tietokoneen kovalevyiltä. Sirukortteja on kuitenkin olemassa myös ainoastaan muistikäyttöön, jolloin ne eivät voi käsitellä niille talletettuja tietoja, vaan ainoastaan suorittaa ennalta määritettyjä toimintoja. Tällaisia muistikortteja ovat esimerkiksi matkapuhelimissa käytettävät pre-paid kortit. Sähköisen reseptin järjestelmissä käytettävillä korteilla on oltava mahdollisuus reseptitietojen lisäämiseen ja käsittelyyn, joten tällaisten korttien tulisi sisältää käyttöjärjestelmä ja mikroprosessori. [Setec 2006]

Sirukorttien teknologia pohjautuu ISO:n (International Organization for Standardization) standardiin ISO/IEC 7816. Vaikka sirukortin fyysiset ominaisuudet ja perustoiminnot on standardoitu, kortilla olevat turvajärjestelmät eivät vielä ole, vaan jokainen valmistaja voi käyttää valitsemaansa turvamenetelmää. Turvajärjestelmissä voidaan kuitenkin käyttää muun muassa Euroopan Yhteisön luomaan ITSEC- (Information Technology Security Evaluation Criteria) kriteeristöä tai kansainvälistä Common Criteria - kriteeristöä. Tunnetuimpia sirukortteja ovat matkapuhelimissa käytetyt SIM-kortit (Subscriber Identity Module) sekä Suomessakin nykyisin pakolliset pankkien sirulliset EMV-kortit (Europay, Master Card, Visa), kuten Visa Electron, joita havainnollistaa kuva 6. Suomessa muun muassa pankeille kortteja toimittaa Setec, joka käyttää korttien käyttöjärjestelmänä itse kehittämänsä SetCOS™ -käyttöjärjestelmää. Käyttöjärjestelmä hallitsee kaikkia kortilla tapahtuvia toimintoja. Se pitää huolta siitä, että kortilla olevat tiedot ovat suojattuina ja että kortille tulevat laitteiden antamat komennot suoritetaan. Sirulle voidaan lisäksi liittää PKI-teknologiaa, jonka avulla kortin haltija voidaan tunnistaa ja tietoja voidaan salata sekä sähköisesti allekirjoittaa. Kaikki salaisten tietojen käsittely tapahtuu kortin sisällä, joten

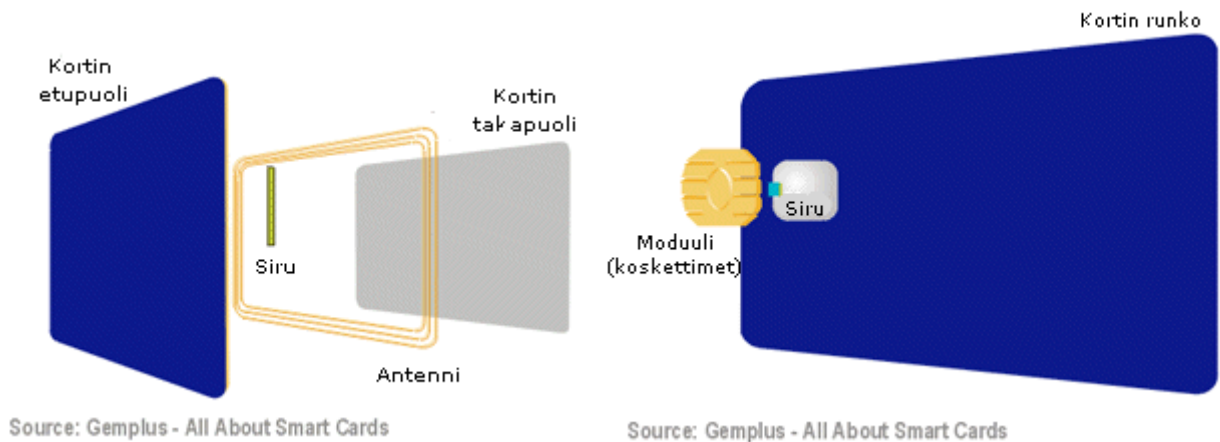
tietoja ei voida kopioida, mikä puolestaan lisää kortin turvallisuutta. Kortilla voi olla käytössä myös henkilökohtainen PIN-koodi (Personal Identification Number), jolla päästään käyttämään kortille tallennettuja sovelluksia. [ISO 2006; ITSEC 1991; Luottokunta 2006; Setec 2006]



© 2005 Setec

*Kuva 6 Sirukortti ja SIM-kortti [Setec 2006; Wikipedia 2006]*

Sirukorttia käytettäessä tarvitaan päätelaite, jotta halutut komennot voidaan siirtää kortille. Kun sirukortti laitetaan päätelaitteeseen, lähettää laite kortille ensin RST (reset) –signaalin. Kortti vastaa RST –signaaliin lähettämällä laitteelle ATR (answer-to-reset) –signaalin. ATR-signaali kertoo laitteelle mitä protokollaa kortti käyttää ja minkä tyyppinen kortti se on. Kun ATR-signaali on lähetetty, voivat siru ja päätelaite kommunikoida keskenään. Niiden välillä kulkevat käskyt ovat paketeissa, joita kutsutaan lyhenteellä TPDU (Transport Protocol Data Unit) ja sirukortissa on tiedostojärjestelmä, joka vastaa toiminnaltaan MS-DOSia. Sirukortin turvallisuutta voidaan ylläpitää lisäämällä kortille PIN-koodi, joka on syötettävä ennen mitään muita komentoja. Kortin ja päätelaitteen välillä kulkevaa tietoa voidaan suojata salauksella, esimerkiksi DES- (Data Encryption Standard) tai RSA- (Rivest, Shamir, Adleman) salauksella tai MAC (Message Authentication Code) tarkistesummalla. [Cyberspace Center 1999]



Kuva 7 Radioaalloilla toimiva ja lukijan vaatima sirukortti [Gemplus]

Vaikka termejä älykortti ja sirukortti käytetään usein tarkoittamaan samaa asiaa, ne voivat kuitenkin olla aivan erilaisia kortteja. Tyypillisimpiä sirukortteja ovat kosketuskortit, mikä tarkoittaa, että kortissa olevaa sirua luetaan koskettamalla sitä lukijassa olevilla liittimillä. Älykorteissa voidaan käyttää myös radioaaltoja, jolloin korttia luettaessa sen ei tarvitse koskettaa lukijaa. Erilaisia kortteja ja niiden rakennetta havainnollistetaan kuvassa 7. Radioaalloilla toimivissa korteissa on sisällä pieni antenni ja kortti voidaan lukea, kun se tuodaan päätelaitteen lähelle. Nämä kaksi ominaisuutta voidaan haluttaessa yhdistää ja yhdistelmäkorttia (Hybrid Smart Card, Dual-Interface Chip Smart Card) voidaan käyttää päätelaitteesta riippuen joko radioaaltojen avulla tai sijoittamalla se lukijaan. Kortilla oleva siru on pieni yleensä kullattu ja piistä valmistettu moduuli, jonka avulla kortti voi kommunikoida päätelaitteen kanssa. Siru istutetaan muoviseen korttiin tehtyyn koloon ja kiinnitetään joko kullalla tai kuumaliimalla. Korteilla itsellään ei ole virtalähdettä, joten kontaktipinnan kautta päätelaitteesta kulkee myös virtaa kortin käyttämiä sovelluksia varten. Radioaaltoja käyttävissä älykorteissa virta saadaan samalla tavalla päätelaitteesta, mutta tällöin kortin tarvitsema energia saadaan nostamalla päätelaitteesta kulkevaa sähkövirtaa. [GSA 2004 s. 16-34]

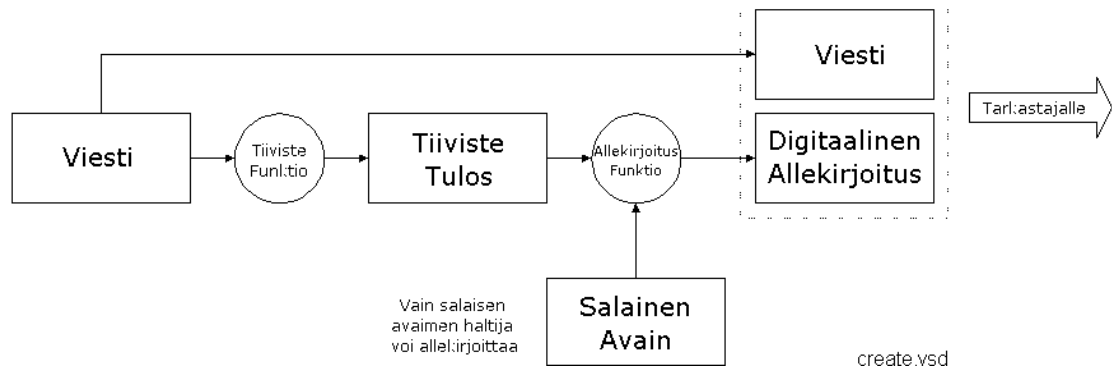
## 4.5 Digitaalinen allekirjoitus

Suomessa ja useissa muissa maissa lääkäri todistaa kirjoittaneensa paperisen lääkemääräyksen lisäämällä siihen oman allekirjoituksensa. Allekirjoitus samalla vahvistaa reseptin aidoksi ja vain allekirjoitetut reseptit voidaan toimittaa apteekeista. Sähköisen reseptin tapauksessa autenttisuus todistetaan lääkärin sähköisellä allekirjoituksella. Lääkäri kirjoittaa sähköisen lääkemääräyksen reseptinkirjoitusohjelmalla, johon on ensin tunnustautunut sähköisen henkilökortin (sirukortin) avulla. Kun resepti on valmis, lääkäri allekirjoittaa reseptin sähköisesti, taas käyttämällä sähköistä henkilökorttiaan, ja lähettää reseptin joko reseptitietokantaan tai apteekkiin. Apteekissa allekirjoitus tarkastetaan ja aidon reseptin lääkkeet toimitetaan potilaalle. Viestintävirasto määrittelee sähköisen allekirjoituksen miksi tahansa sähköisessä muodossa olevaksi allekirjoitukseksi. Näin ollen sähköinen allekirjoitus voi olla myös sähköpostin perään lisätty oma nimi. Sähköisissä resepteissä käytetään kuitenkin digitaalista allekirjoitusta, joka on salausmenetelmän avulla tuotettu sähköinen allekirjoitus. [Viestintävirasto 2005]

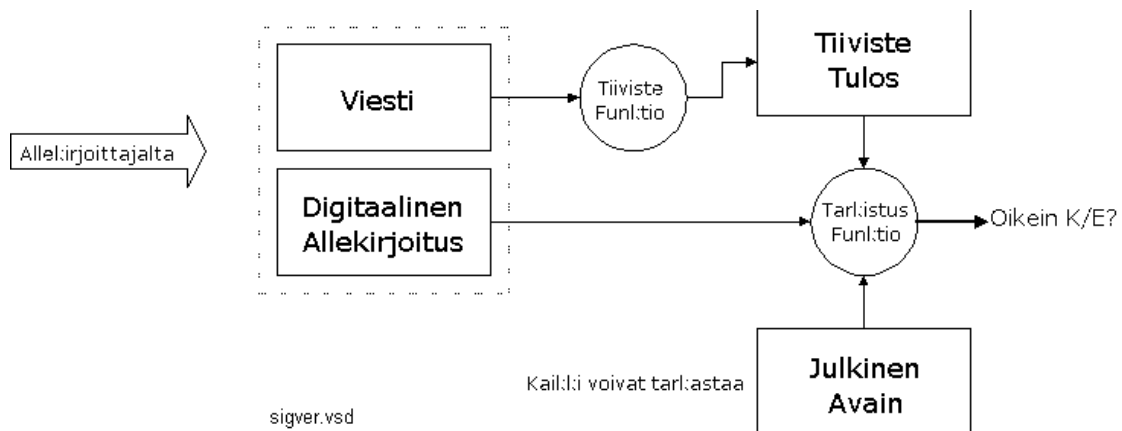
Digitaalinen allekirjoitus ei käytännössä ole allekirjoitus, sillä siinä ei käytetä lähettäjän nimeä millään tavalla, mutta sillä varmistetaan viestin lähettäjän henkilöllisyys ja viestin aitous sekä muuttumattomuus, joten sana allekirjoitus kuvaa sitä hyvin. Digitaalinen allekirjoitus toteutetaan yleensä julkisen avaimen menetelmällä (PKI – Public Key Infrastructure), jossa tietyn algoritmin avulla luodaan kaksi täysin erillistä, mutta matemaattisesti yhteen liittyvää avainta. Toinen avaimista on salainen ja sitä käytetään allekirjoituksen laatimiseen ja/tai lähetettävän tiedon salaamiseen. Julkinen avain annetaan kaikille niille, joille halutaan allekirjoituksen avulla todistaa lähettäjän henkilöllisyys ja viestin muuttumattomuus. Vaikka avaimet ovat ”sukua” toisilleen, julkisen avaimen avulla on laskennallisesti mahdollista selvittää salainen avain. Näin ollen, jos salaisen avaimen haltija pitää avaimen vain omassa tiedossaan, ei hänen allekirjoitustaan voida väärentää. [ABA 2006; Viestintävirasto 2005]

Pilotissa lääkärin sähköisellä henkilökortilla on tämän yksityinen salainen avain. Salainen avain voi olla myös PIN - koodina, USB – avaimessa (Universal Serial Bus) tai SWIM-

sirulla (Subscriber/WAP Identification Module, SWIM = SIM + WIM). On tärkeää, että kukaan ei saa tietää lääkäreiden salaisia avaimia, joten niiden tulisi olla talletettuna laitteelle, josta niitä ei voida lukea, mutta jonka avulla digitaalinen allekirjoitus on helppo suorittaa. Kun resepti on kirjoitettu, ohjelma laskee henkilökortilla olevista tiedoista matemaattisesti tiivisteen (hash), joka tämän jälkeen salataan lääkärin salaisella avaimella (kuva 8). Tiiviste on eräänlainen lyhennetty versio koko lähetettävästä tiedosta ja, jos tietoja ei muuteta, se pysyy koko ajan samana. Salattu tiiviste on puolestaan digitaalinen allekirjoitus, joka salaisen avaimen ansiosta on väärentämätön. Kun digitaalisella allekirjoituksella varustettu sähköinen resepti saapuu apteekkiin, farmaseutti tarkastaa allekirjoituksen purkamalla salauksen julkisella avaimella ja vertaamalla tiivistettä itse laskemaansa tiivisteeseen samasta reseptistä. Jos tiivisteet ovat samoja, eivät tiedot ole muuttuneet matkalla ja niiden lähettäjä on kyseisen salaisen avaimen haltija (kuva 9). [ABA 2006; Viestintävirasto 2005]



Kuva 8 Digitaalisen allekirjoituksen muodostaminen [ABA 2006]



Kuva 9 Digitaalisen allekirjoituksen tarkastaminen [ABA 2006]

#### 4.5.1 Tiiviste

Tiivistefunktio (hash function) on algoritmi, jossa summittaisen mittaisesta viestistä lasketaan matemaattisen kaavan avulla kiinteän mittainen tiiviste (hash, hash value). Viesti voi siis olla minkä mittainen tahansa, mutta käytetystä algoritmista riippuen tiiviste on aina samanmittainen, esimerkiksi 128 tai 256 bittiä. Hyvä tiivistefunktio on yksisuuntainen ja laskee jokaisesta erilaisesta viestistä erilaisen tiivistearvon. Yksisuuntaisuudella tarkoitetaan, että viestistä on helppoa laskea tiiviste, mutta tiivistearvosta on laskennallisesti mahdotonta selvittää viestin sisältö. On tietenkin myös oleellista, että kahdesta viestistä ei pidä pystyä tuottamaan samaa tiivistearvoa. Eräitä tunnettuja tiivistefunktioita ovat SHA-1 (Secure Hash Algorithm) ja MD5 (Message Digest). MD5 kuuluu Ron Rivestin suunnittelemaan tiivistealgoritmien perheeseen. Se kehitettiin vuonna 1991 MD4-algoritmin pohjalta ja se tuottaa annetusta viestistä 128-bittisen tiivisteen. SHA-1 on standardoitu (SHS, Secure Hash Standard FIPS 180; ANSI X9.30) NISTin (National Institute of Standards and Technology) kehittämä tiivistealgoritmi, joka laskee alle  $2^{64}$  bittiä pitkistä viestistä 160-bittisen tiivisteen. SHA-1 on hieman hitaampi kuin MD5, mutta pidempi tiiviste tekee siitä samalla turvallisemman esimerkiksi raan voiman hyökkäyksiä vastaan. [RSA 2000]

#### 4.5.2 XML Allekirjoitus

Sähköisen reseptin pilotissa sosiaali- ja terveysministeriö määräsi, että reseptit on pystyttävä allekirjoittamaan sähköisesti tiettyjen erikseen määritettyjen vaatimusten mukaan. MediWeb Oy määritteli näiden vaatimusten pohjalta kaikkien järjestelmävalmistajien saataville sanomakuvauksen, jossa esitetään esimerkiksi arkkitehtuurivaatimukset ja sanoman rakenne. Sanomakuvauksessa todettiin, että koska reseptikeskus pohjautuu Web palvelu – arkkitehtuuriin, voidaan siinä käyttää kahta digitaalisen allekirjoituksen standardia – XML Signaturea ja XAdESia (XML Advanced Electronic Signatures). MediWebin mukaan World Wide Web Consortiumin XML Signature sopi pilotin käyttöön paremmin, sillä se oli paremmin edustettuna markkinoilla ja täytti hyvin sosiaali- ja terveysministeriön vaatimukset. XML allekirjoitusta voidaan käyttää minkä tahansa digitaalisen sisällön allekirjoittamiseen, jopa itse XML-koodin allekirjoittamiseen, mitä varten ne onkin suunniteltu. Se on menetelmä, jossa voidaan liittää tietty avain tiettyyn tietoon viittaamatta henkilöihin, organisaatioihin tai tietoon, jota allekirjoitetaan. XML – allekirjoitus on oleellinen osa turvallisia XML – sovelluksia, mutta se ei yksinään vielä takaa turvallisuutta tai luottamusta, vaan sitä on syytä käyttää yhdessä muiden turvamenetelmien kanssa. [Bartel et al. 2002; Lehto & Hohteri 2003]

Jotta XML allekirjoitus saadaan tehtyä, on ensimmäisenä tiivistettävä allekirjoitettava tieto. Saatu tiiviste laitetaan tämän jälkeen alkioon, jossa on muutakin tietoa ja koko alkio tiivistetään uudelleen. Tästä saatu tiivistearvo sitten allekirjoitetaan salauksen avulla. XML allekirjoitusta merkitään *Signature*-elementillä, missä ”?” tarkoittaa nollaa tai yhtä tapahtumaa, ”+” tarkoittaa yhtä tai useampaa tapahtumaa ja ”\*” tarkoittaa nollaa tai useampaa tapahtumaa. Alla on esimerkki *Signature*-elementin rakenteesta:

```
<Signature ID?>
  <SignedInfo>
    <CanonicalizationMethod/>
    <SignatureMethod/>
    (<Reference URI? >
      (<Transforms>)?
```

```

        <DigestMethod>
        <DigestValue>
    </Reference>)+
</SignedInfo>
<SignatureValue>
(<KeyInfo>)?
(<Object ID?>)*
</Signature>

```

SignedInfo- elementti määrittää mitä allekirjoitetaan ja mitä algoritmia käytetään. CanonicalizationMethod- ja SignatureMethod- elementit SignatureValue- elementin käytössä ja niitä käytetään estämään tietojen peukaloiminen ja muuttuminen. SignatureMethod on siis algoritmi, jolla allekirjoitettava tieto muutetaan SignatureValue -arvoksi. Reference- elementin URIt (Uniform Resource Identifier) määrittelevät mitä lähteitä on allekirjoitettu ja tarvitseeko niille tehdä jotakin muutoksia ennen tiivisteen laskemista. Itse Reference- elementti sisältää tiivistemenetelmän ja tiiviste- arvon, joka on laskettu allekirjoitettavasti tiedosta. SignatureValue- elementti on allekirjoituksen koodattu arvo eli itse allekirjoitus, joka on laskettu SignedInfo- elementistä CanonicalizationMethod- elementin avulla. KeyInfo- elementti on vapaaehtoinen elementti, jota käytetään ilmoittamaan vastaanottajalle avain, jolla allekirjoitus voidaan todentaa. Jos KeyInfo- elementtiä ei käytetä, täytyy vastaanottajan tietää avain etukäteen tai tietää mistä sen löytää. Myös Object- elementti on vapaaehtoinen ja sitä käytetään kotoituvien allekirjoitusten (enveloping signature) tietojen säilytykseen. [Bartel et al. 2002]

Valmistettu allekirjoitus voidaan validoida käyttämällä menetelmää, jossa ensin kelpoistetaan viitteet (Reference Validation) ja sitten allekirjoitus. Tällä on tarkoitus varmistaa, että lähteen on todellakin allekirjoittanut sen lähettäjä. Ensimmäisenä jokainen Reference- elementin tiiviste tarkastetaan. Jokainen tiivistettä vastaava resurssi etsitään ja siihen käytetään allekirjoituksen luomista varten olevia sovelluksen tuomia muunnoksia sekä tiivistemenetelmää. Saatua uutta arvoa verrataan jo valmiiseen DigestValue-



elementin arvoon ja jos ne ovat samat, on tämä validi. Allekirjoituksen tarkastaminen aloitetaan käyttämällä kanonisointimenetelmää, joka kuvataan CanonicalizationMethod- elementissä. Avain haetaan joko käyttämällä KeyInfo- elementtiä tai muuta vastaavaa keinoa ja allekirjoitus vahvistetaan käyttämällä SignatureMethod- elementissä olevaa menetelmää. [Bartel et al. 2002]

XML allekirjoituksen luomisella on myös ongelmansa, sillä XML-dokumentilla voi olla useampi kuin yksi sallittu muoto. Esimerkiksi tyhjä tila XML-elementtien sisällä on syntaksin kannalta merkityksetöntä, mutta digitaalisen allekirjoituksen tiivistearvon kannalta hyvin merkittävää. Näin ollen XML-dokumentissa oleva tagi <Signature > on sama kuin <Signature>. Tiivistettä ja digitaalista allekirjoitusta laskettaessa nämä kuitenkin tuottavat kaksi erilaista tiivistearvoa ja täten kaksi erilaista allekirjoitusta, sillä yhdenkin bitin muutos lähdetiedoissa aiheuttaa digitaaliseen allekirjoitukseenkin muutoksen. Tämän ongelman ratkaisuna käytetään kanonisointia, mikä onkin pakollinen aina SignedInfo- elementin allekirjoituksissa ja hyvin yleinen myös muiden elementtien käytössä. Kanonisoinnilla taataan, että kaksi loogisesti identtistä dokumenttia tuottaa kaksi identtistä digitaalista allekirjoitusta. XML Allekirjoituksissa käytettäville algoritmeille on annettu suositukset World Wide Web Consortiumin (W3C) toimesta, jotta se toimisi turvallisesti ja samalla tavalla kaikissa käyttötarkoituksissa. Näitä suosituksia esitetään taulukossa 1. [Bartel et al. 2002]

	<b>Vaatus</b>	<b>Suositus</b>	<b>Vaihtoehtoinen</b>
Tiiviste	SHA1		
Koodaus	base64		
MAC	HMAC-SHA1		
Allekirjoitus	DSAwithSHA1 (DSS)	RSAwithSHA1	
Kanonisointi	Canonical XML	Canonical XML with comments	
Muunnos	Enveloped Signature	XPath	XSLT (eXtensible Stylesheet Language Transformations)

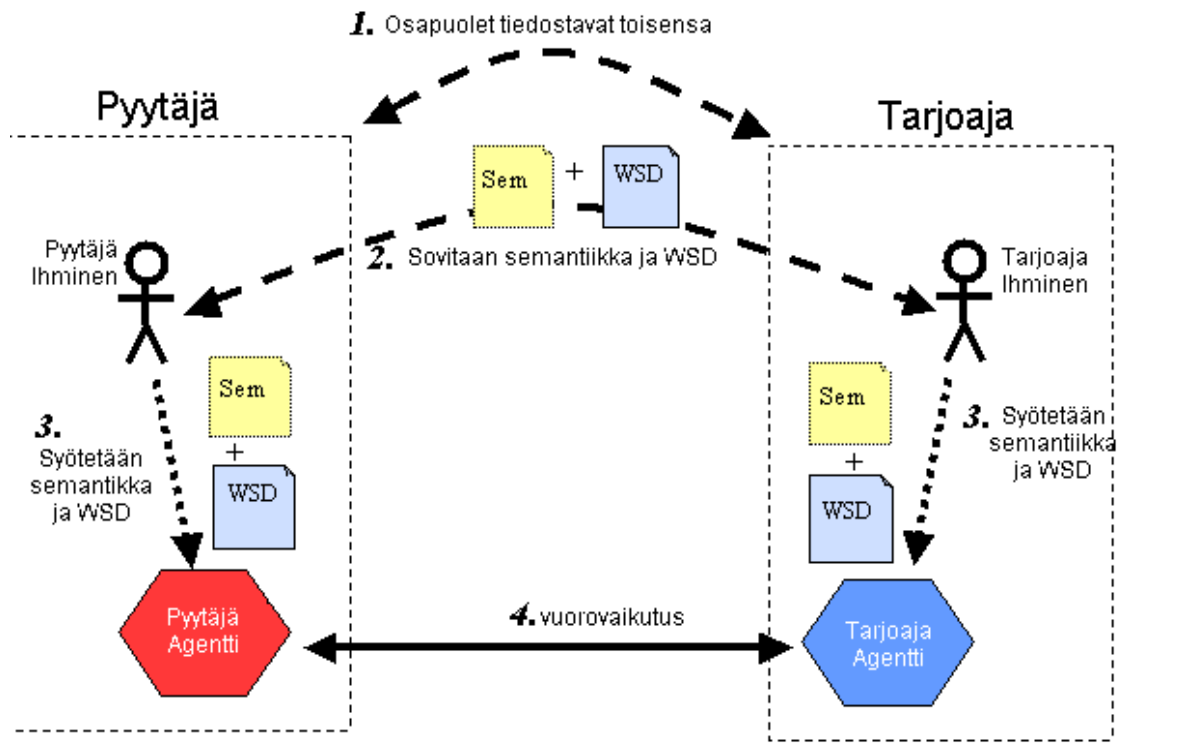
*Taulukko 1 W3C:n määritykset XML-allekirjoituksen kanssa käytettäville algoritmeille [Bartel et al. 2002]*

W3C ei suosittele käyttämään MD5-tiivistealgoritmia XML allekirjoituksen kanssa, sillä salaustekniikoiden kehitys, ja samalla niiden purkamisen kehittyminen, on luonut epäilyjä sen kestämisen suhteen. Vaikka vielä toistaiseksi MD5-algoritmin avulla luotua tiivistettä ei ole pystytty purkamaan, W3C:n suositus on, että XML allekirjoituksessa käytetään SHA-1 -algoritmia. Viestin autenttisuuden tarkastamiseen voidaan käyttää MAC - algoritmia (Message Authentication Code), jolla lasketaan tietynlainen tarkistesumma. Allekirjoitusta ja MAC:ia varten käytettävät algoritmit ovat syntaksiltaan samanlaisia, mutta MAC hyödyntää jaettua salaista avainta. HMAC-SHA1 (keyed Hash Message Authentication Code – SHA1) on algoritmi, jossa MAC lasketaan käyttämällä sekä salaista avainta että tiivistealgoritmia, joka tässä tapauksessa on SHA-1. Algoritmia käyttämällä voidaan varmistua viestin eheydestä ja aitoudesta. Kuten muissakin allekirjoituksissa, myös XML allekirjoituksissa on huomioitava, että ainoastaan se mitä allekirjoitetaan, on turvattu. Esimerkiksi kirjoitettaessa tavallista kirjettä itse kirje allekirjoitetaan, mutta kirjekuorta ei. Näin ollen kirjekuori on altis muutoksille ja vain allekirjoitettu kirje on turvattu. Sama pätee XML allekirjoitukselle. On myös hyvä ymmärtää mitä allekirjoittaa ja allekirjoittaa vain se mitä itse näkee. Sovellukset ja muut ohjelmat saattavat toisinaan lisätä informaation jotain, esimerkiksi automaattisia muotoiluja, profiileja tai suodattimia. On siis otettava huomioon, että allekirjoitettaessa viesti allekirjoitetaan myös nämä tiedot, ellei niitä erikseen erotella. [Bartel et al. 2002]

## **4.6 Web-palvelut**

Web-palvelut ovat eräänlaisia ohjelmistojärjestelmiä, jotka on suunniteltu tukemaan koneiden välistä yhteistyötä verkon välityksellä. Niillä on rajapinta, joka kuvataan konekielellä, esimerkiksi WSDL (Web Services Description Language) -kielellä. Muut laitteet kommunikoivat rajapinnan määrittämällä tavalla käyttämällä viestejä, jotka yleensä ovat HTTP:n (HyperText Transfer Protocol) ja XML:n (eXtensible Markup Language) avulla kuljetettavia SOAP (Simple Object Access Protocol) –viestejä. Web-palveluiden käyttämiseen tarvitaan agentti. Agentti on se osa järjestelmää, joka lähettää ja vastaanottaa viestejä. Yhdellä palvelulla voi olla monta agenttia, jotka saattavat toimia hieman erilailla

vaikka itse palvelu pysyy koko ajan samanlaisena. Web-palvelun etuna on, että se on alusta- ja ohjelmointikieliriippumaton, joten palvelun omistaja ja sen käyttäjä voivat toimia täysin erilaisissa ympäristöissä. Toinen etu on, että Web-palvelujen liikenne kulkee pääsääntöisesti HTTP:n avulla, jolloin välityspalvelimista ja palomureista ei aiheudu haittaa. Web-palvelun toimintaperiaatetta esitetään kuvassa 10. [Booth et al. 2004]

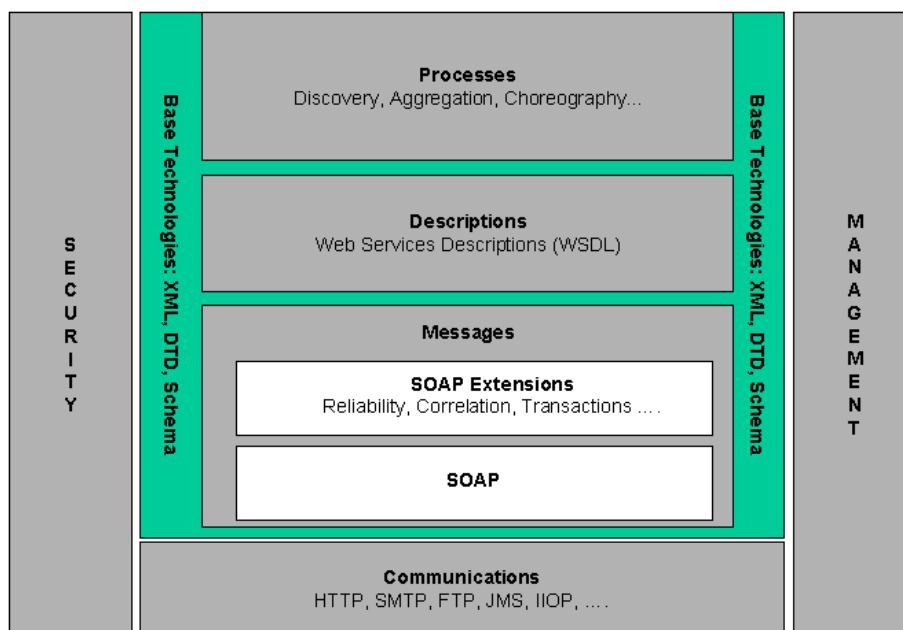


Kuva 10 Web-palvelun käynnistäminen [Booth et al. 2004]

Web-palvelun tarkoituksena on tuottaa jokin palvelu sen omistajan puolesta. Kun Web-palvelua aletaan käyttää, on osapuolien ensin oltava tietoisia toisistaan. Tämä tapahtuu, kun toinen osapuolista ottaa yhteyttä toiseen, jotta palvelu voidaan käynnistää (kuva 10, 1.). Web-palveluita voidaan löytää muun muassa käyttämällä etsintäpalveluja, jotka ovat Web-palveluja myös. Kun haluttu palvelu on löydetty, on niiden pystyttävä kommunikoimaan keskenään. Jotta Web-palvelut voivat vaihtaa viestejä, on palvelun tuottajan ja sen

käyttäjän ensin sovittava yhteisestä kielestä ja toimitavasta (kuva 10, 2.) ja nämä on oltava selvillä molemmilla osapuolilla, vaikka ne eivät lopulta vaihtaisi yhtään viestiä. Palvelun kuvaus ja semantiikka voivat olla esitettyinä jo etukäteen, ne voivat olla standardin mukaisia tai osapuolet voivat keskustella keskenään selvittääkseen ne. Viestien mekaniikka esitetään WSD (Web Service Description) –dokumentteina, jotka ovat konekielisiä määrittelyjä palvelun rajapinnasta. WSD-dokumentissa kerrotaan viestien muoto, datatyypit, siirtoprotokollat ja kuljetuksen jaksotusmuoto sekä sijainti, josta agenttia voidaan kutsua. Kun toimintatavat ovat selvillä, ne kerrotaan agenteille, jotta nämä voivat kommunikoida keskenään (kuva 10, 3.). Tämän jälkeen agentit jatkavat kommunikointia omistajiensa puolesta vaihtamalla SOAP-viestejä (kuva 10, 4.). [Booth et al. 2004; Sotomayor 2005]

Web-palvelu voidaan rakentaa monella eri tavalla ja se perustuu useaan erilliseen kerroksittaiseen toisiinsa yhteydessä oleviin teknologioihin. Kuva 11 esittää eräänlaista Web-palveluteknologiaa kuvaavaa arkkitehtuuripinoa. Siinä näkyvät yleisesti käytössä olevat viestien välittämiseen käytettävä SOAP- protokolla (Simple Object Access Protocol), palvelun kuvauskieli WSDL (Web Services Description Language), pohjateknologiat XML (eXtensible Markup Language), DTD (Document Type Definition) ja Schema, sekä tiedonvälitysprotokollat kuten HTTP (HyperText Transfer Protocol), SMTP (Simple Mail Transfer Protocol) ja FTP (File Transfer Protocol). Osittain useista erilaisista käytettävistä teknologioista johtuen, Web-palveluilla ei ole yhtään laajassa mittakaavassa käytössä olevaa turvallisuusmenetelmää Web-palveluille. On tietenkin oleellista, että palveluita voidaan käyttää turvallisesti, joten palveluiden kehittäjät voivat itse päättää käytettävistä turvallisuusmekanismeista. Eräitä XML-pohjaisia turvallisuusmekanismeja voidaan käyttää autentikoitumiseen, roolipohjaiseen käytönvalvontaan ja muihin välittäjät huomioon ottaviin seikkoihin. [Booth et al. 2004; Sotomayor 2005]



*Kuva 12 Web-palvelun arkkitehtuuria kuvaava pino [Booth et al. 2004]*

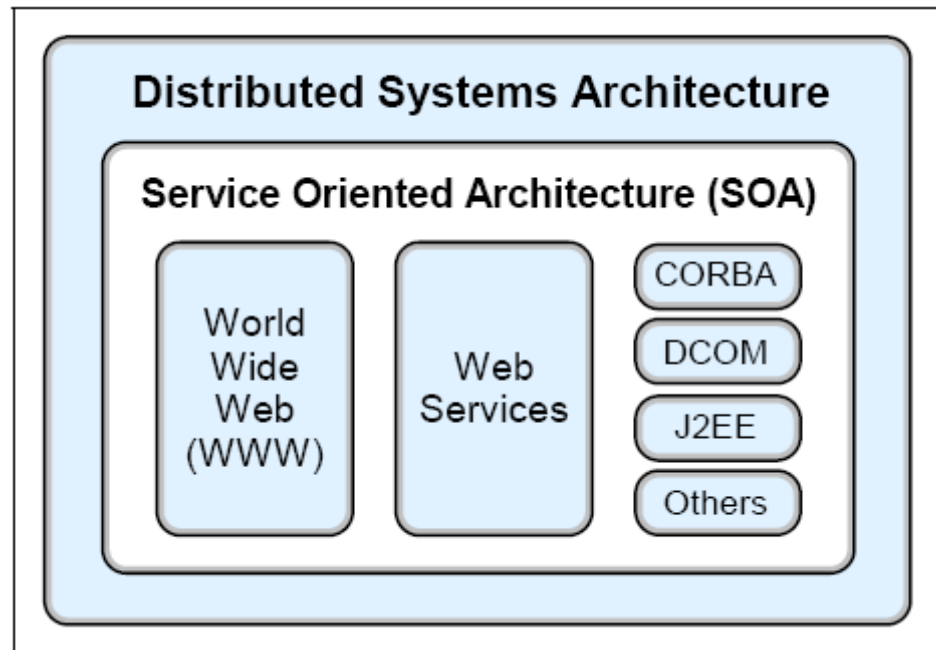
Yksi erikoinen ominaisuus Web-palveluissa on niiden kyky kuvailla itse itseään. Kun Web-palvelua kutsutaan, siltä voidaan samalla kysyä millainen palvelu se on ja miten se toimii. Tämä tapahtuu WSDL-kielen avulla, eli konekielisenä ohjelmistojen välillä. Palvelujen välillä on pystyttävä kommunikoimaan ja osapuolten on tällöin vaihdettava keskenään viestejä. Vaikka periaatteessa myös muut protokollat voisivat kommunikoida Web-palvelujen välillä, suurimmassa käytössä on SOAP, joka määrittää samalla miten kyselyt palvelimelle tulee muotoilla ja kuinka palvelimen tulisi vastata. Kaikki tämä liikenne kulkee palvelimen eli palvelun ja asiakkaan välillä yleensä HTTP:n avulla, vaikka tässäkin tapauksessa myös muut protokollat (esimerkiksi yllä mainitut SMTP ja FTP) ovat täysin mahdollisia. [Sotomayor 2005]

Web-palveluiden haittapuolena ovat tehokkuuden aleneminen ja monipuolisuuden puute. Kun kaikki tarvittava tieto siirretään XML:nä, on se hitaampaa ja vähemmän tehokasta kuin tavallisen binäärikoodin siirtäminen ja saattaa aiheuttaa viivettä. Tavallisimmissa Web-palveluissa tämä on kuitenkin hyväksyttävää, sillä ne eivät toimi reaaliajassa. Web-palvelut eivät myöskään toistaiseksi vielä salli kovin monimuotoisia toimintoja, mutta ne kehittyvät

vähitellen koko ajan ja niiden käyttömahdollisuudet lisääntyvät. Muista vastaavanlaisista palveluista Web-palvelut erotuvat muodostamalla omistajan ja käyttäjän välille vain löysän yhteyden (loosely coupled system), eli heillä ei tarvitse välttämättä olla etukäteen mitään tietoa toisistaan ja ne eivät ole toisistaan riippuvaisia. Tällainen järjestelmä toimii hyvin suurissa verkoissa, kuten internetissä. [Sotomayor 2005]

#### **4.6.1 Palvelukeskeinen arkkitehtuuri**

Jaetut järjestelmät, kuten Web-palvelut, koostuvat useista erilaisista ja toisistaan irrallisista ohjelmistoagenteista, joiden on toimittava yhdessä tiettyjen tehtävien suorittamiseksi. Jokainen ohjelmistoagentti voi lisäksi toimia erilaisessa järjestelmässä ja käyttää erilaisia prosesseja toimintaansa ja niiden on silti pystyttävä kommunikoimaan keskenään verkon välityksellä. Erilaiset toimintaympäristöt luovat viivettä ja mahdollisia puutteita käyttövarmuuteen ja täten haasteita arkkitehtuurien suunnittelijoille. Palvelukeskeinen arkkitehtuuri eli Service Oriented Architecture (SOA) on ohjelmistoarkkitehtuuri, joka määrittää palveluiden käyttöä ohjelmistovaatimuksien tukemiseen. Toisin sanoen SOA määrittää Web-palveluiden käyttöä, komponentteja ja niiden yhteyksiä, järjestelmää sekä sen suunnittelua, ja onkin suurempi kokonaisuus kuin pelkkä Web-palvelu. Tästä huolimatta Web-palveluiden käyttäminen ei edellytä SOA:n käyttämistä ja SOA:n toteuttamiseen on myös useita muita vaihtoehtoja kuin Web-palvelut. SOA ja web-palvelut soveltuvat kuitenkin hyvin palveluille, joita käytetään internetin välityksellä missä toimintavarmuutta ja nopeutta ei voida taata, joissa järjestelmän eri osat ovat erilaisilla alustoilla ja joissa sovellus voidaan paketoita Web-palveluksi muiden käyttöön. Kuvassa 12 on esitetty jaettujen järjestelmien arkkitehtuuria, siihen sisältyvä palvelukeskeinen arkkitehtuuri (SOA) ja SOA:n avulla rakentuvia palvelukeskeisiä järjestelmiä, kuten esimerkiksi Web-palvelut, internet ja J2EE (Java 2 Platform, Enterprise Edition). [Endrei et al. 2004 s.27-28; Booth et al.]



*Kuva 13 SOA:n erilaisia toteutusvaihtoehtoja [Endrei et al. 2004 s.28]*

SOA rakentuu kahdenlaisista elementeistä: toiminnallisista ja palvelun laadullisista. Toiminnallisia elementtejä ovat Web-palvelujen arkkitehtuurissakin esillä olleet siirtotie (esim. HTTP), palvelukommunikaatioprotokolla (SOAP), palvelun kuvaus (WSDL) ja lisäksi itse palvelu (Service), liiketoimintaprosessi (Business Process) ja palveluarkisto (Service Registry). Nämä elementit vastaavat koko järjestelmän toiminnasta kun taas palvelun laadun elementit hoitavat järjestelmän palvelun laatua. Näitä elementtejä ovat menettelytapa (Policy), tietoturva (Security), liiketapahtuma (Transaction) ja hallinta (Management). Toimintatapa määrittää miten palveluita saadaan niiden asiakkaiden ulottuville, mikä on osaltaan myös toiminnallista tapahtumaa ja toimintatapa onkin osaltaan molempia. Tietoturvassa määritetään palvelun tietoturvapoliittikka, eli esimerkiksi käyttäjien autentikointi tai tunnistaminen. Liiketapahtuman on tarkoitus valvoa, että kaikki tarvittavat palvelut suoritetaan asianmukaisella tavalla. Virheen sattuessa liiketapahtumaelementti ilmoittaa asiasta käyttäjälle. Hallinta puolestaan valvoo kokonaisen palvelun toimintaa ja vastaa mahdollisten virheiden hallinnasta. [Endrei et al. 2004 s.26]

Toimintaa SOA-järjestelmässä tapahtuu kolmen eri menettelytavan turvin. Ensimmäisenä tapahtuu *palvelun kuvaaminen ja julkistaminen*, jolloin palvelun käyttäjä etsii haluamaansa palvelua palveluarkistosta (Service Registry tai Service Locator). Täältä käyttäjä saa vastauksena palvelun osoitteen ja kuvauksen rajapinnoista, minkä jälkeen käyttäjä voi kutsua palvelua rajapintojen määrittelemällä tavalla. Tämä on *palvelun paikantaminen*. Kutsu voidaan lähettää joko palvelun välittäjälle tai suoran sen tarjoajalle, ja kolmantena suoritetaan *palvelun käyttäminen* jälleen palvelukuvauksen kertomalla tavalla. Kuten jo nimestäkin voidaan päätellä, palvelukeskeisen arkkitehtuurin pääosassa ovat palvelut. Täten näille asetetaan tiettyjä arkkitehtuurin kannalta oleellisia vaatimuksia. Toiminnan kannalta suurin vaatimus palveluille on rajapinnan alustariippumattomuus. On olennaista, että palvelua voidaan käyttää miltä tahansa alustalta palvelun tarjoajan alustasta huolimatta, eli käyttöjärjestelmällä ja ohjelmointikielellä ei voi olla merkitystä palvelun toimintaan. Alustariippumattomuus voidaan toteuttaa vaivattomimmin käyttämällä viestinvälitykseen ja kommunikaatioon yleisiä standardeja kuten HTTP, XML ja SOAP. Lisäksi palveluiden tulee olla täydellisiä ja riippumattomia toisistaan sekä yhdistetty löyhin yhteyksin (loosely coupled). Tämä tarkoittaa, että palvelut suoritetaan aina kokonaisuudessaan eikä osissa ja käyttäjät eivät ole sitoutuneita mihinkään tiettyyn palveluun, vaan yhteydet muodostetaan rajapintojen avulla tarvittaessa. [Endrei et al. 2004 s. 21-51]

Jotta palveluita voidaan käyttää, on niiden oltava käyttäjien ja muiden palvelujen saatavilla. Tätä varten käytetään palveluarkistoja, jotka arkistovat palvelunkuvauksia ja välittävät niitä eteenpäin kyselijöille. Jokainen palveluntarjoaja voi lähettää tiedon omasta palvelustaan arkistolle, minkä jälkeen käyttäjät voivat löytää ne etsimällä arkistosta. Kuvausten välittäminen on pääsääntöisesti dynaamista, mutta jotkut tarjoavat myös hakupalveluja. SOA-järjestelmä voidaan toteuttaa myös ilman palveluarkistoa, mutta se ei vastaa SOA-mallin periaatteita, joten suunnittelu palveluarkiston kanssa on suotavampaa. Arkiston tulisi olla yleiskäyttöinen, eli kieliriippumaton ja rajapintojen avulla käytettävä, minkä takia se voidaan toteuttaa usealla eri tavalla. Yleisimpiä käyttötapoja ovat suorat pyynnöt, XML-arkistot ja UDDI (Universal Description, Discovery and Integration)-arkistot. Palveluarkistot voivat olla palveluntarjoajan ylläpitämiä tai erillisellä palvelimella.



Joka tapauksessa ne sisältävät palvelujen rajapinnat kullekin palvelulle ominaisessa muodossa ja pyydettyä ne palautetaan käyttäjälle. Koska palveluarkistojen pitää olla toteutettuna palvelujen lailla, ne pohjautuvat yleensä standardeihin. [Endrei et al. 2004 s.139-144]

#### **4.6.2 SOAP - protokolla**

Simple Object Access Protocol eli SOAP on kevyt hajautettuun ympäristöön tarkoitettu tiedonsiirtoprotokolla sovellusten väliseen kommunikointiin. Se pohjautuu XML:ään ja koostuu kolmesta osasta: kirjekuoresta (envelope), koodaussäännöistä (encoding rules) ja etäkutsujen merkintäkäytännöstä. SOAP kirjekuori määrittää kehyksen viestin kuvaamiselle ja sen käsittelylle, eli sen mitä viesti sisältää, kenen viestiä tulee käsitellä ja onko se pakollista vai ei. Koodaussäännöt ilmaisevat mekanismin, jolla voidaan vaihtaa sovellusmääriteltäviä tietotyyppisiä, ja merkintäkäytäntö määrittää miten etäkutsuja ja vastauksia tulee merkitä. SOAP on suunniteltu viestien välitykseen internetissä ja se tarkoitettu erittäin yksinkertaiseksi ja helposti laajennettavaksi protokollaksi. Tästä johtuen joistakin tavallisimpien viestinvälitysprotokollien ominaisuuksista, kuten viestien kokoamisesta ja yhteislähetystä, on luovuttu SOAP-määrittelyissä. SOAP on alusta- ja kieliriippumaton, joten sitä on XML:n tavoin helppo käyttää minkä tahansa teknologian tai ohjelmointikielen kanssa. Nykyisin useat sovellukset kommunikoivat käyttämällä etäkutsuja (Remote Procedure Call, RPC) eri objektien välillä. Etäkutsuilla on kuitenkin se huono puoli, että välityspalvelimet ja palomuurit yleensä estävät kaiken sellaisen liikenteen, mikä saattaa luoda yhteensopivuus- ja turvaongelmia. SOAP on alun perin suunniteltu tuottamaan XML-pohjaisia etäkutsupalveluja verkon tietokoneille, mutta se käyttääkin tiedonsiirrossa hyödykseen HTTP:tä (tai SMTP:tä), jota kaikki internetselaimet ja palvelimet tukevat, joten sen avulla voidaan kiertää välityspalvelin- ja palomuuriongelmat. [Box et al. 2000; Gudgin et al. 2001; SOAP 2006]

Alla olevassa esimerkissä esitetään SOAP viesti, jossa ensin esiintyvä pakollinen `env:Envelope` – elementti määrittää kyseisen XML dokumentin SOAP viestiksi. Viesti

sisältää ala-elementit `env:Header` ja `env:Body`. Käytettävä sovellus määrittää sisällön näille elementeille, mutta niillä on kuitenkin jotain tekemistä itse SOAP viestin ja sen käsittelyn kanssa. Vaikka SOAP Header, eli otsikko, on valinnainen elementti, sen avulla voidaan siirtää tietynlaista valvontatietoa, joka ei kuulu sovelluksen luomaan viestiosaan. Tällaista tietoa voivat olla muun muassa toimintaohjeet tai viestin käsittelyyn liittyvät tiedot. SOAP Body on pakollinen osa SOAP viestiä ja se sisältää kaiken pääasiallisen tiedon, jota viestissä kuljetetaan, eli esimerkiksi sähköisessä reseptissä potilaan ja lääkkeen tiedot. Alla olevassa esimerkissä esitetään ensin SOAP-pyyntö ja siihen SOAP-vastaus, jotka on lähetetty käyttämällä HTTP-protokollaa. SOAP-pyyntöissä haetaan reseptiä käyttämällä `HaeResepti-proseduuria`, missä reseptin hakuperusteena (parametrina) toimii potilaan henkilötunnus (`hetu`). Vastauksena saadaan reseptin tiedot, joista esimerkissä näkyvät reseptin numero ja potilaan nimi. [Mitra 2003; Singh & Huhns 2005 s.21-26]

#### SOAP-pyyntö:

```
<!-- - HTTP-otsikon kentät - -->
POST /temp http/1.1
Host: www.reseptitietokanta.kela
Content-Type: text/xml; charset="utf-8"
Content-Length: xxx
SOAPAction: "http://www.reseptitietokanta.kela/resepti"

<!-- - SOAP-viesti - -->
<?xml version="1.0"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <r:resepti
      xmlns:r="http://www.reseptitietokanta.esim.fi/resepti"
      env:role=" http://www.w3.org/2003/05/soap-envelope/role/next"
      <r:msgID>01234-56789-0</r:msgID>
      env:encodingStyle=" http://www.w3.org/2003/05/soap-
        envelope/resepti.xsd"
    </r:resepti>
  </env:Header>
```

```

    <env:Body>
      <m:HaeResepti xmlns:m="
        http://www.reseptitietokanta.kela/resepti.xsd"
        <m:hetu>010234-567A</m:hetu>
      </m:HaeResepti>
    </env:Body>
  </env:Envelope>

```

### SOAP-vastaus:

```

<!-- - HTTP-otsikon kentät - ->
HTTP /1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: xxx
SOAPAction: "http://www.reseptitietokanta.kela/resepti"

<!-- - SOAP-viesti - ->
<?xml version="1.0"?>
<env:Envelope          xmlns:env="http://www.w3.org/2003/05/soap-envelope"
env:encodingStyle=" http://www.w3.org/2003/05/soap-envelope/resepti.xsd">
  <env:Header>
    <r:resepti
      xmlns:r="http://www.reseptitietokanta.esim.fi/resepti"
      env:role=" http://www.w3.org/2003/05/soap-envelope/role/next"
      <r:msgID>01234-56789-0</r:msgID>
    </r:resepti>
  </env:Header>

  <env:Body>
    <m:HaeReseptiVastaus xmlns:m="
      http://www.reseptitietokanta.kela/resepti.xsd"
      <reseptinro>123</reseptinro>
      <etunimi>Aino</etunimi>
      .
      .
      .
    </m:HaeReseptiVastaus>
  </env:Body>
</env:Envelope>

```

SOAP viestin esimerkissä on ensin HTTP-osa, johon SOAP-viesti on paketoitu. SOAP-otsikko on jaettu osiin, jotka käyttävät omaa nimiavaruutta ja se sisältää yleisimmin tietoa, joka on tarkoitettu viestin väli- tai päätepisteille. Yllä olevassa esimerkissä otsikossa on otsikossa viestin tunnistenumero 01234-56789-0. `Env:Role` -attribuutti ilmaisee, että lohkot tulee käsitellä ensimmäisessä polulla olevassa solmussa, eli välittäjässä (intermediary). Jollei matkalla ole yhtään solmua, lohkot käsittelee viestin vastaanottaja. Runkoelementillä `env:Body` on tytärelementtinä pyynnössä `HaeResepti` ja vastauksessa `HaeReseptiVastaus`, joilla on tarkoitus vaihtaa tietoa lähettäjän ja vastaanottajan välillä. `HaeResepti` pyytää etsimään reseptin, joka täyttää parametrin `hetu` arvon ja `HaeReseptiVastaus` sisältää vastauksen pyyntöön. [Mitra 2003; Singh & Huhns 2005 s. 21-26]

Mahdollisia viestinkäsittelyssä tapahtuvia virheitä varten SOAP:lla on virhemalli, jonka mukaan kaikki SOAP:lle ja sovelluksille ominaiset virheet tulee raportoida yhden ainoan elementin alle. Tämä on `env:Body` - elementin alle sisällytettävä `env:Fault`, joka sisältää kaksi pakollista ala-elementtiä, `env:Code` ja `env:Reason`, sekä vaihtoehtoiset `env:Detail`-, `env:Node`- ja `env:Role`- elementit. `env:Code` sisältää pakollisen `env:Value`- sekä vaihtoehtoisen `env:Subcode` - ala-elementin. Vaihtoehtoiset ala-elementit kertovat sovellukselle ominaista tietoa, missä solmussa virhe tapahtui sekä missä roolissa kyseinen solmu toimi. SOAP viestit kulkevat verkossa käyttämällä hyväkseen muita protokollia, kuten jo aiemmin mainittua HTTP:tä tai TCP:tä. Matkalla lähettäjältä vastaanottajalle SOAP viesti voi käyttää useitakin eri protokollia, joten sen kulkeminen näiden protokollien päällä tai avulla täytyy erikseen määritellä. Määrittelyä kutsutaan SOAP sidonnaksi (SOAP binding), ja siinä esitetään SOAP viesti eräänlaisena tietojoukkona (XML infoset), eli `env:Envelope`- dokumentin elementti- ja attribuuttitietoina. Käytettäessä HTTP:tä SOAP viesti saadaan muodostamalla HTTP -pyyntö tai vastaus SOAP:n koodaussääntöjen mukaan. [Mitra 2003]

### 4.6.3 WSDL - kieli

Tietoliikenneprotokollien ja viestimallien standardoitumisen myötä on yhä tärkeämpää, että hajautettujen järjestelmien kommunikointimenetelmiä voidaan rakenteellisesti kuvata. Web Services Description Language (WSDL) eli Web-palvelujen kuvauskieli on XML:llä kirjoitettu dokumentti, joka nimensä mukaisesti kuvaa Web-palveluja. Siinä määritellään palvelun sijainti ja operaatiot tai menetelmät, joita se käyttää. Tärkeimmät elementit WSDL dokumentissa ovat `portType`, `message`, `types`, `binding`, `port` ja `service`. Elementit määrittävät operaatiot, joita palvelu tuottaa sekä palvelun käyttämät viestit, tietotyypit ja kommunikaatioprotokollan. [Christensen et al. 2001]

Periaatteessa WSDL dokumentissa palvelut kuvataan ryhmänä verkon päätepisteitä tai portteina, jotta siinä voidaan käyttää määritelmiä, kuten jo mainitut `portType`, `messages` ja `binding`. WSDL ei myöskään ole uudenlainen kuvauskieli, vaan se hyödyntää XML Schemaa tietojen kuvaamiseen. Tämän lisäksi se pystyy mukautumaan erilaisiin ympäristöihin ja tulevaisuuden muutoksiin käyttämällä muuntotyypisiä kuvauskieliä laajennusten avulla. Koska WSDL käyttää XML:ää ja XML Schemaa, sen rakenne on näiden kaltainen. Juuressa on `definitions`-elementti, joka määrittää dokumentin WSDL dokumentiksi, ja sen sisällä ovat kaikki määritelmät. `portType`-elementti määrittää palvelun tuottamat toiminnot, joista jokainen viittaa yhteen input-viestiin ja yhteen output-viestiin. `Types` kuvaa tietotyyppejä, joita käytetään viestienvaihdon luonnehtimiseen ja `message` esittää abstraktin kuvauksen tiedosta, jota siirretään. `Binding` määrittää käytössä olevan protokollan sekä toimintojen ja viestien käyttämän tietomuodon. `Port` määrittää yhden tietyn kommunikaation päätepisteen eli osoitteen `binding`:lle ja `service`:ä hyödynnetään yhdistämään toisiinsa liittyvät portit. Alla on esimerkki, josta näkyy WSDL dokumentin rakenne ja käytössä olevat elementit. Esimerkissä on jatkettu SOAP-viestin esimerkkiä. [Christensen et al. 2001]

```
<?xml version="1.0"?>
<!-- wsd:definitions määrittelee joukon samankaltaisia palveluja-->
<wsdl:definitions name="Resepti"
```

```

targetNamespace="http://www.reseptitietokanta.kela/schema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wSDL=" http://schemas.xmlsoap.org/wsdl/">

<wSDL:types>
  <xsd:schema
targetNamespace="http://namespaces.reseptitietokanta.kela"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
    <xsd:element name="HaeResepti">
      <xsd:complexType>
        <xsd:element name="hetu" type="string"/>
      </xsd:complexType>
    </xsd:element>

    <xsd:element name="HaeReseptiVastaus">
      <xsd:complexType>
        <xsd:element name="reseptinro" type="string"/>
        <xsd:element name="etunimi" type="string"/>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
</wSDL:types>

<wSDL:message name="HaeResepti">
  <wSDL:part name="body" element="tsxsd:HaeResepti"?>
</wSDL:message>

<wSDL:message name="HaeReseptiVastaus">
  <wSDL:part name="body" element="tsxsd:HaeReseptiVastaus"?>
</wSDL:message>

<wSDL:portType name="HaeReseptiPortType">
  <wSDL:operation name="HaeResepti">
    <wSDL:input message="ts:HaeResepti"/>
    <wSDL:output message="ts:HaeReseptiVastaus"/>
  </wSDL:operation>
</wSDL:portType>

```

```

<wsdl:binding name="ReseptiSoapBinding" type="ts:HaeReseptiPortType">
<soap:binding style="document" transport="
http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="HaeResepti">
    <soap:operation
      soapAction="http://www.reseptitietokanta.kela/ReseptiSVC"/>
    <wsdl:input>
<soap:body use="literal"
namespace="http://schemas.xmlsoap.org/ReseptiSVC">
      </wsdl:input>
    </wsdl:operation>
  </wsdl:binding>

<wsdl:service name="ReseptiPalvelu">
  <wsdl:documentation>Kelan Reseptipalvelu</wsdl:documentation>
<wsdl:port name="HaeReseptiPort" binding="ts: ReseptiSoapBinding">
  </wsdl:port>
</wsdl:service>

</wsdl:definitions>

```

## 4.7 Liiketoimintamallit

Liiketoimintamallien avulla pyritään kuvaamaan liiketoimintaprosessia kirjallisesti. Tarkoituksena on havainnollistaa prosesseja, jotta kokonaisuudet, rakenteet ja asioiden väliset suhteet pystytään paremmin ymmärtämään. Prosessit muuttuvat yhä monimutkaisemmiksi, joten mallintamisen merkitys korostuu. Niiden kuvaamiseen käytetään nykyään usein mallinnuskieliä, jotka perustuvat tiettyihin säännöstöihin ja merkkijärjestelmiin. Liiketoimintaprosessien kuvaaminen onkin oleellinen osa palvelukokonaisuuksien luomista. Näiden mallintamiseen oli alun perin kaksi kilpailevaa menetelmää; IBM:n WSFL (Web Services Flow Language) ja Microsoftin XLANG. Nämä kuitenkin myöhemmin yhdistettiin ja muodostettiin XML-pohjainen määrittelykieli nimeltään BPEL4WS (Business Process Execution Language for Web Services). Toinen

samaan tarkoitukseen luotu mallinnuskieli on BPML (Business Process Modeling Language). [Andrews et al. 2003 s. 8-11; Singh & Huhns 2005 s. 256-262]

#### **4.7.1 BPEL4WS-määrittelykieli**

BPEL4WS-kielen tarkoituksena on palvelukuvauksien lisänä tarkentaa palveluiden välisiä yhteyksiä. Sitä voidaan käyttää sekä kuvauskielenä että toteutuskielenä liiketoimintaprosesseille. Tarkemmin sanottuna se määrittelee mallin ja kieliopin Web-palveluiden osapuolten välille, jotta liiketoiminta saadaan suoritettua halutulla tavalla. BPEL4WS-kielen yksi etu on siinä, että sen avulla voidaan määrittellä myös toiminnot virhetilanteiden varalle. BPEL4WS dokumentissa käytetään XML:llä kuvaamaan liiketoimintaprosessin ominaisuuksia. Näitä ovat muun muassa partnerit, säiliöt, muuttujat, virheenkäsittelijät ja tapauskäsittelijät. Partnerit ovat yksinkertaisesti ne osapuolet, jotka ovat Web-palvelussa mukana, säiliöt ovat palvelun käyttämiä tietovarastoja, joita tarvitaan esimerkiksi tilatietoja varten, muuttujat määrittelevät ne muuttujat, joita prosessi käyttää, virheenkäsittelijät määrittävät toiminnot virhetilanteissa ja tapauskäsittelijät määrittelevät toiminnot ulkopuolisille toimintoille. Dokumentti siis määrittää prosessiin osallistuvat jäsenet, käytettävät protokollatyypit, osien yhdistämisen ja reititysinformaation. [Andrews et al. 2003 s. 8-11; Singh & Huhns 2005 s. 256-262]

#### **4.7.2 BPML-mallinnuskieli**

BPML on metakieli liiketoimintaprosessien kuvaamiseen hieman samalla tavalla kuin XML on metakieli tiedon kuvaamiseen. Se määrittää XML sanaston ja tavan, jolla voidaan kuvata liiketoimintaprosessin ja siihen liittyvän tiedon abstrakti malli. Etuna on, että koko prosessi voidaan kuvata alusta loppuun, eikä ainoastaan yhden osapuolen näkökulmasta. BPML erottaa toisistaan hallintavirran ja tietovirran, jolloin prosessia voidaan helpommin ylläpitää ja käyttää uudelleen. BPML toimii samankaltaisesti kuin BPEL4WS:kin, mutta siitä on pyritty tekemään yksinkertaisempi ihmisten käyttöä varten. Sen avulla voidaan



mallintaa monimutkaisiakin liiketoimintaprosesseja vaivattomasti esimerkiksi muuttamalla UML-kaavioita sovellukselle ymmärrettävään muotoon. [Cover 2003]

## **4.8 Muita tekniikoita**

Sähköinen resepti Suomessa ja muualla maailmassa käyttää useita erilaisia tekniikoita. Suomessa tiedonsiirrossa on käytetty VPN (Virtual Private Network) ja SSL (Secure Sockets Layer) -tekniikoita. Myös Ruotsissa oli alun perin käytössä VPN, mutta myöhemmin siitä muutettiin VLAN:n (Virtual Local Area Network) käyttöön. Saksassa sähköinen resepti toimitetaan elektronisen henkilökortin avulla, jonka sirukorttitekniikasta kerrottiinkin kappaleessa 4.3. Tanskassa reseptin toteutus on hoidettu sitä varten kehitetyllä postistandardilla, jossa käytetään EDI (Electronic Data Interchange) - tekniikoita ja EDIFACT (EDI For Administration, Commerce and Transport) - standardia ja reseptiliikenne kulkee sekä vanhemmassa VANS (Value Added Network Service)- verkossa että IP-verkossa. Seuraavassa lyhyt kuvaus näistä tekniikoista.

### **4.8.1 VPN - tekniikka**

Virtual Private Network eli VPN on tekniikka, jolla julkisen verkon päälle voidaan rakentaa yksityinen verkko. VPN-verkko toteutetaan joko ohjelmistoilla tai laitteilla niin, että esimerkiksi yrityksen tai muun organisaation sisäinen verkko voidaan ulottaa julkisen verkon, kuten internetin yli turvallisesti. Jotta voidaan varmistua tietojen autenttisuudesta molemmat osapuolet, siis lähettäjä ja vastaanottaja todennetaan ennen liikennöinnin aloitusta. Tiedonsiirron turvaamiseksi lähetettävä tieto salataan, jotta kolmannet osapuolet eivät pääse siihen käsiksi. Liikenne lähetetään käyttämällä todennuksen ja salauksen kanssa tunnelointia. Se tarkoittaa käytännössä sitä, että lähetettävä tieto kapseloidaan pakettiin, joka käyttää eri protokollaa ja salaa näin liikenteen. Tunnelin käytön ansiosta kaikki liikenne voidaan lähettää turvallisesti, sillä sovellustason protokollat eivät vaikuta suojaukseen. [Perlmutter & Zarkower 2001 s.10-13; Viestintävirasto 2004]

VPN voidaan rakentaa usealla tavalla eri valmistajien, palveluntarjoajien ja ohjelmistoyritysten tuottamien ratkaisuiden ansiosta. Tämä voi olla hyvä asia siinä mielessä, että jokainen voi löytää joukosta itselleen tai organisaatiolleen juuri sopivan ratkaisun, mutta huono puoli on se, että sopivan ratkaisun löytäminen valtavasta määrästä vaihtoehtoja voi viedä aikaa ja aiheuttaa turhautumista. Kolmena yleisimpänä mallina voidaan pitää tarjoajamallia, sekamallia ja päästä-päähänmallia (End-to-End), jotka eroavat suurimmalta osalta palveluntarjoajan osallistumisen määrällä. Tarjoajamallissa palveluntarjoaja tekee suurimman työn ja liikenne kulkee tämän verkon kautta lisäämättä asiakkaan verkkoon mitään tai vain välttämättömät osat. Asiakkaan ja palveluntarjoajan verkot ovat täysin erilliset ja molemmat hallitsevat omia osiaan verkosta. Etuna on se, ettei asiakkaan tarvitse välittää ulkopuolisen verkon ylläpitämisestä eikä tarvitse mitään uusia ja kalliita laitteita ja palveluntarjoaja voi hoitaa verkkoa parhaaksi näkemällään tavalla. Haittapuolena on kuitenkin samaisen asian toinen puoli; asiakkaan on luotettava palveluntarjoajaan ja on tästä riippuvainen ja palveluntarjoaja on yksin vastuussa verkon toimivuudesta ja toimivan ratkaisun rakentamisesta. Sekamallissa sekä palveluntarjoaja että asiakas osallistuvat VPN toiminnallisuuteen. Tämä mahdollistaa turvallisen etäkäytön, kun käyttäjiä voidaan tunneloida samaan verkkoon myös fyysisen toimipisteen ulkopuolelta autentikoimalla käyttäjä tarvittaessa. Päästä-päähänmallissa palveluntarjoajalla on vähiten tehtävää. Tämän roolina on vain VPN tiedon kuljettaminen ja asiakas hoitaa itse pääsyn julkiseen verkkoon oman järjestelmänsä välityksellä. Näin voidaan helposti ja turvallisesti yhdistää useita eri toimipisteitä keskenään esimerkiksi internetin tai muun IP-yhteyden välityksellä. [Perlmutter & Zarkower 2001 s.86-99]

#### **4.8.2 SSL - tietoturvaprotokolla**

Toinen tapa siirtää tietoa turvallisesti turvattomassa verkossa on Secure Sockets Layer (SSL) –tietoturvaprotokolla. Myös SSL:n avulla voidaan autentikoida tiedonsiirron osapuolet sekä varmistaa tiedon eheys ja luotettava tiedonsiirto salaamalla sisältö. SSL-liikenteessä suojataan kaikki osapuolten välillä kulkeva tieto, kuten pyydetyn dokumentin URL ja sisältö, evästeet sekä HTTP-otsikoiden sisältö. SSL on sitä käyttävistä sovelluksista

riippumaton ja hyvin monipuolinen, sillä se toimii TCP- protokollan päällä, mutta kuitenkin sovellusten alla. Protokollassa käytetään symmetristä salausta, tiivistettä ja autentikointimenetelmää. Siinä käytetään vahvinta osapuolten yhteistä salausten menetelmää niin, että salausavainten koko määräytyy suurimman yhteisen tason mukaan. Osapuolet voivat todentaa toisensa luotetun kolmannen osapuolen myöntämän X.509-sertifikaatin avulla. Yhteys osapuolten välillä avataan vaiheittaisella neuvottelulla, jossa molemmat autentikoidaan ja samalla luodaan yhteinen istuntoavain liikenteen salaamista varten. Kun molemmat osapuolet ovat autentikoituja ja heillä on istuntoavain, voidaan liikennöinti aloittaa. [HL7 Finland 2005]

### **4.8.3 EDI ja EDIFACT**

Electronic Data Interchange (EDI) viittaa kahden organisaation tietojärjestelmien väliseen yhteyteen, jolla vaihdetaan ennalta määritetyssä muodossa olevaa liiketietoa suojatussa verkossa. Koska käytössä on ennalta määritetty sähköinen rakenne, virheiden määrä vähenee ja tietoja voidaan käsitellä nopeasti, tarkasti ja joustavasti. EDI viesti koostuu tietoelementeistä, joista jokainen esittää toisistaan erottimella rajattua yksittäistä tietoa ja yksi kokonainen viesti on yksi tietosegmentti. Kun yhteen tai useampaan tietosegmenttiin liitetään otsikko (header) ja lopuke (trailer), ne muodostavat tapahtumajoukon (transaction set), joka vastaa tyypillistä liikedokumenttia. EDI:n apuna käytetään lisäarvoverkkoja (Value-Added Network, VAN), joka toimii eräänlaisena postitoimistona viestejä välittävien organisaatioiden välillä ja mahdollistavat viestien lähettämisen milloin tahansa. VAN säilyttää sinne lähetetyt viestit kunnes vastaanottaja erikseen noutaa ne. [Minnesota Department of Labor & Industry 2002]

Euroopan yhteisö on kehittänyt EDI:n käyttöön EDIFACT (Electronic Data Interchange For Administration, Commerce and Transport) –standardin, joka on nykyisin myös ISO:n standardi (ISO 9735). EDIFACT määrittää rakenteelliselle tiedolle syntaksin mukaisen sääntöjoukon, interaktiivisen viestinvälitysprotokollan (I-EDI) ja vakiomuotoisen viestin. Standardin mukainen viestinvaihto koostuu palvelumerkkijonon neuvoista, viestinvaihdon

otsikosta, toiminnallisen ryhmän otsikosta, viestin otsikosta, käyttäjän tietosegmenteistä, viestin lopukkeesta, toiminnallisen ryhmän lopukkeesta ja viestinvaihdon lopukkeesta. Yksi yhteys voi sisältää yhden tai useamman viestinvaihdon. Tällä hetkellä XML valtaa alaa viestinvaihtostandardina ja voi olla, että se tulevaisuudessa syrjäyttää EDIFACT:n. EDIFACT viesti on kooltaan pienempi kuin vastaavanlainen XML viesti, mutta XML on helppolukuisempi ihmisille. XML:ää varten alkaa myös olla enemmän työkaluja kuin EDIFACT:lle, mikä lisää siirtymistä XML:n käyttöön. EDIFACT on kuitenkin vahvasti vielä käytössä varsinkin Euroopassa ja muualla vanhemmissa järjestelmissä esimerkiksi turismipalveluissa ja siviili-ilmailussa, mutta uudemmat teknologiaratkaisut käyttävät enemmän XML-standaria. [UN/EDIFACT 2006]

#### **4.8.4 VLAN - verkkotekniikka**

Virtual Local Area Network (VLAN) on hieman VPN:n verrattava verkkotekniikka. Myös siinä on tarkoitus rakentaa fyysisestä sijainnista riippumaton turvallinen verkko. VLAN:ssa olevat koneet siis toimivat aivan kuten tavallisessa lähiverkossa huolimatta siitä, että yksittäiset koneet tai LAN-segmentit voivat sijaita fyysisesti missä tahansa. Kuten VPN:llekin myös VLAN:n toteutukseen on useita erilaisia tekniikoita, joita eri myyjät markkinoivat. Ratkaisut voidaan karkeasti jakaa porttiryhmittelyyn, MAC-ryhmittelyyn (Media Access Control), verkkokerrosryhmittelyyn ja IP multicast-ryhmittelyyn. Porttiryhmittelyssä kaikki yhden tai useamman kytkimen portin koneet liitetään VLAN:iin. Esimerkiksi yhden kytkimen tapauksessa konfiguroidaan portit 1, 2 ja 3 VLAN A:han ja portit 4, 5 ja 6 VLAN B:hen tai useamman kytkimen tapauksessa VLAN A on kytkimen 1 portit 1 ja 3 ja kytkimen 2 portit 4 ja 5, ja VLAN B on kytkimen 1 portit 2 ja neljä sekä kytkimen 2 portit 1 ja 2. Porttiryhmittely on verrattain helppo konfiguroida ja toimiva tapa määrittellä VLAN, ainoana ongelmana voidaankin nähdä uudelleenkonfiguroinnit käyttäjien vaihtaessa kone portista toiseen. MAC-ryhmittelyssä etuna on, että koneen paikkaa voidaan tarvittaessa vaihtaa, sillä MAC osoite on koodattu laitteen verkkokorttiin, jolloin uudelleenkonfigurointia ei tarvita ja laite säilyy VLAN-verkossa sijainnista huolimatta. Haittana on, että alun perin kaikki käyttäjät on konfiguroitava ainakin yhteen VLAN:iin, ja

käyttäjää voi olla yhdessä LAN:ssa jopa tuhansia. Toinen, mutta jokseenkin vähäinen haitta ovat kannettavat, joita käytetään ajoittain telakka-aseilla. Tällöin molemmilla laitteilla on eri MAC-osoite, jolloin käyttäjää ei voida seurata vaan VLAN määräytyy telakka-aseman konfiguraation mukaan. [Passmore & Freeman 1996]

Verkkokerrokseen perustuvassa ryhmittelyssä VLAN:iin kuulumisen määritetään protokollan tai verkkokerroksen osoitteen mukaan. Vaikka käytössä on määrittely, joka perustuu verkkokerrokseen, ei kyseessä kuitenkaan ole verkkokerrokseen perustuva reititys. Jokainen VLAN nähdään tällöin tavallisena sillattuna topologiaa. Joitakin reititykseen perustuvia ominaisuuksia voidaan kuitenkin lisätä VLAN-ratkaisuihin. IP multicast-ryhmät ovat hieman erilainen tapa muodostaa VLAN. VLAN:n ideahan on broadcast-verkko, mutta IP-paketit voidaan lähettää multicastaamalla tietyille dynaamisesti määritetyille osoitteille ja näin VLAN:iin kuuluvien laitteiden IP-osoitteet voidaan lisätä tähän tiettyyn multicast-ryhmään. Jokainen VLAN:iin kuuluva työasema on kuitenkin multicast-ryhmän jäsen vain tietyn aikaa, mikä mahdollistaa joustavan toiminnan. Nykyisin myyjät yhdistelevät eri ratkaisuja saadakseen aikaan asiakkaille räätälöityjä, joustavia ja hyvin toimivia VLAN:eja. [Passmore & Freeman 1996]

## 5 SEMANTTISEN WEBIN TEKNIKOIDEN HYÖDYNTÄMINEN

Eri terveydenhuolto-organisaatioilla on erilaisia potilasrekisterijärjestelmiä ja nyt myös erilaisia sähköisen reseptin järjestelmiä. Lisäksi apteekeilla on omat järjestelmänsä, ja näiden kaikkien on pystyttävä toimimaan yhdessä sähköisen reseptin järjestelmän toiminnan takaamiseksi. Sähköisen reseptin pilotissa jo määritellään muun muassa XML:n käyttö reseptien siirrossa, jotta kaikki järjestelmät voivat niitä käsitellä. Reseptijärjestelmän toimiminen Web-palvelu –tekniikoiden avulla takaa eri järjestelmien yhteentoimivuuden ja reseptien liikkuvuuden niiden välillä. Semanttisen webin muutkin tekniikat tuovat lisäarvoa sähköiselle reseptijärjestelmälle muun muassa erilaisten kyselyjen muodossa.

### 5.1 XML:n käyttö sähköisessä reseptissä

Kuten alussa jo mainittiin, sähköisen reseptin pilotissakin on mainittu reseptien esittäminen ja siirtäminen XML-muodossa. Koska jokainen käyttäjä voi itse määrittää omat taginsa XML dokumenttiin, voi jokainen XML dokumentti olla hieman erinäköinen. Toimintaperiaate on kuitenkin kaikissa aivan sama; tiedot kuvataan rakenteellisella tavalla mahdollisimman helposti ymmärrettävässä muodossa. Alla on yksinkertaistettu esimerkki siitä, millainen sähköisen reseptin XML dokumentti voisi näyttää.

```
<?xml version="1.0" encoding="UTF-16"?>
<!DOCTYPE reseptirakenne SYSTEM "reseptirakenne.xsd">
<?xml:stylesheet type="text/css" href="tyylitiedosto.css">

<resepti>
  <potilas>
    <sotu>120345-6789</sotu>
    <nimi>
      <etunimi>Pirkko</etunimi>
      <sukunimi>Potilas</sukunimi>
    </nimi>
```

```

    <paino></paino>
    <!--paino ainoastaan alle 12-vuotiailta -->
    <yhteystiedot>Kotipolku 1 12345 Kaupunki</yhteystiedot>
    <kieli>suomi</kieli>
</potilas>

<laake>
    <nimike>Ibuprofein</nimike>
    <!-- Anatomical-Therapeutic-Chemical-luokka -->
    <ATC>M01AE01</ATC>
    <!-- tuotenumero -->
    <VNR>123456</VNR>
    <myyntilupa></myyntilupa>
    <vahvuus>200mg</vahvuus>
    <pakkauskoiko>10 kpl</pakkauskoiko>
    <pakkaus_lkm>1</pakkaus_lkm>
    <laakemuoto>tabletti</laakemuoto>
    <toimitettava_maara>1</toimitettava_maara>
    <kesto>10 päivää</kesto>
    <vaihdonKielto></vaihdonKielto>
    <rajoitukset></rajoitukset>
    <annostus>
        <annostelu>vatsakipuun 1 tabletti aamuin ja illoin
        5 päivän ajan</annostelu>
        <kayttotarkoitus></kayttotarkoitus>
        <lisaohje></lisaohje>
        <sic></sic>
        <vrk_annos></vrk_annos>
    </annostus>
    <erillisselvitys></erillisselvitys>
</laake>

<lisatietoja>
    <pysyva/tilapainen></pysyva/tilapainen>
    <uusi_hoito></uusi_hoito>
    <viesti></viesti>
    <neRep></neRep>

```

```
</lisatietoja>

<laakari>
  <laakarinNimi>
    <etunimi>Timo</etunimi>
    <sukunimi>Tohtori</sukunimi>
  </laakarinNimi>
  <myontamisPaikka>Sairaala</myontamisPaikka>
  <myontamisPvm>04.05.2006</myontamisPvm>
  <svnro>SV 123456</svnro>
  <oppiarvo>LL</oppiarvo>
  <erikoisala>työterveyshuolto</erikoisala>
  <allekirjoitus></allekirjoitus>
</laakari>

<palkkio>
  <summa></summa>
  <laakarin_toimi></laakarin_toimi>
</palkkio>

<apteekki>
  <kommentti></kommentti>
  <oikaisu></oikaisu>
  <pvm></pvm>
  <allekirjoitus></allekirjoitus>
</apteekki>

<toimitus>
  <apteekki_nro>I Apteeeki 123456</apteekki_nro>
  <potilas>Pirkko Potilas</potilas>
  <toimitusaika>05.05.2006</toimitusaika>
  <toimitusmaara>10 kpl</toimitusmaara>
  <valmiste></valmiste>
  <status>kaikki toimitettu</status>
  <toimittaja>
    <etunimi>Aino</etunimi>
    <sukunimi>Apteeekkari</sukunimi>
  </toimittaja>
```



```
<reseptinLaji>A</reseptinLaji>
<reseptinro></reseptinro>
<hinta></hinta>
<korvausLuokka></korvausLuokka>
<allekirjoitus></allekirjoitus>
</toimitus>

<tunniste></tunniste>
</resepti>
```

Juurielementtinä on <resepti> ja sen alla reseptissä oltavat tiedot potilas, lääke, lääkäri, lisätiedot, palkkio, apteekki, toimitus sekä reseptin tunniste. Potilaasta täytyy merkitä nimi ja sosiaaliturvatunnus, jonka perusteella potilas voidaan yksilöidä sekä alle 12-vuotiailta paino. Erona tavalliseen paperireseptiin potilaan tietoihin merkitään myös potilaan yhteystiedot, jotka saadaan potilastietojärjestelmästä ja asiointikieli. Lääkkeestä merkitään lääkkeen myyntinimi, lääkeaineen nimi tai koostumus, mikäli lääke valmistetaan apteekissa (nimike). Myyntiluvallisille valmisteille pakollinen ATC (Anatomical Therapeutic Chemical) –koodi ja VNR (varenummer, pohjoismainen tuotenumero) –numero saadaan lääketietokannasta. Jos lääkäri haluaa määrätä tietyn myyntiluvan haltijan valmistetta, merkitään tämän nimi. Vahvuus ja lääkemuofo (esimerkiksi tabletti, emulsiovoide, tippa, silmälamelli) merkitään paperireseptin tapaan. Lääkemäärän tilalla merkitään pakkauskooko tai seoksen määrä. Mikäli pakkausten lukumääräksi merkitään nolla, ovat pakkausten lukumäärä, toimitettava määrä ja lääkehoidon kesto pakollisia tietoja. Reseptillä on lisäksi mainittava toimitusrajoitukset (esimerkiksi iterointi), reseptin tarkoitus (sairauden hoito/muu/työtaturma) ja erillisselvitys. Mikäli kyseessä on työtaturma, tulee mainita myös työnantaja tai vakuutusyhtiö ja erillisselvitys kirjoitetaan, mikäli lääkkeen korvaaminen edellyttää sitä. [Koponen-Piironen 2001]

Annostukseen tulee merkitä kuinka usein ja monta annosta lääkettä otetaan (esimerkiksi yksi tabletti kaksi kertaa päivässä), käyttöohje ja tyyppi (säännöllinen vai tarvittaessa) sekä käyttötarkoitus ellei sen poisjättämiselle ole vahvoja perusteita. Tarvittaessa tarkempi

selvitys tai muita tietoja lääkkeestä voidaan kirjoittaa lisäohjeeseen. Annostusohje voidaan tarvittaessa koodata, jolloin käyttötarkoitus, lisäohje ja vuorokausiannos on merkittävä erikseen, muuten kaikki voivat olla samassa kentässä. Annostukseen merkitään myös mahdollinen sic-merkintä, jos valmisteen hyväksyty annostusohje ylittää lääkärin määräämän annostuksen. Lisätietoihin merkitään onko kyseessä pysyväis- vai tilapäislääkitys ja uusi lääkitys. Lisäksi merkitään lääkärin viestit apteekille tai potilaalle ja uusimisen kieltävä ne rep –merkintä koodattuna. [Koponen-Piironen 2001]

Lääkärin on aina allekirjoitettava resepti sähköisesti. Lisäksi lääkäri merkitsee SV (sairausvakuutus) –numeronsa, reseptikirjoituspäivämäärän ja -toimipaikan sekä yhteystietonsa. Mikäli lääke määrätään jossakin muualla kuin vastaanotolla, merkitsee lääkäri lisäksi palkkion määrän ja onko lääke määrätty erikois- vai yleislääkärinä. Apteekissa reseptille merkitään mahdolliset korjausmerkinnät ja erilliset toimitusmerkinnät. Jos reseptin tietoja joudutaan korjaamaan, ne talletetaan aina erillisinä. Korjauksissa on ensin vapaaehtoinen kommenttikenttä, johon voidaan kertoa tarkemmin esimerkiksi miksi lääkärille on soitettu tai miksi reseptiä on jouduttu korjaamaan. Lisäksi merkitään itse oikaisu, korjauspäivä ja lopulta resepti allekirjoitetaan sähköisesti. Toimittamismerkintöjä ovat apteekin nimi ja sen numero, potilas tai reseptin hakija, toimituspäivämäärä, valmisteen tuotenumero ja hinta, toimittamatta oleva määrä, reseptin laji, reseptin numero ja lääkkeen luovuttajan sähköinen allekirjoitus. Reseptin lajiin merkitään onko tämä ensimmäinen toimitus vai myöhempi toimitus ja reseptin numero muodostetaan kaavalla vuosi-apteekin numero-reseptin numero. [Koponen-Piironen 2001]

### **5.1.1 XML Schema**

XML Schemalla määritetään XML dokumentissa esiintyvät elementit ja niiden tyypit. Esimerkiksi XML dokumentissa oleva elementti <potilas> määritetään XML Schema dokumentissa <xs:element name="potilas" type="potilas">. Elementit, jotka sisältävät alielementtejä määritetään tyyppiä complexType ja elementit, jotka sisältävät vain tekstiä (kuten string, date, integer) voidaan määritellä tyyppiä simpleType.

SimpleType -tyypillä määritellyille elementeille voidaan tehdä rajauksia, vaikka osa niistä voidaan toteuttaa myös sovelluksissa. Tällaisia rajoitteita ovat muun muassa sosiaaliturvatunnuksen muodostaminen numeroista ja kirjaimista aina samanpituisiksi tai turhan tyhjän tilan poistaminen tekstistä. Näin voidaan määrittää, että jokaisella potilaalla, jolle kirjoitetaan sähköinen resepti, on oltava sosiaaliturvatunnus ja nimi, joka koostuu etu- ja sukunimestä. Jokaisen reseptin numero ja tunniste voidaan myös määrittää tietyn pituiseksi sarjaksi joko numeroita tai numeroita ja kirjaimia. Näin myös esimerkiksi ATC-koodi tai tuotenumero voidaan määrittää ennalta määrättyjen ohjeiden mukaan.

XML Schema määrittää myös, että jokainen käytössä oleva päivämäärä on samaa muotoa, joten niissä ei aiheudu epäselvyyksiä. Esimerkiksi merkintä 07/04/06 voi tarkoittaa maasta riippuen 7.4.2006, 4.7.2006, 6.4.2007 tai 4.6.2007. XML Schema määrittää päiväyksen aina muodossa vuosi-kuukausi-päivä (YYYY-MM-DD) virhetulkintojen estämiseksi. Tarkoituksena on siis määrittää jokainen XML dokumentissa esiintyvä elementti ja sen mahdollinen attribuutti, jotta niitä voidaan käyttää dokumentissa. Seuraavassa on esimerkki XML Schemasta, joka on tehty edellisen XML:n pohjalta.

```
<xs:schema xmlns:xs="http://www.w3.org/1999/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">

  <xs:element name="resepti" type="resepti">
    <xs:complexType name="resepti">

      <xs:element name="potilas" type="potilas">
        <xs:complexType>
          <xs:element name="sotu" type="xs:string"/>
          <xs:element name="nimi" type="xs:string">
            <xs:simpleTypeType>
              <xs:element name="etunimi"
                type="xs:string">
              <xs:element name="sukunimi"
                type="xs:string">
            </xs:simpleTypeType>
          </xs:element>
        </xs:complexType>
      </xs:element>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

        </xs:simpleType>
        <xs:element name="paino" type="xs:string"/>
        <xs:element name="yhteystiedot"
        type="xs:string"/>
        <xs:element name="kieli" type="xs:string"/>
    </xs:complexType>
</xs:element>

<xs:element name="laake" type="laake"/>
    <xs:complexType>
        <xs:element name="nimike" type="xs:string"/>
        <xs:element name="ATC" type="xs:string"/>
        <xs:element name="VNR" type="xs:string"/>
        <xs:element name="myyntilupa" type="xs:string"/>
        <xs:element name="vahvuus" type="xs:string"/>
        <xs:element name="pakkausko" type="xs:string"/>
        <xs:element name="pakkauslkm" type="xs:string"/>
        <xs:element name="laakemuoto" type="xs:string"/>
        <xs:element name="toimitettava_maara"
        type="xs:string"/>
        <xs:element name="kesto" type="xs:string"/>
        <xs:element name="vaihdonKielto"
        type="xs:string"/>
        <xs:element name="rajoitukset" type="xs:string"/>
        <xs:element name="annostus" type="xs:string"/>
        <xs:simpleType>
            <xs:element name="annostelu"
            type="xs:string"/>
            <xs:element name="kayttoTarkoitus"
            type="xs:string"/>
            <xs:element name="lisaohje"
            type="xs:string"/>
            <xs:element name="sic"
            type="xs:string"/>
            <xs:element name="vrk_annos"
            type="xs:string"/>
        </simpleType>
    </xs:complexType>

```

```

        </element>
        <xs:element name="erillisselvitys"
            type="xs:string"/>
    </xs:complexType>
</xs:element>

<xs:element name="lisatietoja" type="lisatietoja">
    <xs:simpleType>
        <xs:element name="pysyva_tilapainen"
            type="xs:string"/>
        <xs:element name="uusi_hoito" type="xs:string"/>
        <xs:element name="viesti" type="xs:string"/>
        <xs:element name="neRep" type="xs:string"/>
    </xs:simpleType>
</xs:element>

<xs:element name="laakari" type="laakari"/>
    <xs:complexType>
        <xs:element name="laakarinNimi" type="xs:string">
            <xs:simpleType>
                <xs:element name="etunimi"
                    type="xs:string"/>
                <xs:element name="sukunimi"
                    type="xs:string"/>
            </xs:simpleType>
        </xs:element>
        <xs:element name="myontamisPaikka"
            type="xs:string"/>
        <xs:element name="myontamisPvm" type="xs:date"/>
        <xs:element name="svnro" type="xs:string"/>
        <xs:element name="oppiarvo" type="xs:string"/>
        <xs:element name="erikoisala" type="xs:string"/>
        <xs:element name="allekirjoitus"
            type="xs:string"/>
    </xs:complexType>
</xs:element>

```

```

<xs:element name="palkkio" type="palkkio">
  <xs:simpleType>
    <xs:element name="summa" type="xs:string"/>
    <xs:element name="laakaran_toimi"
      type="xs:string"/>
  </xs:simpleType>
</xs:element>

<xs:element name="apteekki" type="apteekki"/>
  <xs:complexType>
    <xs:element name="kommentti" type="xs:string"/>
    <xs:element name="oikaisu" type="xs:string"/>
    <xs:element name="pvm" type="xs:string"/>
    <xs:element name="allekirjoitus"
      type="xs:string">
  </xs:complexType>
</xs:element>

<xs:element name="toimitus" type="toimitus"/>
  <xs:complexType>
    <xs:element name="apteekki_nro"
      type="xs:string"/>
    <xs:element name="potilas" type="xs:string"/>
    <xs:element name="toimitusaika" type="xs:date"/>
    <xs:element name="toimitusmaara"
      type="xs:string"/>
    <xs:element name="status" type="xs:string"/>
    <xs:element name="toimittaja" type="xs:string">
      <xs:simpleType>
        <xs:element name="etunimi"
          type="xs:string"/>
        <xs:element name="sukunimi"
          type="xs:string"/>
      </xs:simpleType>
    </xs:element>
    <xs:element name="reseptinLaji"
      type="xs:string"/>

```

```

        <xs:element name="reseptinro" type="xs:string"/>
        <xs:element name="hinta" type="xs:string"/>
        <xs:element name="korvausLuokka"
        type="xs:string"/>
        <xs:element name="allekirjoitus"
        type="xs:string"/>
    </xs:complexType>
</xs:element>

    <xs:element name="tunniste" type="tunniste"/>
</xs:complexType>
</xs:element>
</xs:schema>

```

### 5.1.2 XPath - polkulausekkeet

Kun potilaalle määrätään kerrallaan useampaa kuin yhtä lääkettä, on aina mahdollista, että ne reagoivat keskenään jollakin ei-toivotulla tavalla. Samalla tavalla on mahdollista, että jokin tietty lääke ei sovi tietylle potilaalle tai että potilaan aikaisempi lääkitys on ollut niin tehokasta, että sitä kannattaa käyttää vaivaan jatkossakin. Tätä varten lääkärien on hyvä tietää mitä lääkkeitä potilaalla tällä hetkellä on tai mitä heillä on aikaisemmin ollut. Niin lääkäreitä kuin tilastollisia dokumenttejakin varten on oleellista, että reseptitiedoista voidaan hakea haluttuja tietoja. Kun reseptit ovat XML muodossa, voidaan niistä hakea tietoa osoittamalla haluttuihin elementteihin muun muassa XPath-kielen avulla.

XML Path Language (XPath) on kieli, jolla voidaan osoittaa XML dokumentin eri osiin. XPath ei itse ole XML-pohjainen, vaan se kehitettiin luomaan yhteinen syntaksi ja semantiikka XPointerin ja XSLT:n (eXtensible Style Language Transformations) välille. Nimensä XPath on saanut sen käyttämistä polkulausekkeista, jotka ovat yleisimpiä XPath-lausekkeita. XPathia voidaan käyttää myös solmujen sovittamiseen tiettyyn malliin ja pienimuotoisena kyselykielenä, sillä XPath muotoilee XML dokumentin solmuista koostuvaksi puurakenteeksi. Solmut voivat olla erilaisia ja niitä on yhteensä seitsemän;

elementti-, attribuutti-, teksti-, nimiavaruus-, käsittelyohje-, kommentti- ja dokumentti (juuri)-solmu. [Clark & DeRose 1999]

Yleisimpiä kielen käyttämiä kieliopillisia rakenteita ovat siis polut, joilla määritetään jokin sijainti. Jokainen polku voidaan esittää yksinkertaisen, mutta laaja-sanaisen syntaksin avulla ja syntaksia voidaan käyttää joko lyhennettynä (abbreviated syntax) yleisten tapausten esittämiseksi ytimekkäästi tai lyhentämättömänä (unabbreviated syntax). Lyhennytyssä syntaksissa yksinkertaisin muoto on /A/B/C, jossa valitaan juurielementin A alielementin B alielementit C. Jos halutaan muodostaa vaikeampia lausekkeita, voidaan määritellä akseli, solmutesti tai predikaatti. Akseli on jokin muu kuin oletusarvoinen lapsiakseli, solmutestissä voidaan testata jotakin muuta kuin nimeä ja predikaatti voi sijaita minkä tahansa askeleen perässä hakasuluissa. Täydessä syntaksissa akselit määritetään tarkasti ja sitä seuraa merkintä :: sekä solmutesti, esimerkiksi /child::A/child::B/child::C. Solmutesteillä voidaan testata solmun nimeä tai löytää tiettyjä solmuja. Esimerkiksi `text()` löytää tekstisolmun ja `node()` löytää minkä tahansa solmun. [Clark & DeRose 1999]

Sähköisessä reseptissä XPathia voidaan käyttää kyselykielenä. Esimerkkinä on alla oleva mukailtu sähköisen reseptin XML dokumentti:

```
<resepti>
  <potilas sotu="120345-6789">
    <nimi>
      <etunimi>Pekka</etunimi>
      <sukunimi>Potilas</sukunimi>
      <paino></paino>
    </nimi>
  </potilas>
</resepti>
```



Tässä <resepti> on dokumentti- eli juurisolmu, <etunimi>Pekka</etunimi> on elementtisolmu ja sotu="120345-6789" on attribuutisolmu. Solmut, joilla ei ole lapsia tai vanhempia ovat atomisia arvoja. Tällaisia ovat siis esimerkissä Pekka ja "120345-6789". Jokaisella elementillä ja attribuutilla on kuitenkin yksi vanhempi ja elementeillä voi olla nolla, yksi tai monta lasta (alielementtiä). Polkulausekkeiden avulla voidaan valita solmuja XML dokumentissa seuraamalla tiettyä polkua tai askelia. Esimerkiksi solmunnimellä voidaan valita kaikki solmun lapset (alielementit), "/" valitsee juurisolmun, "." valitsee kyseessä olevan solmun, ".." valitsee kyseisen solmun vanhemman ja "@" valitsee attribuutteja. Näin esimerkistä saadaan valittua kaikki reseptisolmun lapsisolmut lausekkeella resepti, /resepti valitsee juurielementin resepti, resepti/potilas valitsee kaikki potilaselementit, jotka ovat reseptin alielementtejä, //etunimi valitsee kaikki etunimi-elementit huolimatta siitä missä kohtaa dokumenttia ne ovat ja //@sotu valitsee kaikki attribuutit, joiden nimi on sotu. [XPath 2006]

Jotta voidaan löytää tietty solmu tai solmu, jossa on tietty arvo, käytetään predikaatteja, jotka merkitään aina hakasulkuihin. Sähköisessä reseptissä voidaan siis etsiä tietyn lääkärin kirjoittamia reseptejä, tietyn potilaan saamia reseptejä tai kuinka monta kertaa tiettyä lääkettä on määrätty tietyssä aikana. Esimerkiksi /resepti/potilas[Pekka] etsii kaikki reseptielementin potilaselementit, joiden arvo on Pekka, ja //potilas[@sotu='120345-6789'] etsii kaikki potilaselementit, joilla on '120345-6789' arvon omaava sotu-attribuutti. Tuntemattomia solmuja voidaan valita "\*" -merkin avulla tai node()-lausekkeella. Esimerkiksi /resepti/\* valitsee kaikki reseptielementin alielementit ja //potilas[@\*] valitsee kaikki potilaselementit, joilla on jokin attribuutti. Useampia polkuja voidaan valita yhtä aikaa käyttämällä erottimena | -operaattoria. Samalla periaatteella voidaan näin etsiä esimerkiksi lääkkeitä, joista on myönnetty ylempi erityiskorvaus tai joiden hinta on yli 50 euroa tai yhdessä apteekissa toimitetut lääkkeet tietyllä aikavälillä Kansaneläkelaitoksen suorakorvauksen maksamista varten. Kyselyjä voidaan siis käyttää myös muihin tietojen hakemisiin kuin vain lääkäreiden reseptien kirjoittamisen esitiedoksi ja niistä voivat hyötyä sekä potilaat että

terveydenhoitohenkilökunta. On kuitenkin muistettava ottaa huomioon, että yksityisyyden suoja estää yksityishenkilöiden tietojen keräämisen ilman lupaa, ja tilastoitakin varten yksityisyyden suoja tulee ottaa huomioon lain vaatimalla tavalla. [XPath 2006]

## **5.2 XML-pohjaisten kielten käyttö sähköisen reseptin hallinnassa**

Jotta reseptejä ja niitä sisältäviä tietoja voidaan käyttää paremmin, on otettava käyttöön erilaisia kuvauskieliä. Näistä RDF (Resource Description Framework) ja RDFS (RDF Schema) ovat XML:n kanssa yhteensopivia ja erittäin helppoja käyttää, sillä ne käyttävät samaa syntaksia XML:n kanssa. Mikäli tietojen ja niiden suhteiden kuvaamisessa halutaan mennä vielä syvemmälle, on syytä ottaa käyttöön ontologiakieli, kuten OWL (Web Ontology Language). Reseptien hallinta on helpompaa, kun sen tietosisältö on mallinnettu kuvauskielellä ja eri osapuolten väliset suhteet ovat selvillä. Näin myös mahdollistuvat uudet ja erilaiset kyselyt resepteihin ja lääkkeisiin sekä näitä määränneisiin lääkäreihin ja toimittaneisiin apteekkeihin.

### **5.2.1 RDF – kuvauskielen hyödyntäminen**

RDF (Resource Description Framework) on kehys, jonka avulla kuvataan webin resursseja, kuten otsikkoja, toimittajia, päiväyksiä ja sisältöä eli RDF on perusta metadatan luomiselle, tallentamiselle ja käsittelemiselle. RDF:n avulla voidaan luoda tietomalleja, joita voidaan vaihtaa yksittäisten osapuolien välillä RDF:n syntaksin turvin. RDF kirjoitetaan XML:llä ja se on tarkoitettu (myös) koneiden ymmärrettäväksi. RDF:n avulla voidaan muun muassa kuvata sisältöä hakukoneita varten tai kuvata ominaisuuksia kuten hinta tai pakettikoko. Asioiden tunnistamiseen käytetään verkkotunnistimia (URI) ja resursseja kuvataan ominaisuuksien ja niiden arvojen avulla. Resurssi voi olla mitä tahansa, millä on URI (<http://www.apteekki.net>) ja ominaisuus on resurssi, jolla on nimi (kotisivu, apteekkari). Ominaisuuden arvo on sen nimi (Pirjo Proviisori). Jokainen näiden kolmen yhdistelmä on lause, eli objekti-ominaisuus-arvo –kolmikko. Esimerkiksi

```

<?xml version="1.0"?>
<rdf:RDF>
  <rdf:Description rdf:about="http://www.apteekki.net">
    <apteekkari>Pirjo Proviisori</apteekkari>
    <kotisivu>www.apteekki.net</kotisivu>
    <ohjelmisto>Pagemaker 1.0</ohjelmisto>
  </Description>
</rdf:RDF>

```

Esimerkissä määritetään ensin käytettäväksi XML syntaksia <?xml version="1.0"?> ja juurielementti on <rdf:RDF>. Juurielementissä mainitaan tavallisesti myös käytettävät nimiavaruudet, mutta esimerkissä ne on jätetty pois. Seuraavana on kuvauslementti `rdf:Description`, joka sisältää attribuutin `rdf:about`. Attribuutti määrittelee kohteen, jota kuvataan, eli esimerkissä URL:n sivustolle `apteekki.net`. Kohteesta kuvataan apteekkari Pirjo Proviisori ja kotisivu `www.apteekki.net`, joka on tehty käyttäen ohjelmistoa Pagemaker 1.0. Samalla tavalla voidaan kuvata sähköistä reseptiä, sillä RDF:n avulla voidaan lisätä kuvauksia kaikkiin URI:n avulla osoitettaviin resursseihin, jopa kuviin ja reseptitiedostoihin. Alla oleva esimerkki kuvaa sähköistä reseptiä.

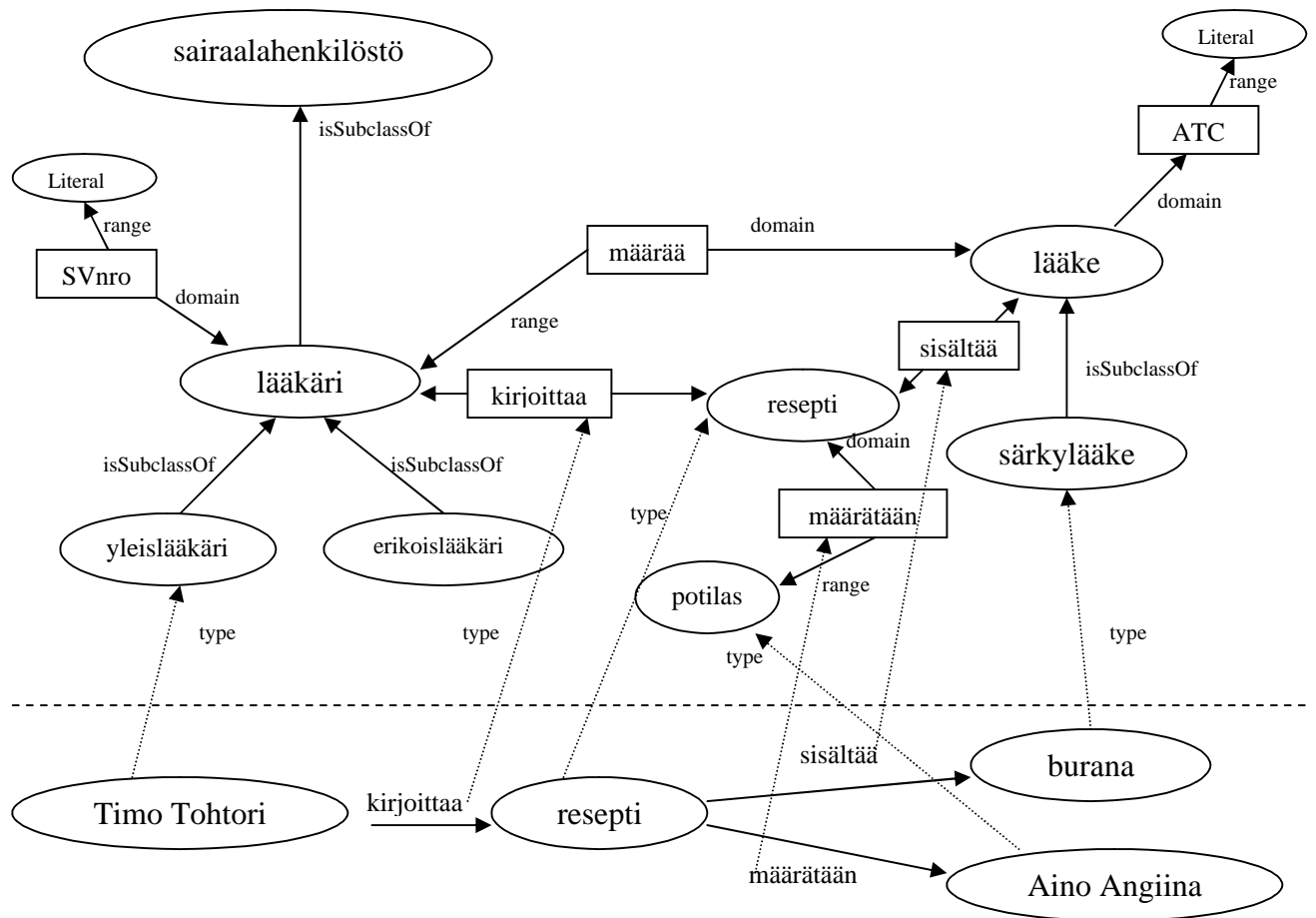
```

<?xml version="1.0"?>
<rdf:RDF>
  <rdf:Description
    rdf:about="http://www.reseptitietokanta.fi/#resepti/200612341234">
    <tyyppi>lääkemääräys</tyyppi>
    <kieli>suomi</kieli>
    <potilas>Pirkko Potilas</potilas>
  </Description>
</rdf:RDF>

```

### 5.2.2 RDFS - kaaviotekniikka

RDF Schemalla jokainen käyttäjä voi siis itse määrittää käytettävissä olevan terminologian ja semantiikan. Sähköisen reseptin tapauksessa RDF Schemalla voidaan määrittää muun muassa eri toimijoiden välisiä suhteita tai asettaa tiettyjä rajoitteita. Otetaan siis esimerkiksi lause ”Timo Tohtori määrää Aino Angiinalle buranaa”. Alla olevassa kuvassa havainnollistetaan RDF:ää ja RDS Schemaa. Katkoviivan alapuolella olevat soikiot ovat instansseja ja yläpuolella olevat ovat luokkia sekä neliöt ominaisuuksia.



Kuva 14 Kaavio RDF:n ja RDFS:n kerroksista ja suhteista toisiinsa [Antoniou & Harmenlen 2004 s. 83-90]

Tästä kaaviosta voidaan rakentaa RDF Schema dokumentti, josta esimerkki seuraavassa:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

<rdfs:Class rdf:ID="erikoislaakari">
  <rdfs:comment> erikoislääkäri-luokka, joka kuuluu lääkäriinluokkaan
  </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#laakari"/>
</rdfs:Class>

<rdfs:Class rdf:ID="yleislaakari">
  <rdfs:comment> yleislääkäri-luokka, joka kuuluu lääkäriinluokkaan
  </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#laakari"/>
</rdfs:Class>

<rdfs:Class rdf:ID="laakari">
  <rdfs:comment> lääkäri-luokka, joka kuuluu luokkaan
  sairaalahenkilöstö
  </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#sairalahenkilosto"/>
</rdfs:Class>

<rdfs:Class rdf:ID="sairalahenkilosto"/>

<rdfs:Class rdf:ID="sarkylaake">
  <rdfs:comment> luokka särkylääke, joka kuuluu luokkaan laake
  </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#laake"/>
</rdfs:Class>

<rdfs:Class rdf:ID="laake"/>
<rdfs:Class rdf:ID="potilas"/>
<rdfs:Class rdf:ID="resepti"/>
```

```

<rdf:Property rdf:ID="kirjoittaa">
  <rdfs:comment> kirjoittaa-ominaisuus
</rdfs:comment>
  <rdfs:domain rdf:resource="#resepti"/>
  <rdfs:range rdf:resource="#laakari"/>
</rdfs:Property>

<rdf:Property rdf:ID="määrää">
  <rdfs:domain rdf:resource="#laake"/>
  <rdfs:range rdf:resource="#laakari"/>
</rdfs:Property>

<rdf:Property rdf:ID="sisaltaa">
  <rdfs:domain rdf:resource="#laake"/>
  <rdfs:range rdf:resource="#resepti"/>
</rdfs:Property>

<rdf:Property rdf:ID="maarataan">
  <rdfs:domain rdf:resource="#potilas"/>
  <rdfs:range rdf:resource="#resepti"/>
</rdfs:Property>

<rdf:Property rdf:ID="SVnro">
  <rdfs:comment> SVnro on lääkäri-luokan ominaisuus, jolla on
kirjaimellisia arvoja
</rdfs:comment>
  <rdfs:domain rdf:resource="#laakari"/>
  <rdfs:range rdf:resource="#&rdf;Literal"/>
</rdfs:Property>

<rdf:Property rdf:ID="ATC">
  <rdfs:comment> ATC on lääke-luokan ominaisuus, jolla on
kirjaimellisia arvoja
</rdfs:comment>
  <rdfs:domain rdf:resource="#laake"/>
  <rdfs:range rdf:resource="#&rdf;Literal"/>
</rdfs:Property>

```

Kaavion ja RDF Schema dokumentin mukaan Timo Tohtori on erikoislääkäri, joka kuuluu lääkäriluokkaan. Lääkäriluokasta sekä erikoislääkäri- että yleislääkäriluokat perivät ominaisuuden SVnro eli sairausvakuutusnumero, joka tulee olla kaikilla lääkäreillä. Lääkärit kuuluvat myös sairaalahenkilökuntaan. Lääkäriluokalla on myös ominaisuudet määrää ja käsittää. Määrää on käsittää ominaisuuden aliominaisuus, eli lääkäri voi määrätä potilaalle lääkettä, jolloin tapahtumassa tarvitaan aina sekä lääkäri että lääke. Burana on särkylääkeluokan tyyppi ja särkylääke kuuluu luokkaan lääke, jolla on ominaisuus ATC. Kuten SVnro-ominaisuuskin, ATC-ominaisuus voi saada ainoastaan kirjaimellisia (literal) arvoja.

### 5.2.3 OWL – ontologiakielen hyödyntäminen

Web Ontology Language on periaatteessa laajennus RDF Schemalle. Se on siis ontologiakieli, jonka avulla voidaan tarkasti määrittää hierarkioita ja ominaisuuksien periytymistä luokasta toiseen. OWL on rakennettu RDF:n päälle ja se kirjoitetaan XML:llä niin, että se on koneiden ymmärtämässä muodossa, jotta tieto on käsiteltävissä eikä ainoastaan näkyvissä. OWL eroaa RDF:stä ja RDF Schemasta siinä, että sen ilmaisuvoima on huomattavasti vahvempi, jolloin koneet tulkitsevat kielen paremmin. OWL:ssa on lisäksi suurempi sanasto ja vahvempi syntaksi, jotka edesauttavat toimintaa koneiden kanssa. [RDF 2006]

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.owl-ontologies.com/unnamed.owl#"
  xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Ontology rdf:about="" />
  <owl:Class rdf:ID="yleislaakari">
    <rdfs:subClassOf>
```

```

        <owl:Class rdf:ID="Laakari" />
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="erikoislaakari">
    <rdfs:subClassOf>
        <owl:Class rdf:about="#Laakari" />
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Sarkylaake">
    <rdfs:subClassOf>
        <owl:Class rdf:ID="Laake" />
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Resepti" />
<owl:Class rdf:ID="Potilas" />
<owl:Class rdf:about="#Laakari">
    <rdfs:subClassOf>
        <owl:Class rdf:ID="Sairaalahenkilosto" />
    </rdfs:subClassOf>
</owl:Class>
<owl:ObjectProperty rdf:ID="kirjoittaa">
    <rdfs:domain rdf:resource="#Laakari" />
    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty" />
    </owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="maarataan">
    <rdfs:domain rdf:resource="#Resepti" />
    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty" />
    </owl:ObjectProperty>
<owl:FunctionalProperty rdf:ID="SVnro">
    <rdfs:domain rdf:resource="#Laakari" />
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty" />
    </owl:FunctionalProperty>

```



```

<owl:FunctionalProperty rdf:ID="ATC">
  <rdfs:domain rdf:resource="#Laake"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="sisaltaa">
  <rdfs:domain rdf:resource="#Resepti"/>
  <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="maaraa">
  <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="#Laakari"/>
</owl:FunctionalProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 2.2, Build 339)
http://protege.stanford.edu -->

```

### 5.2.4 RQL - kyselykieli

Jotta tietoa, jota on talletettu RDF- ja RDF Schema –muodossa, voidaan hyödyntää, siihen täytyy voida tehdä kyselyjä. Näin voitaisiin selvittää esimerkiksi kaikki lääkärit, jotka ovat määränneen särkylääkkeenä buranaa, tai potilaat, jotka ovat saaneet tiettyä lääkettä. RDF:lle on suunniteltu useita kyselykieliä, joista tässä käsitellään RQL (RDF Query Language) ja sitä käytetään nimenomaan RDF ja RDF Schema dokumentteihin tarkoitettuihin kyselyihin. Aiemmin käsitelty XPath on myös erittäin käytännöllinen kyselykieli, mutta se ei suoraan sovellu RDF/RDFS-kyselyihin, vaan paremmin XML-pohjaisiin kyselyihin. Tämä johtuu siitä, että XML:ssä yhtä asiaa voidaan kuvata monella eri tavalla, joten jokaiselle niistä on oltava erilainen XPath-lause. RQL:n avulla pyritään kyselyjen vastaukset löytämään RDF-tasolla, jolloin yksi kyselylause palauttaa halutun tiedon sen kuvaustavasta huolimatta. Lisäksi XML rakentuu puumaisesti kun taas RDF ja

RDF Schema muodostavat verkkomaisen rakenteen ja XML-kyselyiden on tarkoitus kohdistua juuri rakenteeseen, mutta RDF/S-kyselyillä voidaan selvittää myös rakenteiden välisiä suhteita. Tämänkin takia oma kyselykieli on välttämätön. [Antoniou & Harmenlen 2004 s. 100-102]

RQL lähtee liikkeelle aivan yksinkertaisista kyselyistä, joista tarkentamalla päästään monimutkaisempiin kyselyihin. Peruskyselyjä ovat `Class` ja `Property`, jotka palauttavat kaikki dokumentin luokat ja ominaisuudet. Haluttaessa selvittää kaikki tietyn luokan instanssin kirjoitetaan yksinkertaisesti tämän luokan nimi, esimerkiksi `resepti` palauttaa kaikki resepti-luokan instanssit ja samoin sen aliluokkien instanssit. Kun halutaan jättää pois aliluokkien instanssit, käytetään luokan nimen edessä `^`-merkkiä. Näin ollen `^resepti` palauttaa kaikki resepti-luokan instanssit, muttei perittyjä instansseja. Samalla tavalla voidaan hakea ominaisuuksia kirjoittamalla ominaisuuden nimi. Lisäksi joitakin SQL:stä (Structured Query Language) tuttuja ominaisuuksia voidaan käyttää myös RQL:ssä. Esimerkiksi `select` määrittää haetun tiedon järjestyksen, `from` navigoi tietomallin läpi ja `where` määrittää palautettavalle tiedolle ehtoja. Esimerkiksi haluttaessa tietää kaikki lääkärit, jotka ovat määränneet lääkkeitä, kirjoitetaan

```
select X, Y
from {X}maaraa{Y}
```

missä `X` ja `Y` ovat muuttujia, joihin vastaukset sidotaan ja resurssi-ominaisuus-arvokolmikkoa merkitään `from {X}maaraa{Y}`. Samalla periaatteella voidaan etsiä kaikki erikoislääkärit, jotka ovat määränneet lääkkeitä. Tällöin merkitään

```
select X, Y
from erikoislaakari{X}.maaraa{Y}
```

missä pisteellä merkitään implisiittistä liitosta. Eksplisiittisen liitoksen avulla voidaan kysellä esimerkiksi reseptejä, jotka on kirjoittanut lääkäri, jonka SV-numero on 123456

```
select N
from laakari{X}.maaraa{Y}, {C}sukunimi{N}
where Y="123456" and X=C
```

tai vastaavasti voidaan etsiä kaikki reseptit, joita on muokattu esimerkiksi tänä vuonna (tai muun annetun päivämäärän jälkeen)

```
select X, Y
from resepti{X}.last_modified{Y}
where Y >= 2006-01-01
```

Kun halutaan tarkentaa kysely RDF Schemalle tai määrittää, että muuttujat sijoittuvat scheman luokkiin, käytetään luokan edessä \$-merkkiä ja ominaisuuden edessä @ -merkkiä.

Esimerkiksi

```
select X, $X, Y, @Y
from {X:$X}maaraa{Y:@Y}
```

palauttaa kaikki resurssit ja arvot niiltä kolmikoilta, joiden ominaisuus on maaraa sekä kaikki sen aliominaisuudet ja niiden luokat.

Tarkempia kyselyjä voidaan tehdä lisäämällä ehtoja. Näin esimerkiksi voidaan etsiä kaikki reseptit, jotka ovat erikoislääkäreiden kirjoittamia ja sisältävät jotakin särkylääkettä, kirjoittamalla

```
select Z, W
from erikoislaakari.maaraa{Y:@Y}.sarkylaake{Z:$Z}, {V:$V}resepti{W:@W}
where Y=V
```

[Antoniou & Harmenlen 2004 s. 100-104; FORTH 2003]

## 6 XML-TEKNIKOIDEN TOTEUTUS

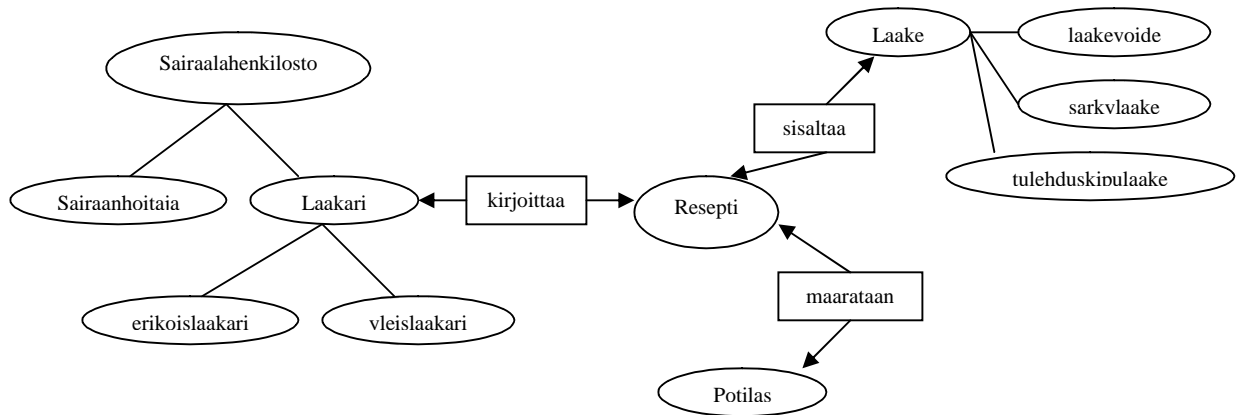
XML ja RDF dokumenttien sekä niihin tehtävien kyselyiden kokeilemiseksi työssä tutkittiin vapaan lähdekoodin omaavaa ohjelmaa Protégé 3.2 betaa. Protégé on eräänlainen ontologia-editori ja tietopohjakehys, jolla voidaan luoda domain-malleja. Se pohjautuu javaan ja sen avulla ontologioita voidaan käsitellä muun muassa RDF(S)- OWL- ja XMS Schema-muodoissa. Protégé on Stanfordin yliopiston Lääketieteen laitoksen informatiikan osaston kehittämä ohjelma, jota tukevat erilaiset akateemiset ja hallinnolliset organisaatiot sekä yritykset. Sen java-pohjaisuus edesauttaa myös ontologioiden käyttöä sovelluskehityksessä ja prototyyppien luonnissa. Protégén lisäksi kappaleen lopussa esitellään myös muunlaisia ontologioita tai RDF:ää hyödyntäviä sovelluksia. [Protégé 2006]

### 6.1 Protégé - ontologiaeditori

Koekäytössä asennettiin HP:n Compaq nx7010 kannettavalle tietokoneelle Protégén 3.2 beta-versio, joka on ohjelman uusin versio. Ohjelmaan kasattiin kuvan 12 tapainen esimerkkiprojekti, johon kuitenkin lisätään luokkia ja arvoja kyselyiden kokeilemista varten. Luokkia on yhteensä 11; pääluokkia ovat lääke, potilas, resepti ja sairaalahenkilöstö. Lääkkeellä on alaluokat särkylääke, lääkevoide ja tulehduskipulääke sekä sairaalahenkilöstöllä alaluokat sairaanhoitaja ja lääkäri, jolla edelleen on alaluokat erikoislääkäri ja yleislääkäri. Luokan lääke attribuutteja ovat nimike, ATC-luokkatunnus, vahvuus ja lääkeuoto, ja nämä periytyvät alaspäin myös alaluokille. Luokalla lääkäri on attribuutteina SVnro eli sairausvakuutusnumero ja erikoisala ja se perii attribuutit etunimi ja sukunimi ylemmältä sairaalahenkilöstöluokalta. Nämä puolestaan periytyvät alaspäin erikois- ja yleislääkäriluokille. Reseptiltä halutaan tietää reseptinumero, kieli, päiväys ja tyyppi sekä potilaalta etunimi, sukunimi ja henkilötunnus. Kokeiluja varten ääkköset jätetään pois luokkien, ominaisuuksien ja arvojen nimistä lähinnä ohjelman englanninkielisyyden takia. Lisäksi on otettava huomioon, että vaikka luokkia tehdessä on

pyrityt ottamaan huomioon kaikki sähköisen reseptin piirteet, niin koe on kuitenkin hyvin rajattu ja sen takia myös luokkien ominaisuuksista on toteutettu vain oleelliset.

Luokkien lisäksi määritetään näille suhteet, jotka ovat seuraavanlaisia: lääkäri *kirjoittaa* reseptin, resepti *määrätään* potilaalle ja resepti *sisältää* lääkkeen. Ominaisuuksille annettiin koemielessä erilaisia keksittyjä arvoja, jotta toimintaa voidaan havainnollistaa tekemällä koekyselyjä Protégén omalla kyselyllä. Alla olevassa kuvassa 13 esitetään eri luokat ja niiden ominaisuudet sekä suhteet ja taulukossa 2 määritellään annetut arvot. Kuvassa 14 esitetään Protégén käyttöliittymää ja ohjelmaan jo talletettuja luokkia.

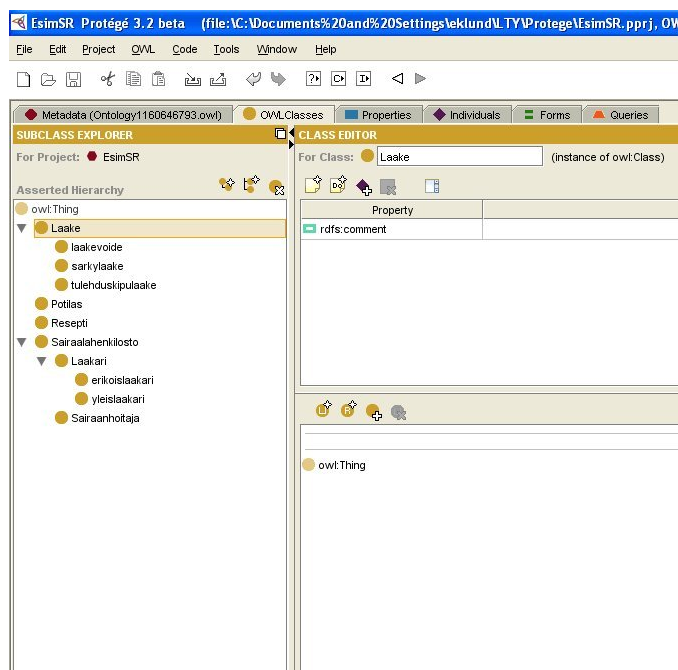


Kuva 15 Esimerkkireseptijärjestelmän luokat ja niiden suhteet

Luokka	Attribuutit	Instanssit
Sairaalahenkilöstö	etunimi, sukunimi, työpaikka	
Sairaanhoitaja	(periytyy ylemmältä luokalta)	Maija, Virtanen, HYS
Lääkäri	SVnro, erikoisala (+ periytyy ylemmältä luokalta)	
Yleislääkäri	(periytyy ylemmältä luokalta)	Pirjo, Virta, KAS, SV123455, yleislääkäri Mika, Nieminen, EKKS, SV123454, yleislääkäri
Erikoislääkäri	(periytyy ylemmältä luokalta)	Anne, Koski, KAS, SV123455, kirurgia Kai, Jokinen, EKKS, SV123453, pediatria
Resepti	reseptinro, tyyppi, kieli, pvm	060101, A, FIN, 2006-01-01 060102, A, FIN, 2004-03-02 060201, A, FIN, 1999-08-05 060202, U, FIN, 2005-06-02 060203, A, SWE, 2002-08-07 060301, U, FIN, 2001-06-17 060401, A, SWE, 2003-09-09

Lääke	nimike, ATC, vahvuus, lääkemuoto	
Lääkevoide	(periytyy ylemmältä luokalta)	Advantan, DO7AC14, 0,1%, voide Elidel, D11AX15, 10mg, emulsiovoide
Särkylääke	(periytyy ylemmältä luokalta)	Aspirin, NO2BA01, 500mg, tabletti Panadol, NO2BE01, 500mg, tabletti
Tulehduskipulääke	(periytyy ylemmältä luokalta)	Burana, MO1AE01, 800mg, tabletti Mobic, MO1AC06, 15mg, tabletti Lodine, MO1AB08, 400mg, depotabletti

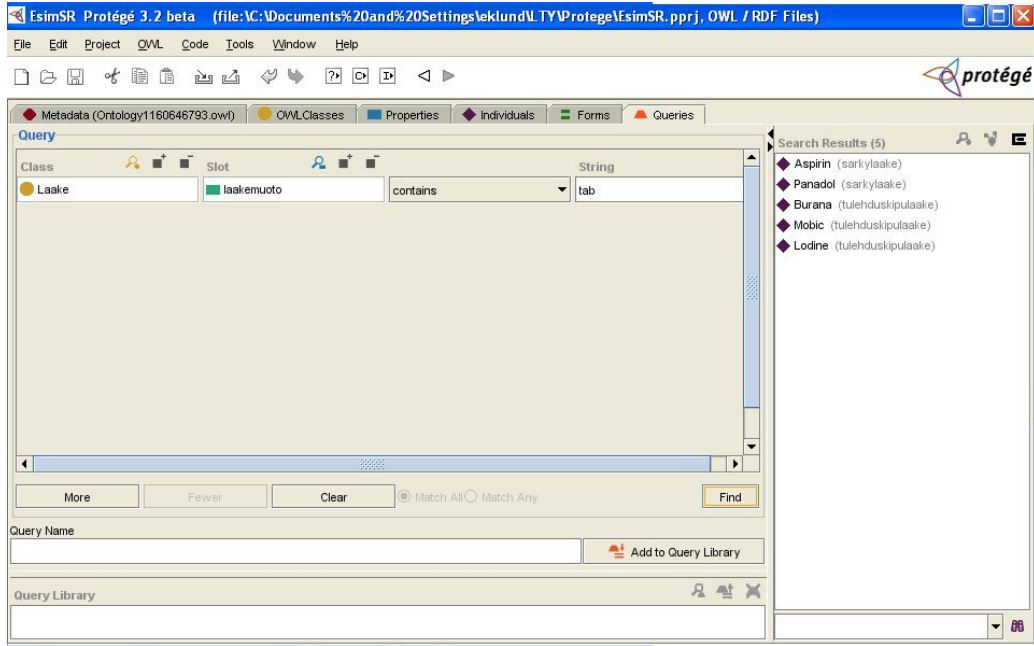
Taulukko 2 Esimerkkireseptijärjestelmän luokat instansseineen



Kuva 16 Protégén käyttöliittymä ja esimerkkijärjestelmään talletetut luokat

Protégélla on kaksi erilaista mahdollisuutta tehdä kyselyjä. Ensimmäinen on kuvassa 15 näkyvällä Queries-välilehdellä oleva kysely ja toinen on luokkien väliset yhteydet graafisesti esittävä Jambalaya-ominaisuus. Kysely-välilehdellä voidaan suorittaa yksinkertaisia kysely talletettuihin luokkiin ja niiden ominaisuuksiin sekä arvoihin ja kyselyt voidaan myös tallettaa myöhempää käyttöä varten. Kuva 15 havainnollistaa ensimmäistä koekyselyä ja Protégén kyselyominaisuuksia. Ensimmäisessä kyselyssä haetaan kaikkia niitä lääkkeitä, joiden lääkemuoto sisältää merkkijonon ”tab”, eli toisin sanoen kaikkia niitä lääkkeitä, jotka myydään tabletteina. Kysely osaa määrittää luokan

särkylääke luokan lääke alaluokaksi ja etsii vastauksia myös sieltä. Kuvassa kysely on vasemmassa reunassa ja vastaus näkyy oikeassa reunassa.



Kuva 17 Protégén kyselyt-välilehti ja esimerkkikysely

Esimerkkitaulukoihin esitettiin myös neljä muuta kyselyä, jotka näkyvät alla olevassa taulukossa.

Kysymys	Kysely (luokka-ominaisuus-arvo)	Vastaus
1. Kuka on kirjoittanut reseptin 060401?	Laakari – kirjoittaa – 060401	Jokinen (erikoislaakari)
2. Mitkä lääkkeet ovat vahvuudeltaan alle 200mg?	Laake – vahvuus – (is less than) 200	Advantan (laakevoide) Elidel (laakevoide) Mobic (tulehduskipulääke)
3. Ketkä sairaalahenkilökunnasta työskentelevät KAS:lla?	sairaalahenkilökunta – työpaikka – KAS	Koski (erikoislaakari) Virta (yleislaakari)
4. Kuka on kirjoittanut reseptin Antti Haapalalle?	Resepti – maarataan - Haapala Laakari – kirjoittaa – 060101	060101 (Resepti) Koski (erikoislaakari)

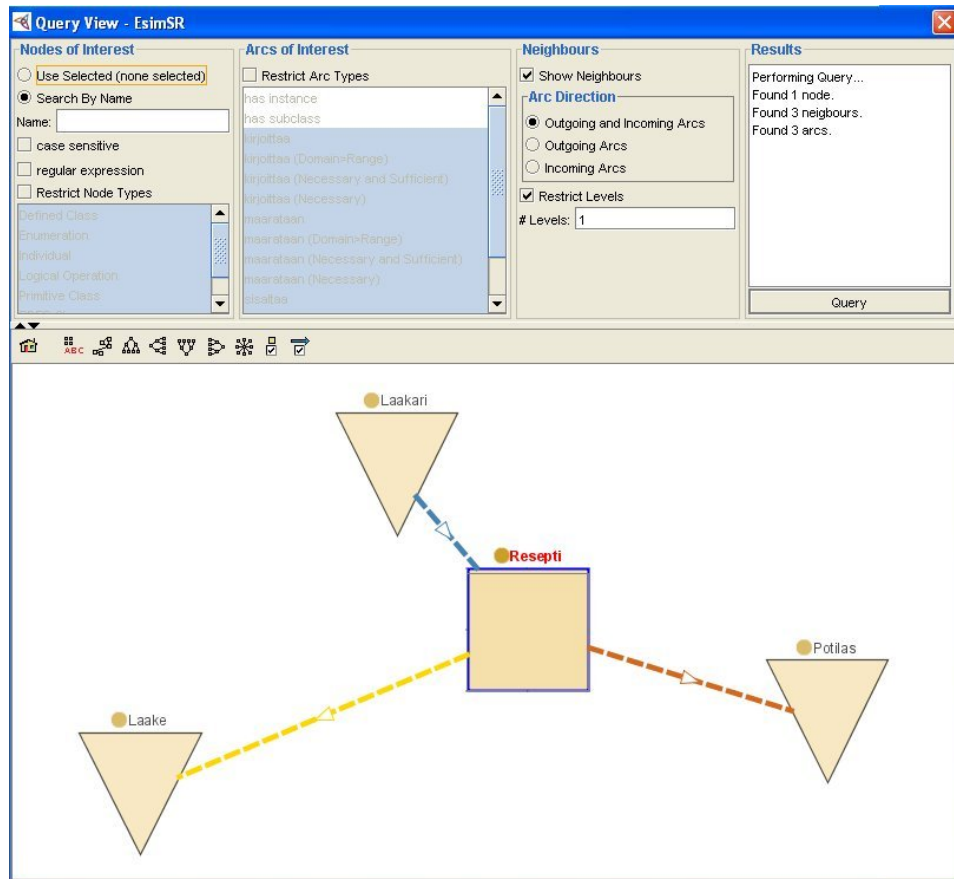
Taulukko 3 Protégelle esitetyjä kysymyksiä ja vastauksia

Protégén kyselyssä voidaan tehdä päällekkäisiä kyselyitä, mutta suhteisiin kohdistuvat kyselyt eivät valitettavasti onnistu. Näin ollen siis voidaan selvittää yhdellä kyselyllä vastaus taulukon ensimmäisiin kysymyksiin ja kahdella päällekkäisellä kyselyllä saadaan vastaus neljänteen kysymykseen, mutta esimerkiksi kysymykseen ”Mitä lääkettä Antti Haapalalle on määrätty?” ei voida helposti selvittää. Tämä onnistuu kokeilemalla mitä lääkettä resepti 060101 sisältää, mutta käytännössä ja varsinkin kaupallisessa käytössä tämä ominaisuus on mahdotonta, sillä ensinnäkin kannan koko on tällöin huomattavasti suurempi. Hyvin pienen esimerkkikannan takia kyselyt ovat tässä testauksessa suppeita ja niihin saadaan vastaukseksi vain instanssien kohdalla ennalta määritetty arvo. Niistä kuitenkin selviää Protégén kyselylehden mahdollisuudet. Pienet kyselyt ovat esimerkkikannalle helppoja ja nopeita selvittää. Ohjelman kyselyä on myös hyvin helppo käyttää, sillä se mukautuu luokkien, suhteiden ja instanssien mukana määritettyihin ominaisuuksiin ja antaa kyselylle näiden mukaan valmiita vaihtoehtoja (kuten contains ja is greater than). Kyselyissä ilmenee virheitä tai nollatuloksia mikäli niitä on myös luokissa tai niiden suhteissa, joten näiden rakentamisessa on oltava huolellinen. Myös luokkien ja suhteiden muuttaminen kyselyiden välillä saattaa vaikuttaa kyselyihin. Kun luokkia ja suhteita muutetaan, instanssit eivät muutu niiden mukana, vaan ne on korjattava erikseen, ja tämä saattaa aiheuttaa ongelmia kyselyissä.

Graafinen kyselymahdollisuus on hieman erilainen Queries-välilehteen verrattuna. Tätä varten avataan ensin kyselyikkuna, joka saadaan esille etsimällä halutun luokan naapurit. Kuvassa 16 näkyy kyselyikkuna, ja Resepti-luokan naapurit. Nuolet osoittavat ominaisuuksien mukaan toisiin luokkiin, eli tässä Resepti määrätään Potilaille, Resepti sisältää Lääkkeen ja Lääkäri määrää Reseptin. Itse kyselyt voidaan suorittaa kirjoittamalla halutun solmun nimi. Tällä graafisella menetelmällä saadaan selville myös vastaus kysymykseen ”mitä lääkettä Antti Haapalalle on määrätty?” kirjoittamalla kyselyyn yksinkertaisesti ”Haapala”. Vastauksena saadaan kuva, jossa näkyy Potilas Haapala, reseptin määrännyt Lääkäri Koski ja Reseptin sisältämä Lääke Advantan. Samalla periaatteella voidaan nyt helposti selvittää kuinka monta reseptiä kullekin potilaalle on määrätty tai kuinka monta reseptiä kukin lääkäri on kirjoittanut ja mitä lääkettä on määrätty



eniten. Protégélla on siis mahdollisuus tehdä vaativampiakin kyselyjä, mutta vain graafisessa muodossa. Tämä on kuitenkin hyvä asia, sillä kuvat ovat erittäin selkeitä ja hyvin havainnollistavia.



Kuva 18 Protégén kyselyikkuna graafisia kyselyihin

## 6.2 Muita sovelluksia

Jena on semanttisen webin ohjelmien rakentamiseen tarkoitettu työkalu, jonka kehitys aloitettiin HP Labsin semanttisen verkon kehitykseen tähtäävässä ohjelmassa (HP Labs Semantic Web Programme). Jenalla on java-pohjainen kehys, joka sisältää muun muassa RDF ja OWL ohjelmointirajapinnan sekä SPARQL kysely-ympäristön. Jena lähtee

liikkeelle ajatuksesta, että on ensin tunnettava RDF-tietomalli, jotta vältetään turhautumiselta ja pettymyksiltä. RDF:n tuntemus onkin siis oleellinen osa Jenan käyttöä, sillä se ei ole yksittäinen ohjelma niin kuin Protégé vaan eräänlainen kirjasto, joka ladataan koneelle, ja jota voidaan hyödyntää Java-ohjelmoinnissa semanttisen verkon palveluja rakennettaessa. Hyvä puoli Jenassa on, että sillä on Protégén tapaan vapaa lähdekoodi. Huonompi puoli on kuitenkin, että sen käyttö vaatii sekä RDF:n että Javan tuntemusta. [Jena 2006]

Kaupallisista sovelluksista mainittakoon Espoolaisen Profium-nimisen yrityksen tuottama SIR (Semantic Information Router), joka on vastikään liitetty osaksi yrityksen samaisen yrityksen Metadata Server-tuotetta. Metadata Serverillä voidaan muuttaa haluttu tieto, esimerkiksi verkkosivuston tiedot, yhtenäiseen muotoon ja koota siitä kuvaavaa metadataa. Tämä tieto talletetaan sitten keskitettyyn tietokantaan, johon voidaan tehdä erilaisia kyselyjä. Metadata Serverin idea on siinä, että tietoa voidaan kerätä mistä tahansa sisällöstä, kuten kuvista, tekstistä, grafiikasta, videoista, äänitallenteista ja datatiedostoista. [Profium 2006]

## 7 YHTEENVETO

Sähköinen resepti on varmasti tulevaisuudessa erittäin käytännöllinen väline, joka tuo säästöä niin lääkäreiden aikaan ja apteekkien toimintaan kuin hallinnollekin. Vaikka sähköisiin resepteihin siirtyminen vaatii vielä aikaa, kehitystä, sekä taloudellista panostusta sitä käyttäviltä, se on epäilemättä hyödyllinen sekä lääkäreille että potilaille. Potilaiden lääketurvallisuus kasvaa kun lääkärit voivat tarkastaa aiemman lääkityksen, oli se sitten määrätty missä tahansa terveydenhuollon yksikössä, ja selata lääketietokannasta juuri potilaalle sopivaa lääkitystä reseptin kirjoituksen yhteydessä. Lääkärit ja terveydenhuollon toimipisteet säästävät samalla aikaa, kun entisiä tietoja ei tarvitse erikseen tilata muualta eikä lääkeoppaita tarvitse selata käsin. Myös apteekit hyötyvät järjestelmästä, sillä heidän ei tarvitse lukea lääkäreiden toisinaan hyvin epäselvää käsialaa eikä kirjata reseptejä uudelleen omaan järjestelmäänsä. Lisäksi Kansaneläkelaitoksen mukana olo mahdollistaa suorakorvausmenetelmän säilyvyyden kaikkien osapuolien eduksi.

Niin hienolta kuin sähköinen resepti kuulostaakin, ei se vielä ole aivan sitä mitä se voisi olla. Koska eri terveydenhuolto-organisaatioilla ja apteekeilla on yhä hyvinkin erilaiset tietojärjestelmät, on varmistuttava, että tiedot siirtyvät näiden välillä moitteettomasti, varmasti ja luotettavasti. Palvelukeskeinen arkkitehtuuri ja eri Web-palvelut tuovat tähän mahdollisuuden, kun niitä käytetään yhdessä jo muodostettujen standardien, kuten HL7:n CDA-määrittelyjen kanssa. Lisäksi hyvin yleistynyt XML-kieli tuo eri järjestelmien välille yhteensopivuutta. Verkon välityksellä liikkuvat reseptitiedot ovat henkilökohtaisia ja arkaluontoisia, mikä puolestaan tuo esille tietoturvakysymyksiä. Lääkärien tunnistamisella voidaan taata reseptin kirjoittajan oikeellisuus ja sähköisellä allekirjoituksella lähettäjä sekä tietojen muuttumattomuus. Myös tässä voidaan hyödyntää XML:ää (XML allekirjoitus).

Vaikka tietojen eheys ja luotettavuus siirtotiellä voitaisiinkin taata, myös fyysiset katkokset verkossa ovat mahdollisia. Tällöin ratkaisuna voisi esimerkiksi olla, että sähköisen siirron lisäksi reseptistä tulostettaisiin nykyistä potilasohjetta vastaava reseptilomake, jossa resepti-

informaatiota liitettäisiin viivakoodiin. Mikäli yhteydet reseptipalvelimelle olisivat poikki, lääkäri voisi silti kirjoittaa ja tulostaa reseptin potilaalle ja apteekissa lääke voitaisiin toimittaa lukemalla lomakkeella olevat viivakoodit. Mikäli myös lääkärin ja/tai apteekkien järjestelmä olisi poissa käytöstä, toimisi lomake reseptinä, jota vastaan lääke voitaisiin toimittaa potilaalle. Toisena vaihtoehtona voisi olla saksassa käytössä oleva älykorttipohjainen reseptijärjestelmä, missä potilaan tiedot ja reseptit talletettaisiin potilaan omalle sirukortille. Reseptipalvelimen käyttö olisi tällöin vain varatoimenpide, sillä lääkäri löytäisi tarvitsemansa sairauskertomuksen kortilta ja potilas kuljettaisi reseptiään mukanaan lääkäriltä apteekille. Tällöin myös tietoturva olisi taattu, sillä kukaan ei pääsisi käsiksi kortilla oleviin resepteihin.

Tiedonsiirtokatkoksista huolimatta myös kansalliseen reseptipalvelimeen perustuva järjestelmä vaikuttaa lupaavalta jo mainitut hyödyt huomioon ottaen. XML-pohjaisten kielten ja ontologioiden käyttö reseptijärjestelmässä voisi tuoda vielä enemmänkin hyötyä sekä reseptejä kirjoittaville että niitä käyttäville. Jo pelkästään XML-muotoisiin resepteihin voidaan tehdä erilaisia kyselyjä, esimerkiksi XPath-kielellä ja näin selvittää tietyn potilaan aikaisemmat reseptit tai tietyn lääkärin määräämät reseptit tietyltä aikaväliltä. RFD(S)-kuvauskielten käyttäminen lisäisi vielä suhteiden merkitsemisen reseptijärjestelmään ja OWL-ontologiakieli syventäisi tätä entisestään. Näihin kuvaustietoihin voitaisiin tehdä jo paljon tarkempia kyselyjä, joilla voidaan selvittää esimerkiksi kaikki tietyn päivämäärän jälkeen kirjoitetut reseptit tai tiettyä lääkeainetta sisältävät reseptit. Näitä kyselyjä voivat hyödyntää myös muut tahot, kuten viranomaiset selvittäessään väärinkäytöksiä tai rikoksia tai kansaneläkelaitos selvittäessään korvausperusteita. Potilaallekin tästä voi olla merkittävää hyötyä tulevaisuudessa, mikäli suunnitteilla oleva terveystaali toteutuu. Portaalin on tarkoitus antaa potilaille mahdollisuus tarkastaa omia tietojaan, lääkityshistoriaansa ja potilasrekisteriään sekä tietoa erilaisista sairauksista terveydenhuollosta ja itsehoidosta. [STM 2004]

Sähköisessä reseptissä on siis vielä kehitettävää, mutta siinä on myös valtavasti potentiaalia. Kun tietoyhteiskuntamme on vahvasti siirtymässä paperittomaan aikakauteen,

ei kai ole ihmekään, että samaa halutaan myös resepteille. Tässä tapauksessa siirtyminen sähköiseen reseptiin saattaa jopa olla suurempi parannus kuin on odotettu, tai mitä muissa tapauksissa on todistettu. On totta, että sähkö- ja tiedonsiirtokatkokset sekä laitteiden viat voivat vaikeuttaa reseptin kirjoittamista ja toimittamista, mutta samalla tavalla myös paperiresepti voi kadota matkalla apteekkiin tai olla liian epäselvä käytettäväksi. Sähköinen resepti kuitenkin vähentää sitä käsittelevän henkilöstön työtä, sillä se on luettavissa samalla tavalla jokaisessa järjestelmässä ilman uudelleen syöttöä järjestelmästä toiseen. Myös väärinkäytöksiä voidaan näin vähentää, sillä reseptejä voivat kirjoittaa vain oikeutetut lääkärit järjestelmään tunnistautumisen ansiosta.

Suomessa sähköisen reseptin kehitys on annettu pilotin päättymisen jälkeen Kansaneläkelaitoksen hoiviin. Heinäkuusta 2006 eReseptinä tunnettu projekti on nyt edennyt ainakin vaatimusmäärittelyihin asti ja konsulttina toiminut Accenture Oy on laatinut järjestelmälle yleiskuvauksen, vaatimukset, tietosisällöt ja prosessikuvaukset. Jäämmekin nyt innolla odottamaan, milloin sähköinen resepti on meidän kaikkien käytössä.  
[STM 2006]

## LÄHTEET

14/2003. Laki sähköisistä allekirjoituksista. 24.1.2003. [Finlex – valtion säädöstietopankin www-sivulla]. Saatavissa: <http://www.finlex.fi/fi/laki/ajantasa/2003/20030014>

95/46/EY. Euroopan parlamentin ja neuvoston direktiivi yksilöiden suojelusta henkilötietojen käsittelyssä ja näiden tietojen vapaasta liikkuvuudesta. 24.10.1995. Luxemburg. [Verkkodokumentti]. Saatavissa: [http://www.europa.eu.int/smartapi/cgi/sga\\_doc?smartapi!celexapi!prod!CELEXnumdoc&lg=fi&numdoc=31995L0046&model=guichett](http://www.europa.eu.int/smartapi/cgi/sga_doc?smartapi!celexapi!prod!CELEXnumdoc&lg=fi&numdoc=31995L0046&model=guichett)

523/1999. Henkilötietolaki. 22.4.1999. Helsinki. [Finlex – valtion säädöstietopankin www-sivulla]. Saatavissa: <http://www.finlex.fi/fi/laki/alkup/1999/19990523>

559/1994. Laki terveydenhuollon ammattihenkilöistä. 28.6.2004. Helsinki. [Finlex – valtion säädöstietopankin www-sivulla]. Saatavissa: <http://www.finlex.fi/fi/laki/alkup/1994/19940559>

564/1994. Asetus terveydenhuollon ammattihenkilöistä. 28.6.1994. Naantali. [Finlex – valtion säädöstietopankin www-sivulla]. Saatavissa: <http://www.finlex.fi/fi/laki/alkup/1994/19940564>

621/1999. Laki viranomaisten toiminnan julkisuudesta. 21.5.1999 Helsinki. [Finlex – valtion säädöstietopankin www-sivulla]. Saatavissa: <http://www.finlex.fi/fi/laki/alkup/1999/19990621>

726/2003. Sosiaali- ja terveysministeriön asetus lääkkeen määräämisestä. 6.8.2003. Helsinki. [Finlex – valtion säädöstietopankin www-sivulla]. Saatavissa: <http://www.finlex.fi/fi/laki/alkup/2003/20030726>

771/2003. Sosiaali- ja terveystieteiden ministeriön asetus sähköisen lääkemääräyksen kokeilusta. 20.8.2003. Helsinki. [Finlex – valtion säädöstietopankin www-sivulla]. Saatavissa: <http://www.finlex.fi/fi/laki/alkup/2003/20030771>

785/1992. Laki potilaan asemasta ja oikeuksista. 17.8.1992. Helsinki. [Finlex – valtion säädöstietopankin www-sivulla]. Saatavissa: <http://www.finlex.fi/fi/laki/alkup/1992/19920785>

1224/2004. Sairausvakuutuslaki. 21.12.2004. Helsinki. [Finlex – valtion säädöstietopankin www-sivulla]. Saatavissa: <http://www.finlex.fi/fi/laki/alkup/2004/20041224>

1999/93/EY. Euroopan parlamentin ja neuvoston direktiivi sähköisiä allekirjoituksia koskevista puitteista. 13.12.1999. Bryssel. [Verkkodokumentti]. Saatavissa: [http://europa.eu.int/smartapi/cgi/sga\\_doc?smartapi!celexapi!prod!CELEXnumdoc&numdoc=31999L0093&model=guichett&lg=fi](http://europa.eu.int/smartapi/cgi/sga_doc?smartapi!celexapi!prod!CELEXnumdoc&numdoc=31999L0093&model=guichett&lg=fi)

ABA. 2006. Digital Signature Guidelines Tutorial. American Bar Association. Section of Science and Technology. Information Security Committee. [Verkkodokumentti]. [Viitattu 21.7.2006]. Saatavissa: <http://www.abanet.org/scitech/ec/isc/dsg-tutorial.html>

Andrews Tony, Curbera Francisco, Dholakia Hitesh, Klein Johannes, Leymann Frank, Liu Kevin, Roller Dieter, Smith Doug, Thatte Satish, Trickovic Ivana & Weerawarana Sanjiva. Business Process Execution Language for Web Services, Version 1.1. [pdf-tallenne]. 2003. [viitattu 20.7.2006]. BEA Systems, International Business Machines Corporation, Microsoft Corporation Inc, SAP AG, Siebel Systems. Saatavissa: <ftp://www6.software.ibm.com/software/dev>

Antoniou Grigoris, van Harmelen Frank. 2004. A Semantic Web Primer. The MIT Press. 272 s. ISBN 0-262-01210-3.

Ball Edward, Chadwick David W., Mundy Darren. 2003. Patient Privacy in Electronic Prescription Transfer. IEEE Computer Society, IEEE Security & Privacy. [Verkkodokumentti]. [Viitattu 25.10.2006]. Saatavissa: LTY:n verkkotietokanta (IEEE), vaatii salasanan.

Bartel Mark, Boyer John, Fox Barb, LaMacchia Brian, Simon Ed (authors), Eastlake Donald, Reagle Joseph, Solo David (editors). 2002. XML-Signature Syntax and Processing. W3C Recommendation 12.2.2002. [Viitattu 28.7.2006]. Saatavissa: <http://www.w3.org/TR/xmldsig-core/>

Booth David, Haas Hugo, McCabe Francis, Newcomer Eric, Champion Michael, Ferris Chris, Orchard David (editors). 2004. Web Services Architecture. W3C Working Group Note 11.2.2004. [Verkkodokumentti]. [Viitattu 31.7.2006]. Saatavissa: <http://www.w3.org/TR/ws-arch/>

Box Don, Ehnebuske David, Kakivaya Gopal, Layman Andrew, Mendelsohn Noah, Nielsen Henrik, Thatte Satish, Winer Dave. 2000. Simple Object Access protocol (SOAP) 1.1. W3C Note 8.5.2000. [Viitattu 1.8.2006]. Saatavissa: <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

Bray Tim, Paoli Jean, Sperberg-McQueen C.M., Maler Eve, Yergeau François, Cowan John (editors). 2004. Extensible Markup Language (XML) 1.1. W3C Recommendation 4.2.2004, edited in place 15.4.2004. [Viitattu 3.8.2006]. Saatavissa: <http://www.w3.org/TR/xml11/>

Chadwick D.W., Mundy D. 2003. Policy Based Electronic Transmission of Prescriptions. Proceedings of the 4<sup>th</sup> International Workshop on Policies for Distributed Systems and Networks. [Verkkodokumentti]. [Viitattu 10.7.2006]. Saatavissa: LTY:n verkkotietokanta (IEEE), vaatii salasanan.



Chadwick D.W., Mundy D. 2004. An XML Alternative for Performance and Security: ASN.1. IEEE Computer Society. [Verkkodokumentti]. [Viitattu 1.11.2006]. Saatavissa: LTY:n verkkotietokanta (IEEE), vaatii salasanan.

Christensen Erik, Curbera Francisco, Meredith Greg, Weerawarana Sanjiva. 2001. Web Services Description Language (WSDL) 1.1. W3C Note 15.3.2001. [Viitattu 3.8.2006]. Saatavissa: <http://www.w3.org/TR/wsdl>

Clark James, DeRose Steve. 1999. XML Path Language (XPath) Version 1.0. W3C Recommendation 16.10.1999. [Viitattu 21.8.2006]. Saatavissa: <http://www.w3.org/TR/xpath>

Common Criteria. 2006. Common Criteria, An Introduction. [Verkkodokumentti]. [Viitattu 9.8.2006]. Saatavissa: <http://www.commoncriteriaportal.org/public/files/ccintroduction.pdf>

Cover Robin. 2003. Business Process Modeling Language (BPML). Cover Pages Technology Report. [Verkkodokumentti]. Päivitetty 29.8.2003. [Viitattu 7.11.2006]. Saatavissa: <http://xml.coverpages.org/bpml.html>

Cyberspace Center. 1999. Guide to Smart Card Technology. Hong Kong University of Science and Technology. [Verkkodokumentti]. [Viitattu 18.7.2006]. Saatavissa: <http://www.cyber.ust.hk/public/pub.htm>

Department of Health. 2006. [Department of Health www-sivuilla]. Electronic Prescription Service. Englanti. [Verkkodokumentti]. [Viitattu 10.7.2006]. Saatavissa: <http://www.dh.gov.uk/PolicyAndGuidance/MedicinesPharmacyAndIndustry/Prescriptions/ElectronicTransmissionOfPrescriptions/fs/en>

Dietzel Gottried T.W., Riepe C.. 2006. Modernizing healthcare in Germany by introducing the eHealthcard. Federal Ministry for Health and Social Security, Bonn-Berlin.

[Verkkodokumentti]. [Viitattu 14.7.2006]. Saatavissa:

<http://www.dimdi.de/static/de/ehealth/literatur/sim52-18-22-germany.pdf>

ElRes. [Mediweb Oy:n www-sivuilla]. [Viitattu 27.6.2006]. Saatavissa:

[http://www.mediweb.fi/elres\\_en.html](http://www.mediweb.fi/elres_en.html)

Endrei M., Ang J., Arsanjani A., Chua S., Comte P., Krogdahl P., Luo M. & Newling T. 2004. Patterns: Service Oriented Architecture and Web Services. [pdf-tallenne]. [Viitattu 1.8.2006]. IBM. Saatavissa: <http://www.redbooks.ibm.com/redbooks/pdfs/sg246303.pdf>

e-recept Stocholm. 2006. [e-recept Stockholm www-sivuilla]. Apoteket AB & Stockholms Läns Landsting I Samarbete. [Verkkosivusto]. Päivitetty 29.6.2006. [Viitattu 11.7.2006].

Saatavissa: [http://www.e-receptstockholm.se/imcms/servlet/GetDoc?meta\\_id=1011](http://www.e-receptstockholm.se/imcms/servlet/GetDoc?meta_id=1011)

FORTH. 2003. FORTH Institute of Computer Science: The RDF Query Language (RQL). [verkkodokumentti]. Päivitetty 18.7.2003. [Viitattu 22.9.2006]. Saatavissa:

<http://athena.ics.forth.gr:9090/RDF/RQL/index.html>

Gemplus. 2006. [Gemplus:n www-sivuilla] About Smartcards; What is a smart card?

[viitattu 19.7.2006]. Saatavissa: <http://www.gemplus.com/smart/cards/basics/what.html>

Ghinea Gheorgita, Moradi Arah, Asgari Shervin, Serif Tacha. 2006. Electronic Prescriptions: A ubiquitous Solution using Jini. Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops. [Verkkodokumentti]. [Viitattu 10.7.2006]. Saatavissa: LTY:n verkkotietokanta (IEEE), vaatii salasanan.

GSA. 2004. Government Smart Card Handbook. U.S. General Services Administration. Helmikuu 2004. [Verkkodokumentti]. [Viitattu 19.7.2006]. Saatavissa: [http://www.smartcardalliance.org/pdf/industry\\_info/smartcardhandbook.pdf](http://www.smartcardalliance.org/pdf/industry_info/smartcardhandbook.pdf)

Gudgin Martin, Hadley Marc, Mendelsohn Noah, Moreau Jean-Jacques, Nielsen Henrik (editors). 2003. SOAP Version 1.2 Part 1: messaging Framework. W3C recommendation 24.6.2003. [Verkkodokumentti]. [Viitattu 2.8.2006]. Saatavissa: <http://www.w3.org/TR/soap12-part1/>

Hart A., Decker S.. 2005. Optimized Index Structures for Querying RDF from the Web. Web Congress 2005. [Verkkodokumentti]. [Viitattu 8.11.2006]. Saatavissa: LTY:n verkkotietokanta (IEEE), vaatii salasanan.

HHS. 2006. [United States Department of Health and Human Services www-sivuilla]. Office of the National Coordinator for Health Information Technology (ONC), E-Prescribing. Päivitetty 19.1.2006. [Viitattu 17.7.2006]. Saatavissa: <http://www.hhs.gov/healthit/e-prescribing.html>

HL7. 2006. [HL7:n www-sivuilla]. Health Level Seven. [Viitattu 9.8.2006]. Saatavissa: <http://www.hl7.org/>

HL7 Finland. 2005. Tietoturvallinen kontekstinhallinta – soveltamisohje versio 1. HL7 Finland Teknisen komitean 17.8.2005 käsittelemä, HL7 Finland dokumenttiarkistoon siirrettävä versio. OID 1.2.246.777.11.2005.16. [Verkkodokumentti]. [Viitattu 10.8.2006]. Saatavissa: [http://virtual.vtt.fi/virtual/hl7/cda/api-rajapinnat/tietoturvallinen\\_kh\\_v1.doc](http://virtual.vtt.fi/virtual/hl7/cda/api-rajapinnat/tietoturvallinen_kh_v1.doc)

Hyppönen Hannele (toim.). 2005. Sähköisen reseptin pilotoinnin arviointi, vaihe 1, loppuraportti. Osaavien keskusten verkoston julkaisuja 1/2005. Helsinki. 175 s. ISBN 951-33-1183-X.

Hyppönen Hannele, Salmivalli Lauri, Suomi Reima. 2005. Organizing for a National Infrastructure Project: The Case of the Finnish Electronic Prescription. Proceedings of the 38<sup>th</sup> Hawaii International Conference on System Sciences 2005. [Verkkodokumentti]. [Viitattu 26.6.2006]. Saatavissa: LTY:n verkkotietokanta (IEEE), vaatii salasanan

Hyppönen Hannele. 2006. Johdatus arvioinnin osa-alueisiin. Sähköisen reseptin arvioinnin loppuseminaari 22.5.2006. [Verkkodokumentti]. [Viitattu 5.7.2006]. Saatavissa: [http://sty.stakes.fi/NR/rdonlyres/0CA1826E-D2BE-4F60-A0A0-4C660C89F29A/4526/Hyppönen\\_johdatus.pdf](http://sty.stakes.fi/NR/rdonlyres/0CA1826E-D2BE-4F60-A0A0-4C660C89F29A/4526/Hyppönen_johdatus.pdf)

ISO. 2006. [ISO:n www-sivuilla]. International Organization for Standardization. Päivitetty 10.7.2006. [Viitattu 18.7.2006]. Saatavissa: <http://www.iso.org/iso/en/ISOOnline.frontpage>

ITSEC. 1991. Information Technology Security Evaluation Criteria (ITSEC), Provisional Harmonized Criteria. Heinäkuu 1991. Euroopan Yhteisön virallisten julkaisujen toimisto. Bryssel, Luxembourg. 171 s. [Verkkodokumentti]. [Viitattu 18.7.2006] Saatavissa : [http://www.ssi.gouv.fr/site\\_documents/ITSEC/ITSEC-uk.pdf](http://www.ssi.gouv.fr/site_documents/ITSEC/ITSEC-uk.pdf)

Java. 2006. [Sun Developer Network www-sivuilla]. Java Card Technology Overview. [Verkkoartikkeli]. [Viitattu 18.7.2006]. Saatavissa: <http://java.sun.com/products/javacard/overview.html>

Jena. 2006. [Jenan www-sivuilla] Jena – Semantic Web Framework. [Viitattu 2.10.2006]. Saatavissa: <http://jena.sourceforge.net/index.html>

Kansaneläkelaitos (Kela). 2006. [Kelan www-sivuilla]. Päivitetty 16.3.2006. [Viitattu 26.6.2006]. Saatavissa: [www.kela.fi](http://www.kela.fi)

Karvounarakis Gregory, Alexaki Sofia, Christophides Vassilis, Plexousakis Dimitris, Scholl Michel. 2002. RQL: A Declarative Query Language for RDF. [Verkkodokumentti]. [Viitattu 8.11.2006]. Saatavissa: LTY:n verkkotietokanta (IEEE), vaatii salasanan.

Kilpivuori Teemu. 2006. Sähköinen resepti ja Pegasos-järjestelmä. [sähköposti]. 25.7.2006. VM-Data, ohjelmistotuoteratkaisut.

Klaukka Timo, Helin-Salmivaara Arja, Huupponen Risto, Idänpään-Heikkilä Juhana E. (toim.). Joka toinen resepti terveystieteistä. Suomen Lääkärilehti 15-16/2005 vsk 61. [Verkkodokumentti]. [Viitattu 22.6.2006]. Saatavissa: [http://www.kela.fi/in/internet/liite.nsf/NET/120506140615MP/\\$File/SLL152006-1724.pdf?openElement](http://www.kela.fi/in/internet/liite.nsf/NET/120506140615MP/$File/SLL152006-1724.pdf?openElement)

Klein M.. 2001. XML, RDF, and relatives. Intelligent Systems, IEEE. [Verkkodokumentti]. [Viitattu 8.11.2006]. Saatavissa: LTY:n verkkotietokanta (IEEE), vaatii salasanan.

Koponen-Piiroinen Hanna-Maija, Kiiski Markku. 2001. Sähköistä reseptiä koskeva esiselvitys. Sosiaali- ja terveysministeriön työryhmämuistioita 2001:27. Helsinki. [Verkkodokumentti]. [Viitattu 26.6.2006]. Saatavissa: <http://pre20031103.stm.fi/suomi/pao/julkaisut/paoisallys120.htm>

Larson Mats. 2001. Medcom 7/12 2001. [Verkkodokumentti]. Päivitetty 15.10.2001. [Viitattu 11.6.2006]. Saatavissa: [http://www.medcom1-4.dk/internetstrategi/konference/presentationer/12\\_%20Mats\\_Larson.pdf](http://www.medcom1-4.dk/internetstrategi/konference/presentationer/12_%20Mats_Larson.pdf)

Larson Mats, Nerander K.G, Malmqvist Gustav. 2003. Sjunet – The National IT infrastructure for Healthcare in Sweden. Information and Communication Technology in Arctic. An International Conference of the Arctic Council 20.-21.10.2003. Saatavissa: <http://vefir.unak.is/ICTConference/Larson.pdf>

Lehto Esa, Hohteri Harri. 2003. Sähköinen lääkemääräys, sanomakuvaus. MediWeb Oy. Päivitetty 10.3.2003.

Linnea. 2006. [Receptumin www-sivuilla]. Linnea apteekkiohjelmistot. Päivitetty 2006. [Viitattu 24.10.2006]. Saatavissa: <http://www.receptum.fi/linnea/index.htm>

Luottokunta. 2006. [Luottokunnan www-sivuilla]. EMV-sirukortit. [Viitattu 18.7.2006]. Saatavissa: <http://www.luottokunta.fi/fi/emv-sirukortit/>

MedCom IV. 2003. [MedComin www-sivuilla]. MedCom IV – Status, plans and projects. MedCom – The Danish Healthcare Data Network / Dec.2003/MC-S177. [Verkkodokumentti]. Päivitetty 12.2003. [Viitattu 13.7.2006] Saatavissa: <http://www.medcom1-4.dk/publikationer/publikationer/MedCom4-ENGELSK.pdf>

Minnesota Department of Labor & Industry. 2002. Electronic Data Interchange (EDI) Implementation Guide. [Verkkodokumentti]. [Viitattu 11.8.2006]. Saatavissa: <http://www.doli.state.mn.us/edi.html>

Mitra Nilo (editor). 2003. SOAP Version 1.2 Part 0: Primer. W3C Recommendation 24.6.2003. [Verkkodokumentti]. [Viitattu 2.8.2006]. Saatavissa: <http://www.w3.org/TR/soap12-part0/>

Multanen Timo. 2006. Sähköinen resepti ja healthNet-järjestelmä. [sähköposti]. 11.8.2006. TietoEnator, Healthcare & Welfare.

NHS. 2006. [NHS:n www-sivuilla]. How it works. [Verkkodokumentti]. Päivitetty 27.3.2006. [Viitattu 10.7.2006]. Saatavissa: <http://www.connectingforhealth.nhs.uk/eps/howitworks/>

Pajukoski Marja. 2006. Oikeudellinen arviointi – Potilaan oikeudellinen asema, Reseptitietokanta, Asetuksen muutos. Sähköisen reseptin arvioinnin loppuseminaari 22.5.2006. [Verkkodokumentti]. [Viitattu 6.7.2006]. Saatavissa: <http://sty.stakes.fi/NR/rdonlyres/0CA1826E-D2BE-4F60-A0A0-4C660C89F29A/4530/Pajukoski.pdf>

Passmore David, Freeman John. 1996. The Virtual LAN Technology Report. [Verkkodokumentti]. [Viitattu 11.8.2006]. Saatavissa: [http://www.3com.com/other/pdfs/solutions/en\\_US/20037401.pdf](http://www.3com.com/other/pdfs/solutions/en_US/20037401.pdf)

Pegasos. 2005. [VM-datan www-sivuilla]. Pegasos potilastietojärjestelmä. [Verkkodokumentti]. [Viitattu 25.7.2006]. Saatavissa: [http://www.wmdata.fi/wmwebb/Services/files/Pegasos\\_yleisesite2005.pdf](http://www.wmdata.fi/wmwebb/Services/files/Pegasos_yleisesite2005.pdf)

Peltonen Hannele. Tietoteknologia on vain työkalu. Dialogi 2/2006. [Verkkolehti]. [Viitattu 21.6.2006]. Saatavissa: <http://www2.stakes.fi/dialogi/06/d206/24.htm>

Perlmutter Bruce, Zarkower Jonathan. 2001. VPN: Virtuaaliset yksityisverkot. Edita Oyj, IT Press. Helsinki. 270 s. ISBN 951-826-468-6

Poliklinikka.net. 2004. [Mtv3:n www-sivuilla]. [Päivitetty 20.9.2004]. [Viitattu 26.6.2006]. Saatavissa: [http://www.mtv3.fi/poliklinikka/laakkeet.shtml/128923?L\\*\\*keinfo](http://www.mtv3.fi/poliklinikka/laakkeet.shtml/128923?L**keinfo)

Profium. 2006 [Profiumin www-sivuilla]. Ratkaisut ja tuotteet. Päivitetty 5.10.2006. [Viitattu 16.10.2006]. Saatavissa: <http://www.profium.com/index.php?id=472>

Protégé. 2006. Protégé – a resource for biomedical ontologies and knowledge bases supported by the National Library of Medicine. Stanford Medical Informatics. [Viitattu 2.10.2006]. Saatavissa: <http://protege.stanford.edu/>

RDF. 2006. RFD Tutorial. W3Schools Forum. [Verkkodokumentti]. [Viitattu 5.9.2006].  
Saatavissa: [http://www.w3schools.com/rdf/rdf\\_owl.asp](http://www.w3schools.com/rdf/rdf_owl.asp)

RIDE. 2006. RIDE D.2.1.1 – Current European practices in eHealth: A Brief Survey of the Initiative by the German Federal Ministry of Health. [Verkkodokumentti]. Päivitetty 15.6.2006. [Viitattu 14.7.2006]. Saatavissa:  
<http://www.srdc.metu.edu.tr/webpage/projects/ride/deliverables/RIDED.2.1.1%20-%20CurrentPracticesGermany.doc>

RSA. 2000. RSA Laboratories' Frequently Asked Questions About Today's Cryptography. Version 4.1. RSA Security Inc. [Verkkodokumentti]. [Viitattu 24.7.2006]. Saatavissa:  
<http://www.rsasecurity.com/rsalabs/node.asp?id=2152>

Ruotsalainen Pekka. 2006. Sähköisen reseptin arvioinnin loppuseminaari 22.5.2006. [Verkkodokumentti]. [Viitattu 6.7.2006]. Saatavissa:  
<http://sty.stakes.fi/NR/rdonlyres/0CA1826E-D2BE-4F60-A0A0-4C660C89F29A/4529/Ruotsalainen.pdf>

Salix. 2001. [Pharmadatan www-sivuilla]. Päivitetty 9.11.2001. [Viitattu 24.10.2006].  
Saatavissa: <http://www.pharmadata.fi/>

Setec. 2006. [Setecin www-sivuilla]. Päivitetty 12.4.2006. [Viitattu 18.7.2006]. Saatavissa:  
<http://www.setec.com/index.php?id=10&L=0>

Singh Munindar P., Huhns Michael N.. 2005. Service-Oriented Computing; Semantics, Processes, Agents. John Wiley & Sons, Ltd. West Sussex, England. 549 s. ISBN 0-470-09148-7

Sjöborg Bengt, Bäckström Tobias, Arvidsson Lars-Bertil, Andersén-Karlsson Eva, Blomberg L. Bengt, Eiermann Birgit, Eliasson Marie, Henriksson Kjell, Jacobsson Lennart,



Jacobsson Ulf, Julander Margaretha, Kaiser per-Olof, Landberg Carina, Larsson Jonas, Molin Björn, Gustafsson Lars L.. 2006. Design and implementation of a point-of-care computerized system for drug therapy in Stockholm metropolitan health region – Bridging the gap between knowledge and practice. Elsevier Ireland Ltd. – International journal of medical informatics 2006. 10 s. [Verkkodokumentti] [Viitattu 11.6.2006] Saatavissa: LTY:n verkkotietokanta (Elsevier), vaatii salasanan.

SOAP. 2006. SOAP Tutorial. W3Schools Forum. [Verkkodokumentti]. [Viitattu 2.8.2006]. Saatavissa: <http://www.w3schools.com/soap/default.asp>

Sotomayor Borja. 2005. The Globus Toolkit 4 Programmer's Tutorial. University of Chigago, Department of Computer Science. [Verkkodokumentti]. Saatavissa: <http://gdp.globus.org/gt4-tutorial/multiplehtml/>

STM. 2004. [STM:n www-sivuilla]. Sosiaali- ja terveysministeriö tiedottaa; Mitä tietoa haluaisit Internetissä terveyden edistämiseksi? Tiedote 55/2004. Päivitetty 11.2.2004. [Viitattu 25.10.2006]. Saatavissa: <http://www.stm.fi/Resource.phx/publishing/documents/1495/index.htx>

STM. 2006. [STM:n www-sivuilla]. Sosiaali- ja terveydenhuollon tiedonhallinta; eReseptin vaatimusmäärittelyt.. Päivitetty 3.7.2006. [Viitattu 25.10.2006]. Saatavissa: <http://www.stm.fi/Resource.phx/vastt/tietoh/eresmaar.htx>

Suomen Apteekkariliitto. 2004. SAL-vuosikatsaus 2004. 27 s. Saatavissa: <http://www.apteekkariliitto.fi/BinaryServlet?rs=623/659/812/:6570/>

Suomen Apteekkariliitto. 2005. SAL-vuosikatsaus 2005. 27 s. Saatavissa: <http://www.apteekkariliitto.fi/BinaryServlet?rs=623/659/812/:7421/>

Tenhunen Emmi. 2006. Sähköinen resepti - potilaskysely. Sähköisen reseptin arvioinnin loppuseminaari 22.5.2006. [Verkkodokumentti]. [Viitattu 6.7.2006]. Saatavissa:

<http://sty.stakes.fi/NR/rdonlyres/0CA1826E-D2BE-4F60-A0A0-4C660C89F29A/4528/Tenhunen.pdf>

TietoEnator. 2006. Effica eResepti Käyttäjänopas. 8.9.2006. 18 s.

Tikkanen Tomi. 2004. Sähköinen lääkemääräys – pilotin ensimmäisen vaiheen kokemuksia Pohjois-Karjalan sairaanhoitopiirissä. 2.6.2004. 12 s.

Tuikka Tommi, Kanala Sari. 2001. XML Ohjelmoinnin perusteet. Oy Edita Ab. Helsinki. 132 s. ISBN 951-826-407-4

UN/EDIFACT. 2006. [UNECE:n www-sivuilla] UN/EDIFACT, United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport. Päivitetty 15.2.2006. [Viitattu 11.8.2006]. Saatavissa:

<http://www.unece.org/trade/untdid/welcome.htm>

Viestintävirasto. 2004. [Viestintäviraston www-sivuilla]. Tietoturvallisuuden perusteet: VPN. Päivitetty 1.9.2004. [Viitattu 10.8..2006]. Saatavissa:

<http://www.ficora.fi/suomi/tietoturva/vpn.htm>

Viestintävirasto. 2005. [Viestintäviraston www-sivuilla]. Sähköinen asiointi: Tietoa allekirjoittajalle. Päivitetty 18.5.2005. [Viitattu 20.7.2006]. Saatavissa:

<http://www.ficora.fi/suomi/tietoturva/sahkoinen.htm>

Wanscher Christina E. 2006. MedCom and the story of the Danish Health Data Network. Danish Centre for Health Telematics. [Artikkeli].

Wikipedia. 2006. Subscriber Identity Module. [Verkkotietosanakirja]. Päivitetty 5.7.2006. [Viitattu 18.7.2006]. Saatavissa: <http://en.wikipedia.org/wiki/SIM>

XPath. 2006. XPath Tutorial, W3Schools. [verkkodokumentti]. [Viitattu 21.8.2006].

Saatavissa: <http://www.w3schools.com/xpath/default.asp>

Yang Yanjian, Han Xiaoxi, Bao Feng, Deng Robert H.. 2004. A Smart-Card-Enabled Privacy Preserving E-Prescription System. IEEE Transactions on Information Technology in Biomedicine, Vol. 8, No. 1, March 2004. [Verkkodokumentti]. [Viitattu 25.10.2006].

saatavissa: LTY:n verkkotietokanta (IEEE), vaatii salasanan.

YT Tieto. 2006. Tieto- ja asiantuntijapalvelu, Yhdessä terveyttä Oy. [Verkkodokumentti].

[Viitattu 22.6.2006]. Saatavissa:

[http://www.pshp.fi/proke/pirke/Dokumentit/YTTieto\\_yleisesittely\\_kes%C3%A4kuu2006.pdf](http://www.pshp.fi/proke/pirke/Dokumentit/YTTieto_yleisesittely_kes%C3%A4kuu2006.pdf)

# 1 LAINSÄÄDÄNTÖ SUOMESSA

Sähköisessä reseptissä tietoverkkojen välillä liikkuu henkilökohtaista arkaluontoista tietoa. Paperisen reseptin muodossa potilas hallitsee itse tätä arkaluontoista tietoa ja sen joutumista muiden käsiin. Sähköisessä reseptissä potilaalla ei kuitenkaan ole mahdollisuutta huolehtia itse reseptin etenemisestä apteekkiin, jolloin hänen yksityisyytensä on pidettävä turvattuna ja hänelle on luotava luottamus järjestelmään ja sen toimintaan. Potilaalla on pysyttävä samat oikeudet ja edut kuin paperisen reseptin kanssakin. Tästä esimerkkinä toimii hyvin potilaan mahdollisuus valita apteekki, josta hän lääkkeensä noutaa. [Koponen-Piironen et al. 2001 s.22]

## 1.1 Potilaan oikeudet

Euroopan parlamentin ja neuvoston antaman direktiivin (95/46/EY) mukaan jokaisen jäsenvaltion on turvattava asukkaidensa perusoikeudet, perusvapaudet ja yksityisyys. Suomessa tämä on toteutettu perustus- ja henkilötietolain (523/1999) avulla. Tämän lisäksi terveydenhuollossa sovelletaan enimmäkseen lakia viranomaisten toiminnan julkisuudesta (621/1999) ja potilaslakia (785/1992). Henkilötietolaki pyrkii ylläpitämään henkilön tietosuojaa ja perusoikeuksia erilaisia henkilötietoja käsitellessä. Se myös tukee tietojenkäsittelytapojen kehittämisessä ja niiden noudattamisessa. Lakia sovelletaan henkilötietojen manuaaliseen ja automaattiseen käsittelyyn, joten se koskee myös potilasrekisterejä ja rekisteritietokantaa, jossa potilaan tietoja säilytetään. Rekisterin käyttö on oltava etukäteen suunniteltua ja rekisterinpitäjän, tässä tapauksessa terveydenhuolto-organisaation ja Kansaneläkelaitoksen, on käsiteltävä tietoja laillisesti ja huolellisesti, niin ettei rekisterissä olevan henkilön yksityisyyttä missään vaiheessa loukata. Rekisterinpitäjän on siis huolehdittava tarpeellisesta tietoturvasta, sillä tiedot eivät saa joutua ulkopuolisten käsiin ja niiden on pysyttävä muuttumattomina ja virheettöminä. [523/1999; Hyppönen 2005 s.33-38; Koponen-Piironen et al. 2001 s.22-23]

## Liite1

Peruseriaatteena rekisterin tietojen käsitteleminen on kielletty ilman kyseisen henkilön suostumusta. Tämän takia sähköisen reseptin käyttöön ja sen tallentamiseen reseptitietokantaan, sekä lääkärin tai farmaseutin pääsyyn tietokantaan tutkimaan aikaisempia reseptejä, on annettava kirjallinen suostumus. Lisäksi suostumuksen antajan on saatava ennen suostumuksen antamista tarpeellinen määrä tietoa, jonka perusteella hän on pätevä antamaan vapaaehtoisen suostumuksensa. Poikkeuksena tietoja voivat käsitellä sähköisen reseptin näkökulmasta terveydenhuollon toimiyksiköt ja ammattihenkilökunta hoidon vaatimalta osalta, vakuutuslaitokset niiden vastuun selvittämiseksi sekä sosiaalihuollon (Kansaneläkelaitos) viranomaiset, jotka määräävät etuuksia tai korvauksia. Lisäksi tietoja voidaan käyttää tieteellistä tai historiallista tutkimusta sekä tilastointia varten, jos henkilötietoja ei paljasteta tai rekisteriä käytetään vain tilastollisiin tarkoituksiin niin, etteivät yksittäiset henkilöt ole siitä tunnistettavissa. [523/1999; Hyppönen 2005 s.33-38; Koponen-Piironen et al. 2001 s.22-23]

Laki potilaan asemasta ja oikeuksista koskee terveyden- ja sairaanhoidon järjestämistä henkilölle ja potilaalla tarkoitetaan jokaista terveyden- tai sairaanhoitopalveluja käyttävää tai niiden kohteena olevaa henkilöä. Jokaisella potilaalla on lain mukaan itsemääräämisoikeus, mikä sähköisen reseptin yhteydessä tarkoittaa, että potilas voi itse päättää haluaako hän sähköisen vai tavallisen paperisen reseptin. Laissa myös määrätään, että kaikki potilasasiakirjoihin sisältyvät tiedot ovat salassa pidettäviä ja tätä voidaan soveltaa myös reseptirekisteriin. Tällöin Kansaneläkelaitos sekä terveydenhuolto-organisaatiot ovat velvoitettuja pitämään resepteissä kulkevat tiedot salaisina ja muuttumattomina, sekä takaamaan tarvittavan tietoturvan niiden liikkeessa toimipisteiden välillä. Potilaalla on myös oikeus tarkastaa potilasrekisterissä tai reseptitietokannassa olevat häntä koskevat tiedot, eli esimerkiksi nähdä tiedot hänen terveydentilastaan tai hänelle määrätyt lääkkeet.[785/1992; Hyppönen 2005 s.55; Koponen-Piirainen et al.] 2001 s. 22-23]

## 1.2 Lääkkeen määrääminen

Laki terveydenhuollon ammattihenkilöistä (559/1994) pyrkii tuomaan lisäturvaa potilaille ja parantamaan terveydenhuollon palveluiden laatua lähinnä varmistamalla, että ammattihenkilöillä on tarpeellinen koulutus ja pätevyys kyseiseen toimeen ja valvomalla tätä. Lain 22 § säättää, että laillistetut lääkärit voivat päättää potilaan lääketieteellisestä tutkimuksesta, taudinmäärityksestä ja siihen liittyvästä hoidosta. Lisäksi pykälässä todetaan lääkärin oikeus määrätä potilaalle apteekista lääkkeitä, jotka liittyvät taudin hoitamiseen. Samainen pykälä on lisäksi tuottanut sosiaali- ja terveysministeriön asetuksen lääkkeen määräämisestä (726/2003), jossa määritetään paremmin erilaiset lääkkeet ja lääkärin oikeudet määrätä niitä. Lain 24 §:ssä määrätään, että terveydenhuollon ammattihenkilöiden yleinen ohjaus kuuluu sosiaali- ja terveysministeriölle ja heitä valvoo ja ohjaa terveydenhuollon oikeusturvakeskus. Vuonna 2000 lakiin lisättiin pykälä 24 a, joka yhdessä alkuperäisen lain 3 §:n kanssa määrää terveydenhuollon oikeusturvakeskuksen sen valvontaan kuuluvana osana ylläpitämään terveydenhuollon ammattihenkilöiden keskusrekisteriä. Rekisterissä tulee ammattihenkilöstä mainita seuraavat tiedot:

- nimi, henkilötunnus, kotiosoite
- toimiminen itsenäisenä ammatinharjoittajana, ammatinharjoittamisoikeus ja sen rajoittaminen tai poistaminen, ammatinharjoittamislupa ja sen peruminen sekä oikeus käyttää ammatinimikettä tai sen kieltäminen
- ammatinharjoittamisen perusteena oleva koulutus
- mahdolliset tiedot väärinkäytöksistä, viraltapanemisesta ja muusta vastaavasta
- tiedot erikoislääkärin oikeudesta ja siihen liittyvästä koulutuksesta.

Rekisteristä voidaan helposti tarkastaa lääkärin oikeus määrätä lääkkeitä, joten sitä voidaan näin ollen käyttää hyödyksi sähköisessä reseptissä lääkärin oikeuksien tarkastamiseen. Ongelmaksi muodostuvat kuitenkin lääketieteen opiskelijat, jotka voivat sosiaali- ja terveysministeriön asetuksen (564/1994) mukaan neljännen vuoden opintojensa jälkeen toimia tilapäisesti lääkärin toimessa ja määrätä lääkkeitä (ainoastaan) hoitamilleen potilaille. Lääketieteen opiskelijoita ei kuitenkaan merkitä terveydenhuollon

oikeusturvakeskuksen rekisteriin, joten heidän oikeuksiensa tarkastaminen on hankalampaa. Sosiaali- ja terveysministeriön asetus sähköisen lääkemääräyksen kokeilusta (771/2003) rajoittaa sähköisen reseptin käytön kokeilun ajaksi vain laillistettujen lääkärin ja hammaslääkärin käyttöön, sillä heidän oikeutensa kirjoittaa resepti voidaan tarkastaa terveydenhuollon oikeusturvakeskuksen rekisteristä. Asetuksessa ei kuitenkaan mainita, että reseptin kirjoitusoikeus tulisi tarkastaa juuri tästä rekisteristä. Kun lääketieteen opiskelija rinnastetaan lääkäriin, voitaisiin hänen oikeutensa myös sähköisen reseptin kirjoittamiseen tarkastaa tällöin kyseisen terveydenhuollon toimintayksikön rekisteristä. Jos tulevaisuudessa päätetään reseptin kirjoitusoikeuksien tarkastamisesta nimenomaan terveydenhuollon oikeusturvakeskuksen rekisteristä, on tehtävä muutoksia myös lakiin terveydenhuollon ammattihenkilöistä. [559/1994; 564/1994; 726/2003; 771/2003]

### **1.3 Sähköinen lääkemääräys**

Lääkemääräystä koskevaa sosiaali- ja terveysministeriön määräystä lääkkeiden määräämisestä (726/2003) käsiteltiin jo johdannossa, jossa kerrottiin mitä lääkemääräyksen tulee sisältää. Määräyksessä määritellään reseptilääkkeeksi lääke, jonka apteekki voi toimittaa ainoastaan lääkärin tai hammaslääkärin erillisellä määräyksellä. Samassa todetaan, että resepti on se lääkärin tai hammaslääkärin kirjoittama määräys, jolla apteekki voi toimittaa potilaalle lääkkeen. Resepti kirjoitetaan kansaneläkelaitoksen hyväksymälle lomakkeelle joko käsin, kirjoituskoneella tai automaattista reseptinkirjoitusohjelmaa käyttäen ja se on voimassa vuoden määräämis- tai uusimispäivästä alkaen. Aivan kuten potilasrekisterit ja reseptitietokanta, resepti on asiakirja, joka sisältää arkaluontoisia ja henkilökohtaisia tietoja ja se tulee pitää salassa. Reseptin tietoja ei siis voi luovuttaa ulkopuolisille ilman potilaan hyväksyntää. Määräyksessä todetaan myös, että sähköinen lääkemääräys voidaan laatia ja toimittaa apteekkiin erikseen säädetyllä tavalla. [726/2003]

Sähköisen lääkemääräyksen pilotointia varten sosiaali- ja terveysministeriö teki asetuksen sähköisen lääkemääräyksen kokeilusta (771/2003) vuoden 2003 syksyllä. Asetuksessa kuvailaan sähköisen reseptin välittämistä, toimittamista ja teknistä sisältöä koskevia

## Liite1

asioita, sekä kokeilussa toimijat. Sähköinen resepti määritetään laillistetun lääkärin tai hammaslääkärin tietoverkon avulla reseptitietokantaan toimittamaksi lääkemääräykseksi ja reseptitietokanta näiden reseptien ja niiden toimitustietojen säilytyspaikaksi, jota Kansaneläkelaitos ylläpitää. Vaikka sähköinen resepti pohjautuu paperiseen reseptiin ja lääkemääräysasetukseen, joudutaan kokeilussa poikkeamaan hieman lääkemääräyksen säädöksistä, sillä järjestelmä edellyttää muun muassa lähettäjien ja vastaanottajien tunnistamista. Näin ollen lääketieteen ja hammaslääketieteen opiskelijat eivät voi kokeilun aikana määrätä sähköistä reseptiä, sillä heidän tietojaan ei ole terveydenhuollon oikeusturvakeskuksen rekisterissä. Sähköisellä reseptillä voidaan lisäksi määrätä vain yhtä lääketta kerrallaan, ja sillä ei voi määrätä lainkaan huumausainetta, pro auctore-lääkettä tai lääketilausta eikä sitä voi uusia tai muuttaa kirjalliseksi. Kokeilun aikana ei myöskään voida sähköisesti määrätä lääkkeitä eläimille. [771/2003; Koponen-Piironen et al. 2001]

Asetuksen mukaan potilaan on annettava kirjallinen suostumus sähköisen reseptin käyttöön, vanhojen reseptien selailuun ja lääkkeen toimittamista koskevien tietojen siirtämiseen johonkin toiseen terveydenhuollon toimipisteeseen. Potilas voi milloin tahansa peruuttaa suostumuksensa, jos niin haluaa. Hän voi niin ikään suojata oman sähköisen reseptinsä henkilökohtaisella tunnusluvulla eli Pinokoodilla, jolloin reseptin tietoja voivat käsitellä ainoastaan ne, joille potilas on tunnuslukunsa antanut. Lääkärin kirjoitettua sähköinen resepti, hänen tulee tulostaa potilalle kirjallinen potilasohje, ellei potilas tästä varta vasten kieltäydy. Potilasohjeessa ovat potilaan nimi, lääkkeen nimi, vahvuus, käyttötarkoitus, lääkkeen määrääjä ja päiväys sekä mahdollinen PIN-koodi, jolla reseptin saa avattua apteekissa. Reseptit säilyvät tietokannassa 20 kuukautta. [771/2003; Koponen-Piironen et al. 2001]

Apteekissa reseptitietokantaa käyttävien farmaseuttien ja proviisorien tulee tunnistautua järjestelmään sähköisen tunnisteen avulla. Tunnistetut henkilöt voivat käsitellä haettavien lääkkeiden reseptejä ja merkitä niihin toimittamistietoja sekä muita reseptejä potilaan erillisellä suostumuksella. Potilaan on puolestaan pystyttävä tarvittaessa todistamaan henkilöllisyytensä sekä tuotava mukanaan potilasohje ja sairausvakuutuskortti.



Kansaneläkelaitoksella on potilaan suostumuksesta ja PIN-koodista huolimatta sairausvakuutuslakiin (364/1963) perustuva oikeus käsitellä reseptejä sairausvakuutuskorvauksen päättämistä varten. Lisäksi potilas itse voi itse tarkastaa reseptitietokannassa olevat omat tietonsa. Jokaisen sähköisiä reseptejä käsittelevän laitoksen on tehtävä Kansaneläkelaitoksen kanssa sopimus reseptitietojen säilyttämisestä, ylläpidosta ja käsittelystä reseptitietokannassa. Tässä sopimuksessa voidaan määrittää myös potilaan oikeus tarkastaa omat tietonsa suoraan Kansaneläkelaitoksen rekisteristä. Teknisen toteutuksen tulee olla tietoturvallista niin, että kaikki reseptiä koskevat tiedot siirretään tietoverkossa salattuina ja muuttumattomina. Nämä asetuksen säädökset ovat voimassa joulukuussa 2005 tehdyn muutoksen mukaan 31.12.2007 asti. [771/2003]

#### **1.4 Sähköinen allekirjoitus**

Koska sähköinen resepti varmistetaan lääkärin sähköisellä allekirjoituksella ennen sen lähettämistä reseptitietokantaan, koskee sitä myös laki sähköisistä allekirjoituksista (14/2003). Sähköinen allekirjoitus on digitaalisessa muodossa oleva tieto, joka liitetään johonkin toiseen sähköiseen tietoon kirjoittajan tai lähettäjän henkilöllisyyden varmistamiseksi. Lain mukaan allekirjoitus on luotava turvallisesti käyttämällä laitetta, joka on Euroopan yhteisön standardien mukainen, suojaa allekirjoituksen väärentämiseltä, ei muuta muita allekirjoitettavia tietoja eikä allekirjoituksen tietoja voi saada selville. Suomessa viestintävirasto voi nimetä ammattitaitoisen tarkastuslaitoksen, joka arvioi voidaanko allekirjoitukset toteuttaa tietyllä laitteella. Turvallisella laitteella kirjoitettu sähköinen allekirjoitus pätee myös oikeustoimissa. Sähköisen allekirjoituksen ja yleisen varmennetoiminnan hoitaminen ja valvominen kuuluvat liikenne- ja viestintäministeriölle. Viestintävirasto voi antaa erinäisiä teknisiä määräyksiä ja suosituksia sähköisistä allekirjoituksista, jotta ne olisivat mahdollisimman varmoja ja tietoturvallisia. [14/2003]

## 1.5 Lääkkeiden korvaaminen

Suomessa sairaanhoitoa ja lääkärin määräämiä lääkkeitä korvataan sairausvakuutuslain (1224/2004) mukaan. Lain mukaan korvataan sairauden hoidosta, lyhytaikaisesta työkyvyttömyydestä sekä raskaudesta ja lapsen hoidosta aiheutuneita kustannuksia, joihin ovat oikeutettuja Suomessa asuvat henkilöt. Sairausvakuutuslain toimeenpanija ja valvoja on Kansaneläkelaitos. Sairausvakuutuskorvauksia voi saada lääkärin suorittamista ja/tai määräämistä tutkimuksista ja hoidoista, lääkärin määräämistä lääkkeistä ja sairauden hoitoon liittyvistä matkakustannuksista sekä soveltuvin osin raskauden ja synnyttämisen aiheuttamista kustannuksista. Lääkkeiden ostosta aiheutuneita kustannuksia korvataan, mikäli lääke on lääkärin määräämä sairauden hoitoon ja se on lääkelain mukainen valmiste. Lisäksi voidaan korvata kustannuksia kliinisistä ravintovalmisteista, mikäli ne ovat lääkärin määräämiä ja niitä tarvitaan osana sairauden hoitoa. [1224/2004]

Kansaneläkelaitos myöntää kaikille lain mukaan vakuutetuille Suomen asukkaille sairausvakuutuskortin, josta näkyvät suku- ja etunimet sekä henkilötunnus. Sairausvakuutuskorttiin merkitään mahdolliset työpaikkakassan jäsenyydet, oikeudet erityiskorvaavuuksiin, sairausvakuutuksen voimassaoloaika ja suorakorvausmenettelyoikeuden puuttuminen. Apteekin toimittaessa lääkkeitä potilaalle voidaan suorittaa suorakorvausmenettely, mikä tarkoittaa, että apteekki veloittaa asiakkaalta lääkkeestä sairausvakuutuskortin perusteella alennetun hinnan ja perii sairausvakuutuslain mukaisen korvauksen Kansaneläkelaitokselta. Myös sähköisessä reseptissä tulee olla ominaisuus, jolla apteekit voivat myöntää korvauksen lääkkeestä suoraan. Sairausvakuutuskortti tulee siis olla apteekissa mukana aina, huolimatta siitä onko resepti paperinen vai sähköinen. [1224/2004]

## 2 PILOTOINTI

Suomessa sähköistä reseptiä on kokeiltu paikallisina projekteina jo muutamaan otteeseen 1980-luvun lopulta lähtien. Kansaneläkelaitos suoritti toimikorttiin pohjautuvia kokeiluja, joista yksi osa käsitteli reseptejä. Kokeilu toteutettiin vuosina 1989-1993 ja tällöin 5000 omalääkärikokeilussa toimineelle henkilölle annettiin henkilökohtainen toimikortti. Kokeilu ei kuitenkaan toiminut odotetulla tavalla, ja reseptinkäsittelyn osalta ongelmia huomattiin muun muassa reseptinkirjoitusohjelmissa, työasemien suorituskyvyissä ja varusohjelmissa. Finnet-yhtiöt alkoivat kehittää 1990-luvun alussa omaa TROPPI-hankettaan, jonka tavoitteena oli luoda paperisen reseptin ohelle sähköinen resepti. Samoihin aikoihin Mediweb ja Yliopiston Apteekki alkoivat suunnitella tietokantaan perustuvaa sähköistä reseptiä, josta lääkitystietoja voitaisiin kerätä tilastollisiin tarkoituksiin eräänlaisen tietoleikkurin avulla. Kehitys eteni hyvin; ensimmäistä prototyyppiä alettiin kehittää vuonna 1995 ja vuonna 1996 Orion ja Suomen Lääkedata Oy liittyivät mukaan kehitystyöhön. TROPPI-hanke haki TEKES:ltä rahoitusta omaa sähköisen reseptin hankettaan varten, mutta edellytyksenä oli toimiminen Mediwebin kanssa yhteistyössä eikä heillä lopulta ollut tarpeeksi resursseja yhteistoimintaan. [Hyppönen 2005 s. 20-21; Koponen-Piiroinen et al. 2001 s. 19-20]

TROPPI kariutui vuoden 1997 lopulla, mutta Mediweb aloitti ensimmäisen pilottinsa vuonna 1996 Helsingissä ja jatkoi sitä toisella pilotilla ensin Kauppatorin apteekin ja Aleksin lääkäriaseman välillä sekä myöhemmin vuonna 1998 Koillismaan neljän kunnan alueella sekä Oulussa. Pilottien aikaan myös Pharma Point Oy oli kehittänyt oman sähköisen reseptin järjestelmän, mutta senkin lupaava alku kariutui rahoituksen puutteeseen TEKES:n vaatiessa Pharma Pointilta samaa kuin TROPPI-hankkeelta. Vuonna 1995 sosiaali- ja terveysministeriö julkaisi määräyksen lääkkeiden määräämisestä ja Lääkelaitos määräyksen lääkkeiden jakelusta. Nämä yhdessä mahdollistivat myös sähköisen reseptin toiminnan ja sosiaali- ja terveysministeriö, kauppa- ja teollisuusministeriö, Tekes, kansaneläkelaitos, Työterveyslaitos, Stakes sekä Suomen Kuntaliitto aloittivatkin terveystalveta parantavan tietoteknologian tutkimisen vuonna 1998. Mediwebin ELRES-

## Liite 2

hanke pääsi mukaan tähän Makropilotiksi nimettyyn kokeiluun, joka ei kuitenkaan koskaan ehtinyt pilotointiin asti ja lopetettiin vuonna 2000. Kaikista hankkeista on niiden lopputuloksista huolimatta saatu paljon kirjallista materiaalia, jota voidaan hyödyntää muissa vastaavissa projekteissa ja sähköisen reseptin kehittämisessä. [Hyppönen 2005 s. 20-21; Koponen-Piironen et al. 2001 s. 19-20]

Erinäisten sähköisten reseptien hankkeita hidastutti ja vaikeutti kansallisten standardien ja säädösten puute lähinnä vaadittavien tekniikoiden osalta. Lisäksi useat eri järjestelmät ja niiden yhteensopimattomuus sekä teknisen tietoturvan kysymykset hankaloittivat kehitystä. Vuonna 2001 sosiaali- ja terveysministeriö aloitti toimintamalleja selkeyttävän tutkimuksen yhdessä Kansaneläkelaitoksen, lääkelaityksen, ja Suomen Apteekkariliiton kanssa. Sosiaali- ja terveysministeriön esityksestä Kansaneläkelaitos käynnisti syksyllä 2001 projektin, jonka tavoitteena oli selvittää sähköisen lääkemääräyksen vaatimuksia lähinnä tekniseltä puolelta ja rajoituksia pääsääntöisesti lain puolelta. Marraskuussa 2001 valmistui projektiryhmien laatima muistio, joka annettiin Kansaneläkelaitokselle. Muistiossa kansallisen sähköisen reseptin tavoitteiksi annettiin muun muassa tehokas ja turvallinen valtakunnallinen ratkaisumalli, standardien- ja säädöstenmukaisuus, eri toimijoiden huomiointi toteutuksessa, potilasturvallisuuden parantaminen, toiminnan nopeutuminen, helpottuminen ja laadun parantuminen sekä reseptiväärennösten ehkäisy. [Hyppönen 2005 s. 21-22; Koponen-Piironen et al. 2001 s. 1-12]

Projektin oli määrä saada aikaan toimiva keskitetty reseptijärjestelmä, joka olisi käytössä vuoden 2003 loppuun mennessä. Kesällä 2002 Helsingin ja Uudenmaan sairaanhoitopiiri, Kymenlaakson sairaanhoitopiiri ja Turun kaupungin terveystoimi saivat sosiaali- ja terveysministeriön kutsun liittyä mukaan sähköisen reseptin pilotointiin. Kaikki kolme organisaatiota vastasivat kyllä. Puolisen vuotta kutsujen jälkeen myös Pohjois-Karjalan sairaanhoitopiiri liittyi mukaan pilotointiin. Pilotissa pyrittiin saamaan mukaan keskeisimmät potilastietojärjestelmät ja apteekkien järjestelmät sekä perusterveydenhuolto ja mahdollisuuksien mukaan erikoissairaanhoito ja tarkoituksena oli aloittaa tekninen kokeilu vuoden 2003 alussa. Osana pilotointia edellytettiin, että jokainen osapuoli vastaa

omista kustannuksistaan. Marraskuussa 2002 pilotin ohjausryhmä kokoontui yhdessä sosiaali- ja terveysministeriön valitseman projektipäällikön kanssa. Projektipäällikkö oli ainoa, jolle projektissa maksettiin palkkaa ja hänen tehtävänä oli helpottaa työtä eri projektialueilla. Muita kiinteitä työryhmiä ei koottu vaan jokainen osa-alue toimi osana pilottiverkkoa, jossa jokainen vastasi omasta toiminnastaan ja kustannuksistaan. Tämä ei välttämättä ollut hyvä ajatus, ja toimijat toivoivatkin yhteistä organisointia ja selkeää suunnitelmaa. [Hyppönen 2005 s. 21-22; Koponen-Piironen et al. 2001 s. 1-12]

## **2.1 Helsingin ja Uudenmaan sairaanhoitopiiri**

Sosiaali- ja terveysministeriön toimesta oli vuonna 2001 aloitettu teknologian hyödyntämishanke, joka oli tuonut HUS:iin (Helsingin ja Uudenmaan sairaanhoitopiiri) UUMA-nimeä kantavan Uudenmaan aluetietojärjestelmähankkeen. UUMAn tavoitteena oli kehittää uudenlaisia tehokkaita palvelumalleja, joissa hyödynnetään uutta tekniikkaa. Sähköisen reseptin liittäminen osaksi UUMA-hanketta oli siis hyvin luontevaa. Helsingissä ei kuitenkaan vielä ollut käytössä sähköistä potilastietojärjestelmää ja kaikki laitoksen reseptit kirjoitettiin käsin. Sähköistä reseptiä valittiin kokeilemaan HYKS:n sairaala-apteekki, Auroran infektio- ja poliklinikka ja Meilahden gastroenterologian poliklinikka. Koska sairaala-apteekin reseptit eivät siirry Kansaneläkelaitokselle, sitä varten kehitettiin oma reseptitietokanta ja tämän oli tarkoitus korvata aiemmin käytössä ollut telefaxresepti. Sairaala-apteekkia ei voitu tämän takia myöskään pitää osana kansallista pilottia, vaan siihen kuuluivat vain sairaalan ulkopuolelle toimitettavat reseptit. Aurorassa ja Meilahdessa oli käytössä 53 toimikorttia, jotka jakautuivat laitosten kesken puoliksi. Pilotin järjestelmän toimittajaksi valittiin Mediweb Oy, joka toimitti HUS:lle erillisen selainpohjaisen reseptinkirjoitusjärjestelmän sähköisen potilasrekisterin puuttumisen takia. Järjestelmä saatiin toimimaan syksyllä 2004, jolloin ensimmäiset reseptit toimitettiin sairaala-apteekin ulkopuolelle. Vuoden 2004 loppuun mennessä noin 80 reseptiä oli toimitettu neljään kokeilussa mukana olleeseen apteekkiin ja vuoden verran toimineeseen sairaala-apteekkijärjestelmään oli toimitettu jo noin 4000 reseptiä. [Hyppönen 2005 s. 27-28]

HUS:n järjestelmä toimii ensin tunnistamalla lääkärin tämän toimikortin avulla. Reseptiä lähetettäessä HUS:n palvelin ottaa internetin käyttämällä salatun yhteyden Kansaneläkelaitoksen palvelimelle, jolloin ne tunnistavat toisensa palvelinvarmenteiden avulla. Reseptiohjelma lähettää allekirjoitetun reseptin reseptitietokantaan, missä palvelin tunnistaa sen käyttäjän sähköisestä allekirjoituksesta. Toistaiseksi lääkärit eivät kuitenkaan näe apteekkien toimitusmerkintöjä ja reseptitietokanta on avoin kaikille TEO:n myöntämän varmenteen omistajille. [Hyppönen 2006 s. 109]

## **2.2 Kymenlaakson sairaanhoitopiiri**

Kymenlaakson sairaanhoitopiirissä oli käynnistymässä Kaakkois-Suomen yhteinen KAAPPO-projekti vuoden 2002 lopulla. Projektin yhteydessä sairaanhoitopiirille oli tulossa TietoEnatorin toimittama Effica-potilastietojärjestelmä, jonka osaksi sähköistä reseptiä kaavailtiin. Ensimmäisenä Effican sai käyttöönsä kotkan kaupungin omistama Kymijoen työterveys, ja tätä ehdotettiin samalla ensimmäiseksi sähköisen reseptin pilotoijaksi. Uuden Effica-järjestelmän rinnalla toimi vielä Finstar, jota käyttivät pääasiassa ne osastot, joihin Efficaa ei vielä voitu käyttää. Finstarissa olivat myös yli vuoden vanhat potilaskertomukset ja se toimi yhä päivystyskäytössä iltaisin, öisin ja viikonloppuisin. Effican versio 3.2 ei myöskään sisältänyt kattavaa lääketietokantaa, vaan jokaisen lääkärin tuli päivittää tietoja lääkkeistä ja mahdollisia reseptipohjia. Kymijoen työterveyden käytössä oli yhteensä yhdeksän toimikorttia, joista kolme oli Karhulan työterveysasemalla, missä pilotoinnin oli tarkoitus alkaa. Loppuvuodesta 2004 sähköisen reseptin järjestelmää testattiin Kymijoen työterveydessä ja sen oli tavoitteena aloittaa toimintansa tammikuussa 2005 osana Effica-järjestelmää. Vielä toukokuussa 2005 Kymenlaakson pilotti oli jäissä, sillä se odotti kokeilussa mukana olevien apteekkien saavan integroidun lääkkeiden toimitusjärjestelmän. [Hyppönen 2005 s.28-29, 81-82]

### **2.3 Turun terveystoimi**

Turussa terveystoimen ja kaupunginsairaalan käytössä oli NovoGoup Oyj:n tuottama Pegasos-järjestelmä. Lisäksi Varsinais-Suomen sairaanhoitopiirissä oli käynnissä sähköisen asioinnin WELLCOM-hanke, jossa oltiin juuri ottamassa käyttöön uutta sähköistä potilaskertomusta. Sähköisen reseptin pilotointi liitettiin osaksi tätä hanketta ja Novo Group ilmoitti voivansa valmistaa sähköisen reseptin osaksi Pegasos-järjestelmää. Pegasos sisälsi lisäksi Suomen Apteekkariliiton lääkevalmisteiden tiedostoon perustuvan lääketietokannan, jota päivitettiin kaksi kertaa kuukaudessa. Pilotti toteutettiin pääsääntöisesti Kirkkotien terveysasemalle, jossa oli käytössä Kotkan tavoin yhdeksän toimikorttia. Vuoden 2004 lopulla järjestelmä saatiin kokeiltavaksi Turun terveystoimelle ja sen oli tarkoitus aloittaa toimintansa vuoden 2005 alussa. [Hyppönen 2005 s.29]

### **2.4 Joensuu**

Pohjois-Karjalan sairaanhoitopiiristä Joensuu ilmoitti halukkuutensa olla mukana sähköisen reseptin pilotoinnissa, sillä pilottikutsuissa oli luvattu kaikille halukkaille mahdollisuus osallistua toimintaan. Joensuussa sijaitseva Pohjois-Karjalan keskussairaala liittyi mukaan pilottiin puolisen vuotta muiden kutsuttujen jälkeen ja siellä kokeiluun mukaan tulivat silmätautien, naistentautien sekä kurkku-, nenä- ja korvatautien osastot. Pohjois-Karjalan sairaanhoitopiirissä oli käytössä Doctorexin toimittama HealthNet-järjestelmä ja keskussairaalassa SecGo:n toimittamat ja varmentamat toimikortit, joita oli pilotissa olleilla osastoilla yhteensä 22 kappaletta. Vuoden 2004 alussa HealthNetiin yhdistetty Pharmapoint-lääketietokanta oli ollut käytössä vain hetken aikaa, ja useat lääkärit kirjoittivat siitä huolimatta reseptinsä käsin. Joensuussa toiminta oli kuitenkin reipasta ja sähköisten reseptien käyttö aloitettiin toukokuussa 2004. Saman vuoden joulukuussa sähköisiä reseptejä oli kirjoitettu jo 254 kappaletta. Joensuussa järjestelmä toimii lähettämällä henkilökohtaisesti allekirjoitetut reseptit ensin paikalliselle allekirjoituspalvelimelle, joka lähettää ne edelleen Kansaneläkelaitoksen reseptitietokantaan. Potilaan suostumus pyydetään tulostettavaan lomakkeeseen ja reseptin tiedot talletetaan myös HealthNet-järjestelmään. [Hyppönen 2005 s.29, 63]

## **2.5 Mediweb Oy**

Vantaalainen vuonna 1996 perustettu Mediweb Oy oli ollut mukana sähköisen reseptin kehittämisessä jo aikaisemmin muun muassa sosiaali- ja terveysministeriön Makropilotissa. Kilpailuttamisen jälkeen Mediweb valittiinkin toteuttamaan sähköisen reseptin natiivijärjestelmä ja se vastasi reseptitietokannan ylläpidosta, sanomien käsittelystä sekä lääkäreiden ja apteekkien tilapäisistä selainohjelmista. Natiivijärjestelmä on selainpohjainen potilastietojärjestelmistä täysin erillinen sovellus, jolla voidaan kirjoittaa ja lähettää reseptejä reseptitietokantaan. Natiivijärjestelmän kanssa työasemalla tarvitsee olla vain sähköiseen allekirjoitukseen vaadittavat työkalut, selainsovellus hoitaa loput. Sähköisen reseptin käyttäjille suunnattu ohjelmisto toimii Windows-pohjaisena Explorer-selaimen avulla ja verkon välityksellä käytävään kommunikointiin käytetään SSL-protokollaa. Ongelmana on, että ohjelmisto sallii reseptien katselun kaikille lääkäreille, koko tietylle organisaatiolle tai kaikille apteekkeille. Tämä on täysin potilaslain, pilotin vaatiman potilaan suostumuksen ja sairaaloiden politiikan vastaista, ja se tulisi korjata viimeistään ennen ohjelmiston levittämistä. [ElRes 2006; Hyppönen 2005 s.29]

## **2.6 Apteekit**

Sosiaali- ja terveysministeriö päätti kokeiluun osallistuvista terveydenhuolto-organisaatioista, joten apteekkipuolen toimijoiden kokous päätti antaa Suomen Apteekkariliiton valita kokeiluun mukaan tulevat yksityiset apteekit niiltä paikkakunnilta, joissa kokeilua toimitettiin. Paikkakunnilta valittiin mahdollisimman monta apteekkia, joilla oli halukkuutta ja tietotekniset valmiudet osallistumiseen. Yksityisten apteekkien yhtenä valintakriteerinä oli niiden kuuluminen apteekkien yhteiseen SAL Apteekki verkkoon, jotta tarvittava tietoturva voitiin taata. Lisäksi Yliopiston apteekki osallistui mukaan voimavarojensa mukaan niillä paikkakunnilla, joilla se toimii. SAL Apteekki verkossa sähköiset reseptit voidaan kuljettaa apteekin ja Kansaneläkelaitoksen välillä vahvasti salattuina ja järjestelmään kirjaudutaan terveydenhuollon



## Liite 2

ammattihenkilöille myönnettävällä kortilla. Apteekkiverkkoa valvotaan etänä palomuurien avulla ja tiedonsiirrossa käytetään VPN:ää (Virtual Private Network) julkisen Internetin päällä. Alun perin kaikki apteekkien pilotit oli tarkoitus toteuttaa selainpohjaisella Mediwebin tuottamalla natiiviohjelmalla. Jo vuoden 2004 lopulla Kymenlaakson ja Turun apteekit yhdessä terveydenhuolto-organisaatioiden kanssa halusivat odottaa omiin järjestelmiinsä integroitua reseptin kirjoitus- ja toimitusohjelmia erinäisten järjestelmävaikeuksien takia. Tekniset ongelmat apteekeissa ja muualla kokeilussa hidastuttivatkin pilotin etenemistä siinä määrin, että vuoden 2005 aikana apteekeista toimitettiin vain noin 400 sähköistä reseptiä. [Hyppönen 2005 s.30, 81-82; Suomen Apteekkariliitto 2004 s. 6-7; Suomen Apteekkariliitto 2005 s. 10]

Apteekkariliitto oli valinnut jokaiselta pilottialueelta mukaan yhden apteekin, ja Helsingin alueelta kolme. Vaatimuksena tosiaan oli apteekkien kuuluminen liiton SAL- apteekkiverkkoon. Helsingissä huomattiin vuoden 2004 lopulla, että jopa 94 prosenttia potentiaalisista potilaista kieltäytyi sähköisestä reseptistä, sillä heidän suosimansa apteekit eivät kuuluneen kokeilun piiriin. Turussa pelättiin samaa ongelmaa, vaikka siellä oli odotettu muita paikkakuntia suurempia reseptimääriä. Jotta Turun odotettu volyyymi saataisiin saavutettua, ongelman ratkaisemiseksi terveysaseman lähiapteekki liitettiin mukaan apteekkiverkkoon ja sitä kautta pilottiin. [Hyppönen 2006 s 87-88]

Tällä hetkellä apteekkien järjestelmien toiminnoissa on vielä eroja. Yliopiston apteekissa jokainen reseptitietokannan käyttäjä voi tunnistautua suoraan Kansaneläkelaitoksen palvelimelle toimikortilla ja liikenne palvelimien välille kulkee salattuna. Yksityisissä apteekeissa yhteys Kansaneläkelaitoksen palvelimelle muodostetaan SSL:n (secure Sockets Layer) avulla ja käyttäjät tunnistetaan Yliopiston apteekin tavoin toimikortilla. Reseptiohjelmisto muodostaa suojatun yhteyden paikallisen verkon yli, jolloin järjestelmä on suojattu sekä apteekin oman järjestelmän, reseptiohjelmiston sekä Kansaneläkelaitoksen puolesta. Kaikki lääkkeiden toimitusmerkinnät allekirjoitetaan apteekissa ja vaikka kaikki toimikortin haltijat pääsevät reseptejä katsomaan, niitä voidaan korjata vain allekirjoituksella. [Hyppönen 2006 s. 110]

## **2.7 Kokemuksia sähköisestä reseptistä**

Suomessa sähköisen reseptin projektin viimeisin kokoontuminen oli toukokuun 22. päivä, jolloin osapuolet arvioivat pilotointia ja projektin etenemistä sekä tekivät suunnitelmia tulevaan. Kokoontumisessa arvioitiin muun muassa sähköisen reseptin käyttöä, potilaiden suhtautumista, tietoturvaa, oikeudellisia seikkoja, taloudellisia seikkoja, kansainvälistä sijoittumista sekä hankkeen jatkotyötä. Tällä hetkellä (kesä 2006) sähköinen resepti toimii miltei vuonna 2001 tehdyn esiselostuksen kuvaamalla tavalla. Potilas menee lääkärin vastaanotolle ja antaa lääkärille suostumuksensa sähköisen reseptin käyttöön. Lääkäri kirjoittaa potilaalle sähköisen reseptin käyttäen hyväkseen lääketietokantaa ja potilaan aikaisempia reseptejä. Sähköinen resepti lähetetään salattuna ja lääkärin sähköisellä allekirjoituksella varustettuna kansaneläkelaitoksen ylläpitämään reseptitietokantaan. Potilaan noutaessa lääkettä apteekista farmaseutti tai proviisori tunnistautuu järjestelmään ja noutaa reseptin tietokannasta lääkkeen toimittamista varten. Apteekki myöntää potilaalle tälle kuuluvan suorakorvauksen ja informoi asiasta kansaneläkelaitosta, joka tarkastaa reseptin ja lääkkeen noudon reseptitietokannasta. [Hyppönen 2006]

Arvioinnin tavoitteena oli ennakoida vaikutuksia järjestelmän suunnitteluvaiheessa vuonna 2001, arvioida rakentamista ja sen vaikutuksia järjestelmän kehitysvaiheessa vuoden 2002 aikana, arvioida käyttöönottoa pilotoinnissa vuosina 2003-2005 sekä vuodesta 2006 eteenpäin arvioida järjestelmän leviämistä täyttöön ja sen vakiintumista eri terveydenhuollon organisaatioihin. Käyttötesteissä huomattiin järjestelmässä vielä olevan puutteita ja kehitettävää. Esimerkiksi apteekin käyttöliittymän huomattiin lääkkeen toimituksen jälkeen muutama otteeseen vaihtaneen lääkkeen pakkauskoko, eikä järjestelmä ollut varoittanut asiasta lainkaan. Tällainen on tietenkin korjattava, sillä pakkauskoon täytyy pysyä reseptissä samana koko prosessin ajan. Huomattiin myös, että sähköisen käyttäjät tarvitsevat enemmän koulutusmahdollisuuksia, sillä kaikki eivät sitä hallitse parhaalla mahdollisella tavalla. Ennen sähköisen reseptijärjestelmän leviämistä laajempaan käyttöön, on syytä vielä suorittaa mittavia käyttötestejä, joissa järjestelmää kokeillaan kaikissa mahdollisissa käyttöympäristöissä ja kaikissa sen mahdollistamissa toiminnoissa. Vanhojen ongelmien, kuten epäselvien reseptien poistuttua, voi sähköinen

## Liite 2

resepti aiheuttaa uudenlaisia virhemahdollisuuksia, joihin on syytä varautua. Pilotointi on kuitenkin osaltaan onnistunut varsin hyvin, sillä se on tuonut esille tarvittavia kehityskohteita, joita ilman järjestelmän pilottia ei välttämättä olisi havaittu ennen kansallista käyttöä. Suomessa sähköisen reseptin kokeiluasetusta on jatkettu vuoden 2007 loppuun asti, joten virheiden korjaamiseen ja järjestelmän kehittämiseen on hyvin aikaa. [Hyppönen 2006]

Pilotin hidastumista voidaan selittää monellakin tavalla. Suurimmiksi osasyiksi muodostuvat organisoinnin puutos ja projektin vaatimusten jatkuva muuttuminen. On ilmeistä, että kansallinen projekti vaatii yhteisen toimintasuunnitelman. Tällaista ei esiselvitystä lukuun ottamatta kuitenkaan ollut, vaan koko projekti hoidettiin eräänlaisena verkottuneena toimintona, jossa jokainen osa-alue vastasi omista toimistaan ja kustannuksistaan. Projektille oli palkattu vain yksi työntekijä eli projektipäällikkö, joka ei tietenkään voinut olla useassa paikassa samaan aikaan. //KPLJAKO// Järjestelmävaatimukset ja muut järjestelmän tekniset piirteet muuttuivat useaan otteeseen projektin aikana, mikä puolestaan aiheutti järjestelmien toimittamisen viivästymisiä ja käyttöönoton hidastumista. Esimerkiksi projektin alussa pilotin kaikkien toimijoiden oli tarkoitus aloittaa erillisellä selainohjelmalla. Selainohjelmassa oli kuitenkin paljon puutteita ja sen toiminta yhdessä muiden järjestelmien kanssa oli kankeaa. Vuoden 2003 alussa päädyttiinkin ratkaisuun, jossa suositettiin eri organisaatioiden järjestelmiin integroitavaa reseptinkirjoitus- tai toimitusohjelmaa. Tämä sai eri laitokset odottamaan integroitujen järjestelmien valmistumista ja aiheutti järjestelmätoimittajille muutoksia sopimuksiin ja jo valmiina oleviin osiin, mikä puolestaan jarrutti koko pilottia. Samalla tavalla muun muassa reseptien allekirjoittaminen tuotti päänvaivaa ja viivästyksiä eri osapuolille. Alkuvaateiden mukaan järjestelmien piti pystyä allekirjoittamaan vain yksi resepti kerrallaan, mutta vuoden 2004 lopulla siirryttiin vaatimaan mahdollisuutta usean reseptin allekirjoittamiseen samanaikaisesti. [Hyppönen 2006 s.80-82] //JATKA TÄHÄN YHTEISIÄ ASIOITA//

Odotukset sähköisen reseptin toiminnasta olivat sen parissa työskentelevien osalta suhteellisen alhaiset. Miltei kaikki odottivat sen tuovan muutoksia työkuvaan, mutta vain

## Liite 2

pieni osa oli sitä mieltä, että järjestelmästä saataisiin suurta hyötyä kaikille. Kaikki olivat kuitenkin mukana projektissa reippaina ja odottivat sen tehostavan toimintaa ja parantavan palvelun laatua. Esimerkiksi kun reseptejä ei enää kirjoiteta käsin, eivätkä ne voi vioittua matkalla, virheiden määrä alenee ja potilaiden turvallisuus paranee. Sähköisen reseptin ohjausryhmä suoritti haastatteluja eri toimijoiden parissa saadakseen selville miten sähköiseen reseptiin suhtaudutaan ja miten se on toiminut. Eniten kritiikkiä saivat muuttuneet vaatimukset ja organisoinnin puute, jotka vääjäämättä johtivat koko projektin aikataulun venymiseen. Lisäksi tiedonkulun toivottiin olevan parempaa kaikilla osapuolilla. Lääkärrien mielestä vanhan reseptinkirjoitusjärjestelmän, toisin sanoen atk-reseptin tai käsinkirjoitetun lomakkeen, suurin haitta on ollut sen hitaus. He eivät kuitenkaan usko sähköisen reseptin tuovan paljoakaan säästöä aikaan ja vaivaan, ainakaan alussa. Haastatellut lääkärit totesivat, että he odottavat sähköisen reseptin tuovan parannuksia pääsääntöisesti luotettavuuteen sekä vähentävän mahdollisten virheiden ja väärinkäytösten määrää. Järjestelmän yleistyminen on viime kädessä kiinni lääkäreistä. Jos he eivät käytä järjestelmää, sähköisiä reseptejä ei mene reseptitietokantaan eikä sieltä apteekkiin. [Hyppönen 2006 s. 73-74, 93-96]

Aptekeissa sähköisen reseptin toivotaan tuovan helpotusta epäselviin resepteihin, joista on aina soitettava lääkärille. Sähköisessä reseptijärjestelmässä lääkärin kirjoittama resepti näkyy apteekin järjestelmässä aivan samanlaisena, joten kaikki käsialan tulkintavirheet tai niistä aiheutuvat viiveet jäävät pois. Sähköinen resepti helpottaa myös apteekkihenkilökunnan työtä huomattavasti. Paperireseptin kaikki tiedot on kirjattava apteekin järjestelmään lääkkeen toimitusta ja Kansaneläkelaitoksen korvausta varten. Sähköisessä reseptissä resepti-informaatio tulee suoraan apteekin järjestelmään, jossa sitä voidaan käsitellä ilman erillistä reseptin käsittelyä. Apteekit uskovat järjestelmän nopeuttavan lääkkeiden toimitusta ja parantavan potilaiden turvallisuutta, vaikkakin pelkäävät myös järjestelmän opetteluun hidastuttavan toimintaa alussa. Sekä apteekkihenkilökunta että lääkärit kokevat riskinä tietoturvallisuuden puutteet ja muut mahdolliset tekniset ongelmat. Lääkärit eivät kuitenkaan ole niin toiveikkaita järjestelmän suhteen kuin apteekkihenkilökunta. [Hyppönen 2006 s. 75-76]

### **2.7.1 Potilaiden mielipiteitä**

Suomen sähköisen reseptin projekti suoritti myös mielipidekyselyn reseptiä käyttäneiden potilaiden keskuudessa lähettämällä lomakkeita eri toimintayksiköihin. Lomakkeissa kyseltiin ensin taustatietoja, kuten ikä ja sukupuoli, lääkkeiden käyttömäärä, sähköisen reseptin käyttökerrat sekä niiden mahdolliset virheet ja korjaukset. Lisäksi kyseltiin apteekissa ja terveydenhuolto-organisaatiossa asioinnista eli kokeilun tiedottamisesta, suostumuksen antamisesta, onnistumisista ja potilasohjeesta sekä mahdollisista näkemyksistä sähköisen reseptin parantamiseksi. Vaikka otos oli rajattu, vastausprosentti oli yhteensä 54. Sähköisen reseptin tiedottamisesta todettiin, että on tärkeää saada tietoa reseptin käytön hyödyistä ja haitoista sekä oikeuksista tarkastaa omat tiedot. Suurin osa potilaista oli sitä mieltä, että he olivat saaneet tarpeeksi tietoa reseptin eduista, mutta eivät niiden haitoista, omista oikeuksista tai tietoturvasta. Lääkemääräyksen laatiminen on silti onnistunut pääsääntöisesti hyvin. Potilaat kertoivat, että vain yhdellä (2 %) sähköisen reseptin kirjoitus ja lähettäminen eivät onnistunut, suurimmalla osalla reseptin kirjoittaminen ja lähettäminen onnistui ongelmitta ja muissakin tapauksissa vain pienten ongelmien jälkeen. [Tenhunen 2006]

Suostumuksen osalta puolet potilaista totesi haluavansa suostua sähköiseen reseptiin ilman erillistä kieltoa, mutta he halusivat rajoittaa lääkärin oikeuksia katsoa aikaisempia reseptejä ilman erillistä suostumusta. Potilasohjeisiin olivat tyytyväisiä ne, jotka sen olivat saaneet. Lääkäri oli tulostanut potilasohjeen 65 prosentille potilaista, ja loputkin olisivat sellaisen halunneet, ettei tieto lääkkeestä jäisi vain oman muistin varaan. Jonkinlainen dokumentti lääkkeen määräämisestä on siis hyvä jäädä potilaalle. Sähköisen lääkemääräyksen toimittaminen oli onnistunut kaikille potilaille, joskin hieman hitaanlaisesti apteekkihenkilökunnan opetellessa samalla järjestelmän toimintaa. Kun apteekkien henkilökunta tottuu uuteen järjestelmään, se varmasti nopeuttaa lääkemääräyksen toimittamista ja mukavoittaa näin ollen potilaidenkin oloa. Tieto noutamattomista lääkkeistä jäi apteekkeilla huomioitta, sillä suurin osa vastaajista ei tiennyt tai ei saanut minkäänlaista merkintää mahdollisista jäljellä olevista lääkkeistä. Vain pieni osa potilaista tiesi varmasti saaneensa merkinnän ja tiedon siitä, että heillä on vielä lääkkeitä noutamatta.

Järjestelmän ja käytännön yleistyessä, varmasti myös noutamattomien lääkkeiden merkinnät löytyvät jokaiselta potilaalta. [Tenhunen 2006]

Keskeisinä hyötyinä potilaiden kannalta voidaan siis nähdä reseptien uusimisen helpottuminen, niiden säilyminen tallessa, väärinkäytöksiä ja päällekkäisten lääkitysten väheneminen, turvallisuus, nopeus ja helpompi asiointi etenkin vanhuksille ja paljon matkustaville. Hyödyt tulevat täysiin oikeuksiinsa kuitenkin vasta kun järjestelmän käyttö yleistyy ja siinä ilmenneet viat saadaan korjattua. Varsinkin nopeampi reseptin kirjoittaminen ja toimittaminen saavutetaan kun sitä käyttävät terveydenhuollon ammattilaiset ovat tottuneet järjestelmään ja osaavat käyttää sitä. Vielä nyt sähköisen reseptin haittoja ovatkin riittämättömät tietoturva ja tietosuoja-ominaisuudet, paikoittainen hitaus, järjestelmän pieni käyttöalue ja potilasohjeiden puuttuminen tai unohtuminen. Nämä ovat korjattavissa olevia puutteita, joihin varmasti kiinnitetään huomiota ennen kansallista levitystä. [Tenhunen 2006]

### **2.7.2 Tietoturva**

Suomen sähköisen reseptin projektissa on tavoitteena ollut, että järjestelmään voidaan luoda niin täydellinen tietoturva, että se vakuuttaa kaikki reseptin käyttäjät järjestelmän ylläpitäjistä potilaisiin. On tärkeää, että viestit pysyvät muuttumattomina ja luottamuksellisina, ne ovat tarvittaessa saatavilla viiveettä, käsittelystä voidaan pitää kirjanpitoa ja jokainen tapahtuma on jäljitettävissä. On myös yhtä oleellista, ettei yksikään resepti pääse ulkopuolisten käsiin ja potilaille sekä muille käyttäjille ei koidu järjestelmän käytöstä harmia. Yhtäläillä järjestelmän on toimiakseen oltava standardien mukainen, ja noudatettava Suomen lainsäädäntöä sekä muita määräyksiä. Lisäksi väärinkäytöksiä on pyrittävä estämään, mikä toteutetaan lähinnä käyttäjien autentikoinnilla ja erilaisilla käyttöoikeuksilla. [Ruotsalainen 2006]

Pilottijärjestelmässä sekä tulevassa kansallisessa järjestelmässä luottamuksellisuus pyritään takaamaan vaatimalla asiayhteys reseptien tarkasteluun, pyytämällä potilaan suostumus reseptin käyttöön ja käsittelyyn sekä estämällä reseptien kulkeutuminen ulkopuolisille.

Reseptijärjestelmää käyttävät tunnistetaan PKI-järjestelmän (Public Key Infrastructure) ja sirukortin avulla, ja tieto liikkuu verkossa joko VPN:n (Virtual Private Network) tai SSL:n (Secure Sockets Layer) avulla. Käyttöoikeuksien hallinnassa on silti huomattu puutteita, sillä jokainen järjestelmään pääsevä pääsee myös toisten rekisterinpitäjien tietoihin käsiksi, asiayhteyttä ei tarkasteta ja potilaat eivät pääse katsomaan reseptitietokannan lokia, eli sitä ketkä ovat heidän tietojaan katsoneet. Lisäksi pilotin aikana ei toteutettu potilaan mahdollisuutta suojata oma reseptinsä henkilökohtaisella PIN-koodilla. Tulevaisuudessa olisikin yksityisyyden suojan kannalta oleellista mahdollistaa potilaalle oikeus päättää siitä, ketkä hänen tietojaan voivat katsella ja nähdä ketkä niitä ovat jo katselleet. Koko projekti tarvitsee siis kunnollisen riskianalyysin ja tietoturvalitiikan, ja jokaisen hankkeen osan tulee olla standardien, lakien ja määräysten mukainen ennen kansallista levitystä, muuten potilaan oikeuksia saatetaan loukata. [Ruotsalainen 2006]

### **2.7.3 Oikeudellinen asema**

Suomessa sähköisen reseptin kanssa asioitaessa potilaalla on aina oikeus tiedonsaantiin. Potilas- ja henkilötietolakien mukaan potilas saa itse päättää hoidostaan, joten hänelle on annettava tarvittava määrä tietoa hänen omasta terveydentilastaan ja eri hoitomahdollisuuksista. Potilaan on myös saatava tarvittava määrä tietoa sähköisen reseptin käytöstä, tietojen luovutuksesta ja käyttötarkoituksesta, jotta hänen suostumuksensa on vapaaehtoinen. Pilotoinnin aikana tietyissä tiedonvälitystapahtumissa on huomattu puutteita. Potilaalle on ylimalkaisesti kerrottu sähköisen reseptin periaate, mutta ei voida olla varmoja, että jokainen on ymmärtänyt mihin kaikkeen hän suostumuksensa antaa. Varsinkin potilasohjetta potilaat pitivät tärkeänä, mutta valitettavan usein lääkärit eivät tätä potilaalle tulostaneet. Syynä voi olla se, että kokeiluasetuksen mukaan potilasohje ei ole pakollinen, mutta tällöin potilaille olisi pitänyt ilmoittaa, että potilasohjeen voi myös erikseen pyytää, jolloin sen saa varmasti. Myös huomion kiinnittäminen tietoturva-asioihin on jäänyt taka-alalle, aivan kuin erillisen suostumuksen tarpeesta vanhempien reseptien katsomista varten. Potilaita on kuitenkin tarpeeksi informoitu mahdollisuudesta peruuttaa suostumus milloin tahansa. [Pajukoski 2006]

### **3 SÄHKÖINEN RESEPTI MAAILMALLA**

Suomen lisäksi useat muut maat ovat kokeilleet sähköistä reseptiä ja sen erityyppisiä järjestelmiä. Suomen keskitetyn tietokantamallin lisäksi sähköinen resepti voidaan toteuttaa useallakin eri tavalla. Englannissa reseptit lähetetään suoraan potilaan haluamaan apteekkiin ja ne tunnistetaan viivakoodien avulla, Ruotsissa järjestelmä toimii kansallisessa terveydenhuollon verkossa keskitetyn postilaatikon avulla, Tanskassa käytetään apuna EDI:ä ja Saksassa on käytössä toimikorttipohjainen ratkaisu. Ja tässä on vasta muutama mahdollinen vaihtoehto. Alla on kuvattu eri maiden sähköisten reseptien toimintaa ja niiden kehittämisen etenemistä.

#### **3.1 Englanti**

Englannissa on EPS-järjestelmää (Electronic Prescription System) testattu muutamaan otteeseen vuodesta 1997 lähtien. Kokeilussa on ollut kolme eri periaatteella toimivaa järjestelmää: Transcript, Pharmacy2U ja Flexiscript. Transcript-järjestelmässä potilaalle annetaan paperilappunen, jossa on kaksi viivakoodia ja potilas voi mennä mihin tahansa haluamaansa apteekkiin lääkettä noutamaan. Farmaseutti lukee viivakoodeista reseptin tiedot ja ne validoivan sähköisen allekirjoituksen ja toimittaa niiden perusteella lääkkeen potilaalle. Pharmacy2U-mallissa potilas voi mennä vain tiettyyn, ennalta määrättyyn apteekkiin noutamaan lääkettä. Lääkäri kirjoittaa potilaalle reseptin, jonka hän allekirjoittaa, salaa ja lähettää potilaan kanssa sovittuun apteekkiin. Potilas menee kyseiseen apteekkiin, missä farmaseutti tarkastaa reseptin ja toimittaa lääkkeen potilaalle. Flexiscript-malli toimii Suomen sähköisen reseptin tavoin. Siinä lääkäri kirjoittaa reseptin ja lähettää sen yhteiseen keskusvarastoon, josta farmaseutti voi noutaa sen potilaan tullessa hakemaan lääkettään. Näiden lisäksi Englannissa on tehty tutkimuksia myös muun muassa Menettelytapapohjaisesta reseptien sähköisestä siirrosta ja Jiniin perustuvista ratkaisuista (Jini on Sun Microsystemsien ohjelmisto, jolla yritetään yksinkertaistaa eri laitteiden välisiä yhteyksiä ja niiden jakamista verkossa).[Chadwick & Mundy 2003; Ghinea et al.] 2006]

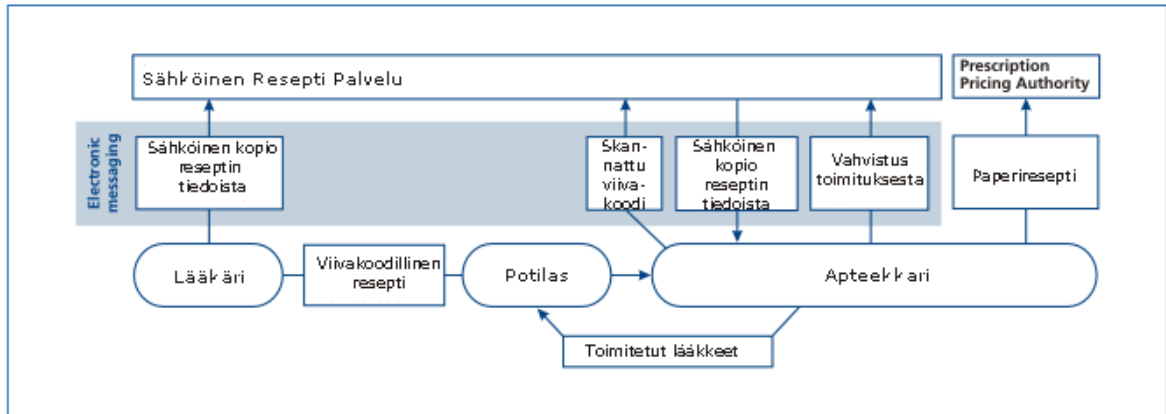


Englannin terveystieteiden ministeriö on ollut voimakkaasti mukana kehittämässä kansallista ETP-järjestelmää (Electronic Transfer of Prescriptions), sillä Englannissa kirjoitetaan päivittäin miltei 1,3 miljoonaa paperista reseptiä ja määrän odotetaan nousevan viiden prosentin vuosivauhdilla. Sähköinen resepti helpottaisi etenkin uusittavien reseptien käsittelyä, joita on noin 70 prosenttia päivittäisistä tapauksista. Osana terveystieteiden ministeri Lordi Philip Huntin aloittamaa 'Farmasia tulevaisuudessa' – ohjelmaa vuonna 2001 aloitettiin projekti, jonka avulla liitettiin kaikki tietokoneistetut lääkäriasemat paikallisiin apteekkeihin käyttämällä NHSnetiä. NHSnet on NHS:n (National Health Service), eli Englannin kansallisen terveystieteen palvelun, toimeenpanema turvallinen suuralueverkko, joka yhdistää toisiinsa eri NHS:n organisaatiot. Sen avulla sekä lääkäriasemat että apteekit ovat yhteydessä myös lääkkeiden korvauksista päättävään PPA:han (Prescription Pricing Authority) ja yhteyksien avulla voidaan helposti toteuttaa myös ETP-järjestelmää. Vuonna 2001 alkaneeseen kansalliseen ETP-pilottiin valittiin kolme konsortiota: Transcript, Pharmacy2U ja Sema. Pilottien kokemuksista pääteltiin, että ETP-järjestelmä olisi teknisesti toimiva ja toisi etuja potilaille, apteekkeille ja lääkäreille. EPS-järjestelmän (Electronic Prescription System) levitys kansalliseksi alkoi vuonna 2005 ja sen on tarkoitus olla valmis vuoden 2007 lopulla. [Department of Health 2006]

EPS-järjestelmän kansallinen levitys toteutetaan useammassa vaiheessa, jotta siirtyminen järjestelmään sujuisi ongelmitta ja tarpeelliset lääkkeet saataisiin toimitettua järjestelmämuutoksista huolimatta. Ensimmäisessä vaiheessa sairaaloiden, lääkäriasemien, apteekkien ja muiden terveystieteen organisaatioiden laitteita aletaan päivittää EPS-järjestelmää varten. Potilaat saavat vanhan järjestelmän mukaisesti paperisen reseptin, joka muistuttaa vanhaa käytössä ollutta lomaketta. Paperiin on kuitenkin tulostettu viivakoodi, josta resepti voidaan tunnistaa apteekissa. Kun lääkäri kirjoittaa reseptin, hän tulostaa potilaalle oman paperisen viivakoodin sisältävän version ja lähettää samalla sähköisen kopion reseptistä sähköiseen reseptijärjestelmään. Apteekissa farmaseutti tunnistaa reseptin lukemalla viivakoodin ja reseptin tietojen uudelleen kirjoitukselta apteekin järjestelmään vältytään. Jos apteekissa ei vielä ole mahdollista käyttää EPS:ää, voidaan reseptin kanssa

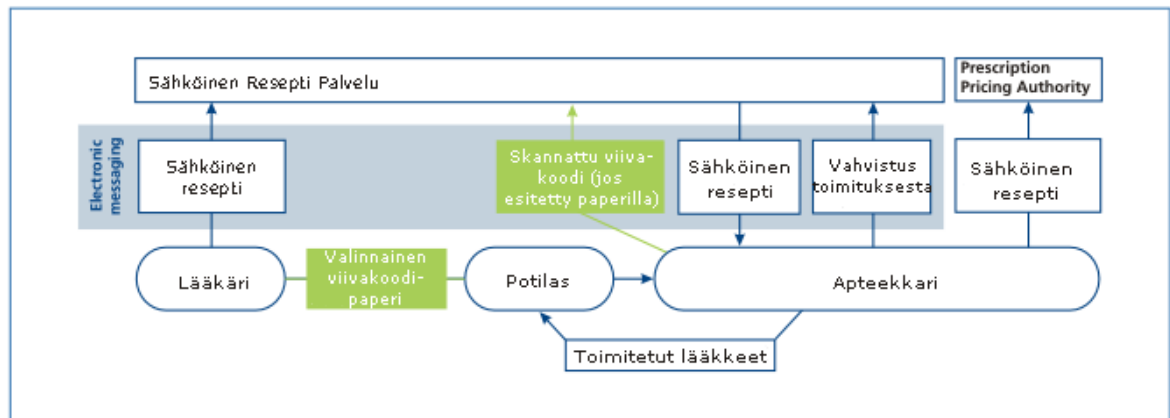
### Liite 3

menetellä aivan kuten paperireseptien kanssa aiemminkin, ja potilas saa tarvitsemansa lääkkeet. Vaiheen 1 tapahtumaketjua havainnollistetaan kuvassa 3. [NHS 2006]



Kuva 19 EPS-järjestelmän käyttöönoton vaihe 1 [NHS 2006]

Vaiheessa 2 jokaiseen sähköiseen reseptiin voidaan liittää lääkärin sähköinen allekirjoitus, jonka avulla sähköisestä reseptistä saadaan laillinen lomake, jota vastaan lääkkeitä voidaan luovuttaa. Jokaiselle reseptinkirjoitusoikeuden omaavalle henkilölle annetaan älykortti, jonka avulla sähköinen allekirjoitus tehdään, ja joiden avulla reseptien kirjoitusta voidaan valvoa. Sähköisen allekirjoituksen myötä potilaat voivat valita apteekin, josta he mieluiten ja säännöllisesti lääkkeensä hakevat ja resepti lähetetään suoraan sinne. Tällöin potilaalle ei jää reseptistä minkäänlaista kirjallista dokumenttia, elleivät he sitä halua. Jokainen halukas voi myös saada paperisen ykkösvaiheen kaltaisen reseptin, jolla lääkkeet voi noutaa apteekista, ja vaikka potilas olisi valinnut tietyn apteekin, johon reseptit normaalisti lähetetään, voidaan sitä muuttaa milloin tahansa. Vaihetta 2 havainnollistaa kuva 4. [NHS 2006]



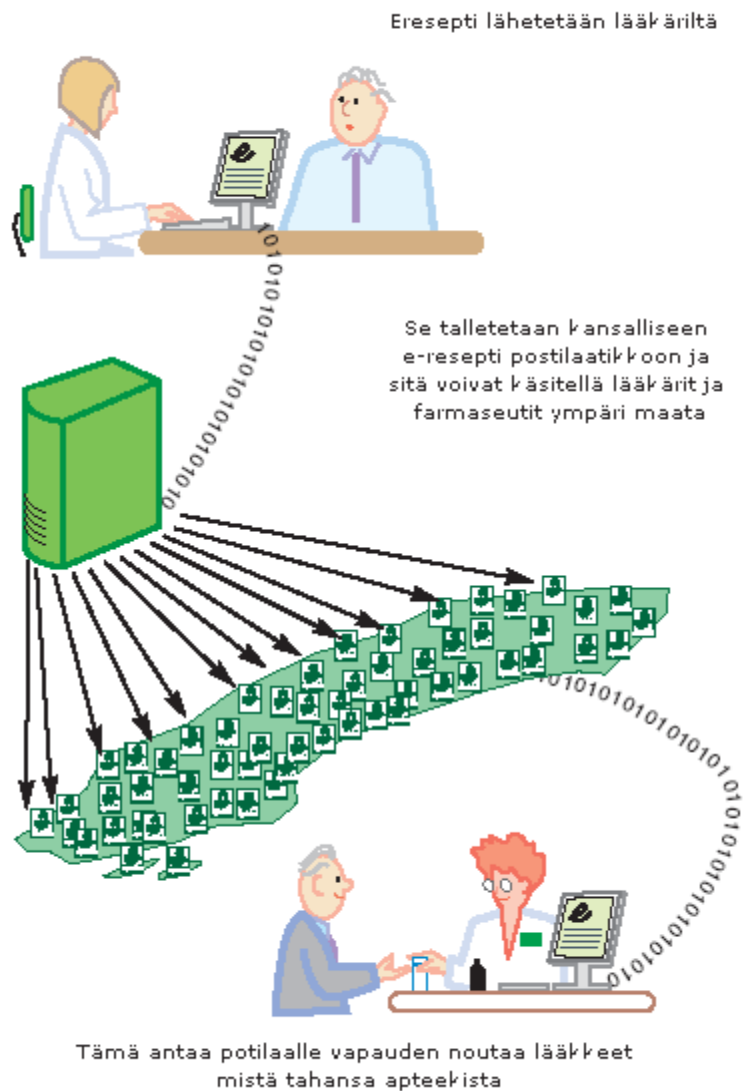
Kuva 20 EPS-järjestelmän vaihe 2 [NHS 2006]

Kolmannessa vaiheessa on tarkoitus päästä asteittain eroon paperisesta reseptistä ja siirtyä kokonaan sähköiseen reseptiin. Paperinen resepti on vielä erikoistilanteissa mahdollinen, mutta niistä pyritään luopumaan kokonaan. Myös PPA (Prescription Pricing Authority) aloittaa siirtymisen korvaavuuksien määrittämiseen sähköisiä reseptejä vastaan. Neljännessä vaiheessa on tarkoitus päästä käyttämään sähköistä reseptiä pääsääntöisenä lääkkeenmääräämismuotona. Paperisia reseptejä vastaan lääkkeitä ei enää luovuteta ilman erittäin painavaa syytä. Englannissa lainsäädäntö sallii jo sähköiseen reseptiin siirtymisen, mutta ohjeet järjestelmän toimintaan ovat vielä aluilla. Jotta NHS ja PCT (Primary Care Trust) voivat tehokkaasti käyttää EPS-järjestelmää, on terveysministeriön annettava tälle menettelyohje. [NHS 2006]

### 3.2 Ruotsi

Ruotsissa sähköinen resepti ja kansallinen potilastietoverkko ovat jo pitkällä. Vuonna 1998 Ruotsissa aloitettiin Sjunetin, eli kansallisen sairaanhoidon avuksi tarkoitetun IP-pohjaisen laajakaistaverkon kehitys seitsemän kunnan alueella. Sjunet toi sairaanhoitohenkilökunnan ulottuville uusia mahdollisuuksia kuten telelääketieteen, varmemman sähköpostin, IP-puhelut ja videoneuvottelut. Vuoden 2001 syksyyn mennessä Sjunet kattoi koko Ruotsin, minkä jälkeen myös Apoteket AB (Ruotsin ainut, valtion organisoima apteekkilaitos)

lisättiin mukaan verkkoon. Kaikki Ruotsin apteekit ovat siis samassa verkossa yhdessä lääkkeitä määräävien lääkäreiden ja sairaanhoitajien kanssa. Sjunetin avulla kaikki Ruotsin 21 kunnan omat palomuurit on kytketty toisiinsa ja kaikki kunnan verkossa toimivat organisaatiot voivat näin ollen saada yhteyden internetiin ja Sjunetiin. Verkon toiminnan alkaessa yhteys toimi virtuaalisen verkon (VPN – Virtual Private Network) avulla, mutta vuodesta 2003 lähtien on verkko rakennettu VLAN-tekniikan (Virtual Local Area Network) avulla. [Larsson 2001; Sjöborg et al 2006 s. 4]



Kuva 21 Ruotsissa käytössä oleva sähköisen reseptin järjestelmä [e-recept Stockholm 2006]

Yhteinen verkko helpotti myös sähköisen reseptin esille tuomista ja leviämistä. Ensimmäisiä järjestelmiä kehiteltiin jo vuonna 1999 ja yhteinen reseptipalvelin vuonna 2001. Vuoden 2006 huhtikuussa jo yli puolet ensimmäistä kertaa kirjoitettavista resepteistä oli sähköisiä. Ruotsissa sähköisen reseptin järjestelmä toimii niin, että reseptin määrännyt lääkäri tai sairaanhoitaja lähettää reseptin keskuspalvelimelle, eli eräänlaiseen postilaatikkoon. Postilaatikko antaa lääkärille ilmoituksen reseptin perillemenosta tai virheilmoituksen, mikäli järjestelmässä on jotain vialla, jolloin potilaalle voidaan tilalle antaa paperinen resepti. Noutaessaan lääkettä potilas voi valita minkä tahansa apteekin, sillä kaikki Ruotsin apteekin kuuluvat samaan verkkoon. Farmaseutti noutaa potilaan tunnistusta vastaan halutun reseptin postilaatikosta ja toimittaa lääkkeen potilaalle. Ruotsin järjestelmän toimintaa ja yksinkertaisuutta hahmotetaan kuvassa 5. [Larsson et al. 2003; Larsson 2001; Sjöborg et al 2006 s. 4]

Ruotsissa sähköinen resepti on otettu hyvin vastaan. Apoteket AB kertoo, että kuukausittain lähetetään sähköisesti jo yli miljoona sähköistä reseptiä ja jopa 95 prosenttia sähköistä reseptiä kokeilleista potilaista olisi valmis käyttämään sitä uudelleen. Yleistyneen käytön ansiosta lääkärit ja apteekkihenkilökunta säästävät ajassa noin puoli tuntia joka päivä. Tämä aika voidaan käyttää potilaiden kanssa keskusteluun tai neuvontaan, jolloin kaikki hyötyvät järjestelmästä. Järjestelmän lähettämät resepti-viestit kulkevat XML-muodossa ja vuoden 2006 kesäkuussa tätä varten kehitettiin yhdessä joidenkin kuntien, Apoteket AB:n ja Carelink AB:n kanssa oma standardi. Standardi kulkee nimellä Baseline 1.0 ja sen pitäisi tulevaisuudessa tuoda lisää ominaisuuksia ja hyötyä sähköiseen reseptiin. Carelink AB on ruotsalainen ICT- organisaatio (Information and Communication Technology), joka toimii yhteistyössä Ruotsin terveydenhuollon kanssa. Se kehittää muun muassa Sjunetiä jatkuvasti paremmaksi ja tehokkaammaksi. [e-recept Stockholm 2006]

### 3.3 Tanska

Tanskassa erilaisia terveydenhuollon tietotekniikkaratkaisuja on suunniteltu jo 1980-luvulta lähtien. Ensin suunnitelmat ja kokeilut olivat pääsääntöisesti paikallisia, mutta vuodesta 1992 lähtien on esiintynyt myös suurempia pilottiprojekteja. Kolme ensimmäistä suurta projektia koskivat sähköistä tiedonsiirtoa, EDI:ä (Electronic Data Interchange). Näiden seurauksena perustettiin laajempialainen organisoitu projekti, johon liittyivät hallitus, kunnat, yksityisiä yrityksiä ja terveydenhuollon organisaatioita. Projekti sai nimen MedCom – Tanskalainen Terveydenhuollon Tietoverkko. MedComin tarkoituksena oli tuottaa kansallisia standardeja eri terveydenhuollon toimipisteiden väliseen kommunikointiin, kuten reseptien lähettämiseen. MedComin alla toimi 25 pilottia vuosina 1994-1996 ja niiden seurauksena käynnistettiin uusi projekti – MedCom II. MedCom II:n ansioista EDI kommunikointi organisaatioiden välillä yleistyi ja vuonna 1999 jo yli 2000 eri toimipistettä oli liittynyt yhteiseen tietoverkkoon. Sähköinen tiedonsiirto on jokapäiväinen osa tanskalaista terveydenhuoltoa ja suurin osa lääkäreistä, sairaaloista ja apteekkeista käyttää sitä. Sähköisen tiedonsiirron kerrotaan säästävän päivittäin 50 minuuttia aikaa lääkärin ja terveydenhoitajien vastaanotoilla ja kuukausittain lähetetään jo yli 2,3 miljoonaa sähköistä viestiä. [MedCom IV 2003]

Sähköisen reseptin kehittäminen aloitettiin Tanskassa jo 1988, jolloin käynnistettiin vuoteen 1991 asti kestänyt Amager-projekti. Sen tavoitteena oli luoda standardi reseptien siirtoon lääkäreiltä apteekkeihin. Sähköisessä reseptissä käytetään EDI-teknologioita, EDIFACT – standardia (Electronic Data Interchange For Administration, Commerce, and Transport) ja toimintaa varten kehitettyä postilaatikkomallia. Nykyisin tiedonsiirto tanskalaisessa terveystietoverkossa toimii kahdella eri tavalla. Vanhanmallinen VANS – verkko (Value Added Network Service) käsittelee EDIFACTiin perustuvat viestit ja muu liikenne hoidetaan IP-verkon (Internet Protocol) avulla. Tiedonsiirto näiden kahden verkon välillä tapahtuu sitä varten kehitetyn postistandardin avulla. IP-verkko puolestaan yhdistää kaikki Tanskan terveydenhuollon suojatut verkot yhteen kansalliseksi terveystietoverkoksi keskuskeskittimen avulla. [MedCom IV 2003; Wanscher 2006]

### 3.4 Saksa

Saksan sosiaali- ja terveysministeriö ilmoitti vuoden 2003 toukokuussa aloittavansa kehittämään uutta infrastruktuuria telematiikalle, jotta sähköistä kommunikointia voitaisiin parantaa ja lisätä sekä silloinen kansalaisten sairausvakuutuskortti voitaisiin muuttaa sähköiseksi. Saksassa sähköisen reseptin kehittäminen aloitettiin vuonna 2003 kansallisen elektronisen terveystiedon yhteydessä. Tarkoituksena oli, että lääkäri voisi siirtää reseptin tiedot potilaan terveystiedolle, jolloin tämä hallitsisi omia tietojaan koko ajan ja voisi valita minkä tahansa apteekin, josta lääkkeen noutaa. Näin välttyttäisiin sekavilta paperiresepteiltä, eikä farmaseuttien tarvitsisi enää siirtää tietoja paperilta heidän tietojärjestelmiinsä vaan se siirtyisi automaattisesti potilaan terveystiedon luettuna. Reseptin toimittamista eivät terveystiedon käytettäessä näin ollen haittaa katkokset tietoverkoissa tai muissa järjestelmissä. [Dietzel et al. 2006]

Vuoden 2006 alusta lähtien saksalaisten sairausvakuutuskortteja on alettu muuttaa elektronisiksi terveystiedoiksi, jotka toimivat sekä sairausvakuutuskorttina että tietovarastona potilasrekistereille, sähköisille resepteille ja muulle vastaavalle tiedolle. Jokaisessa kuvallisessa kortissa on mikrosiru, jolla voidaan tunnistaa ja autentikoida potilas sekä salata ja sähköisesti allekirjoittaa tietoja. Kortin saavat kaikki sairausvakuutuksen haltijat, sillä Saksassa sairausvakuutus on pakollinen vain tietyn tulorajan alittaville kansalaisille ja tämän tulorajan ylittävillä se on vapaaehtoinen. Vakuutusettomia on kuitenkin vain murto-osa saksalaisista. Vaikka saksalaiset saavat elektronisen terveystiedon, on sen käyttö kiinni potilaasta. Jokainen voi itse päättää haluaako käyttää korttia esimerkiksi sähköisten reseptien kuljettamiseen vai haluaako vielä tyytyä perinteiseen paperireseptiin. Tällä hetkellä (kesä 2006) elektronisen terveystiedon toimintaa testataan erilaisilla koeprojekteilla ja sähköisen reseptin kuljetus kortilla on ensimmäinen pakollinen ominaisuus, joka kortilla halutaan toimivan. Tämän jälkeen kortille voidaan siirtää muita vapaaehtoisia tietoja, kuten aiemmat lääkitykset tai terveystiedot hätätilanteita varten. [Dietzel et al. 2006; RIDE 2006]

### **3.5 Yhdysvallat**

Yhdysvalloissa presidentti George W. Bush on kaudellaan pyrkinyt lisäämään sähköisten terveyspalvelujen käyttöä ja yksi näistä on sähköinen resepti. Ensimmäisen sähköistä reseptiä koskevat lait allekirjoitettiin vuoden 2003 lopulla ja siitä lähtien on sähköistä reseptiä ja siihen liittyviä lakeja ja standardeja pyritty kehittämään mahdollisimman tehokkaasti. Uusimmat sähköisen reseptin toimintaa helpottavat säädökset julkaistiin lokakuussa 2005 USA:n terveys ja ihmispalveluiden ministeriön toimesta (HHS – United States Department of Health and Human Services). Vuoden 2006 tammikuussa USA:ssa käynnistettiin sähköisen reseptin pilotti, jolla testataan sähköistä reseptiä varten suunniteltuja standardeja käytännössä. Pilotin on tarkoitus kestää koko kalenterivuosi 2006 ja siitä saatavien kokemusten avulla suunnitellaan lopulliset sähköisen reseptin standardit, jotka tulevat käyttöön viimeistään 1. huhtikuuta 2008. [HHS 2006]