**TEKNIIKKA JA LIIKENNE**

**Tietotekniikka**

**Tietoliikennetekniikka**

**INSINÖÖRITYÖ**

# Kertatunnistaminen tietoliikennepalveluissa

**Työn tekijä: Jesse Kauppinen**
**Työn valvoja: Jukka Louhelainen**
**Työn ohjaaja: Risto Mäkipää**

**Työ hyväksytty: ___. ___. 2008**

**Jukka Louhelainen**
**lehtori**

Metropolia

**ALKULAUSE**

Tämä insinöörityö tehtiin Risto Mäkipää Innovationille ja keksintösäätiölle. Työn tarkoituksena oli tutkia, kuinka eDWarden voitaisiin toteuttaa mobiiliympäristössä. Kiitän Risto Mäkipäätä hänen antamistaan ohjeista ja neuvoista koskien työni tekemistä. Sain häneltä paljon hyödyllistä tietoa eri standardeja tutkiessani. Kiitän myös Timo Seppää ja keksintösäätiötä heidän antamastaan tuesta. Lisäksi haluaisin kiittää valvojaani lehtori Jukka Louhelaista hänen antamistaan ohjeista insinöörityötäni koskien sekä lehtori Jussi Alhorinnettä työni kieli- ja ulkoasun ohjauksesta.

Helsingissä 7.11.2008

Jesse Kauppinen

Metropolia

**TIIVISTELMÄ**

**Työn tekijä:** Jesse Kauppinen

**Työn nimi:** Kertatunnistaminen tietoliikennepalveluissa

**Päivämäärä:** 7.11.2008 **Sivumäärä:** 100 s.

**Koulutusohjelma:** Tietotekniikka

**Suuntautumisvaihtoehto:** Tietoliikennetekniikka

**Työn valvoja:** lehtori Jukka Louhelainen

**Työn ohjaaja:** Risto Mäkipää

Tämän insinöörityön lähtökohtana oli tutkia eDWardenia, joka on sähköisten palvelujen järjestelmä, sekä sitä, kuinka eDWarden voitaisiin toteuttaa mobiiliverkossa. Työssä tarkastellaan eDWardenin erilaisia suojausprotokollia (SSL, PKI, SOAP jne.), joita eDWarden toteutuksessa voidaan käyttää, sekä joitain palvelualustoja (OMA-standardeja ja Citrix), joihin eDWarden voitaisiin integroida. Lisäksi työssä käydään pintapuolisesti läpi pakettikytkentäisiä verkkoja, jotka mahdollistavat sähköiset palvelut mobiiliverkossa. Työssä tutkitaan myös hieman millaisia palveluita on tarjolla sekä millaisia tietoturvauhkia mobiiliverkoissa on.

Työn lopuksi luodaan eDWardenille arkkitehtuuri käyttäen tässä työssä läpikäytyjä protokollia. Esimerkkien avulla tutkitaan muutamia eDWardeniin suunniteltuja palveluja. Yhdestä palvelusta tehdään lisäksi HTML:llä demo, jossa näytetään palvelun toiminta käyttäjän kannalta.

eDWarden on sähköisten palvelujen järjestelmä, joka on kehitetty korjaamaan nykyisten sähköisten palveluiden aiheuttamia ongelmia. eDWarden on verkosta ladattava työpöytä, joka mahdollistaa kertakirjautumisen käytön sähköisissä palveluissa. eDWardenin avulla voidaan kaikki kansalaista koskevat tiedot koota yhteen paikkaan. Lisäksi e-desktop toimiympäristön avulla voidaan luoda, tallentaa ja allekirjoittaa omia asiakirjoja. eDWardenin avulla on mahdollista toteuttaa kahden kansalaisen välinen asiointi.

**Avainsanat:** eDWarden, Open Mobile Alliance, SSO.

Metropolia

**ABSTRACT**

**Name:** Jesse Kauppinen

**Title:** Single Sign-On on electric transaction services

**Date:** 7.11.2008 **Number of pages:** 100

**Department:** Information Technology **Study Programme:** Telecommunication

**Instructor:** Lecturer, Jukka Louhelainen

**Supervisor:** Risto Mäkipää

The purpose of this graduation study is to examine eDWarden, which is a electric transaction service and examine how it could be executed on mobile network. In this graduation study are also examined security protocols (like SSL, PKI, SOAP etc.) and some service platforms (OMA standards and Citrix) where eDWarden could be integrated. There is also quick look on packed switching networks that make possible of provide electronic transaction services on mobile phones and examined what kind of electronic transaction services there are and what kind of security issues there is on mobile networks.

In the end of this graduation study, working eDWarden architecture is created using standards that has been examined in this graduation study and examined with examples fem of the services that eDWarden can provide. A demo is made one of those examples using HTML-language, to show how that particular service works on mobile users perspective.

eDWarden is an electronic transaction service that is developed to solve that electronic transaction services currently have. EDWarden is an e-desktop program that provide Single Sign-On services for electronic transaction services. With eDWarden it is possible to collect all information regarding a legal person in on place and with e-desktop environment user can create, sign and save his own documents. With eDWarden it is possible to have verified conversation between two legal persons.

**Keywords:** eDWarden, Open Mobile Alliance, SSO.

**SISÄLLYS**

**LYHENTEET**

| | |
|---|---|
| AAA | Authentication, Authorization and Accounting; Tunnistus, valtuuttaminen ja tilastointi. |
| AH | Authentication header; tunnistusotsikko. |
| API | Application programming interface; ohjelmointirajapinta. |
| ASP | Authentication Service Provider; tunnistuspalvelun tarjoaja. |
| AuC | Authentication Center; Autentikointi (Tunnistus) keskus. |
| BSC | Base Station Controller; tukiasemaohjain. |
| BSS | Base Station Subsystem; tukiasemajärjestelmä. |
| BTS | Base Transceiver Station; tukiasema. |
| CA | Certificated Authority ; varmenteen antaja. |
| CDC | Connected Device Configuration; mobiili Javan monipuolisempi konfiguraatio. |
| CEK | Content Encryption Key; sisällön salausavain. |
| CLCD | Connected Limited Device Configuration; mobiili Javan pieniresurssisille laitteille tarkoitettu konfiguraatio. |
| CoA | Care-of Address; mobiililaitteille Mobiili-IP:ssä määriteltävä IP-osoite silloin kuin ne ovat muualla kuin kotiverkossa. |
| DCF | DRM Content Format; pakkaustapa. |
| DER | Distinguished Encoding Rules; koodausalgoritmi, jolla tietorakenteet esim. X.509-varmenne saadaan verkossa siirrettävään muotoon. |
| DHCP | Dynamic Host Configuration Protocol; Protokolla joka määrittää IP-osoitteen verkossa vierailevalle laitteelle. |
| DNS | Domain Name System; Hakemisto, joka kartoittaa domainien nimet IP-osoitteisiin. |
| DRM | Digital Rights Management; Digitaaliset käyttöoikeudet. |
| DTD | Document Type Definition; Dokumentin tyypin määritys. |
| EAP | Extensible Authentication Protocol; tunnistusprotokolla. |
| EAPoL | Extensible Authentication Protocol over LAN; EAP-tunnistus lähiverkon yli. |
| eDAPI | eDesktop Application programming interface; eDWarden työpöydän rajapinta. |
| eID | electronic personal Identification Card ; älykortti. |
| ESP | Encapsuling Security Payload ; hyötykuorman turvallisuus kapsulointi. |
| GGSN | Gateway GPRS Support Node; yhdyskäytävä solmu. |
| GPS | Global Position System; satelliittipaikannus järjestelmä. |

| | |
|---|---|
| GPRS | General Packet Radio Service; pakettikytkentäinen tiedonsiirtopalvelu. |
| GSM | Global System for Mobile communication; kännykkäverkko. |
| HST | Henkilön sähköinen tunnistus. |
| HTML | Hypertext Markup Language; kuvauskieli, jolla verkkosivut rakennetaan. |
| HTTP | Hypertext Transfer Protocol; selaimien käyttämä tiedonsiirtoprotokolla. |
| HTTPS | Hypertext Transfer Protocol over Secure Socket Layer; selaimien käyttämä tiedonsiirto protokolla, jossa SSL-suojaus. |
| ICA | Independent Computing Architecture; Citrixin käyttämä etäyhteyden muodostamisprotokolla. |
| ID-FF | Identity Federation Framework; Liberty Allianssin verkkotunnistus standardikehys. |
| ID-WSF | Identity Web Service Framework; Liberty Allianssin tekemä tunnistusverkkopalvelun kehysstandardi. |
| IETF | Internet Engineering Task Force; Internet-protokollien standardoinnista vastaava organisaatio. |
| IKE | Internet Key Exchange; internetavainten vaihtoprotokolla. |
| IMSI | International Mobile Subscriber Identity; Kansainvälinen mobiilitilaajan tunnistus. |
| IP | Internet Protocol; Internet-protokolla. |
| IV | Initialization Vector; Aloitusvektori. |
| J2ME | Java 2 Mikro Edition; mobiililaitteiden Java-ympäristö. |
| J2SE | Java 2 Platform, Standard Edition; työpöytäkoneille tarkoitettu Java-ympäristö. |
| JDEC | Java Database Connectivity; Java tietokantarajapinta. |
| JNDI | Java Naming and Directory Language; Java hakemistorajapinta. |
| LAN | Local Area Network; lähiverkko. |
| LDAP | Lightweight Directory Access Protocol; hakemistopalvelujen käyttöön tarkoitettu verkkoprotokolla. |
| LTE | Long Term Evolution; 3G-tekniikka mobiilitekniikka. |
| MAC | Message Authentication code; Viestin tunnistuskoodi. |
| MCS | Mobile Services switching Centre; mobiilikeskus. |
| MEP | Message External Pattern; viestin ulkoinen kuvio. |
| MMS | Multimedia Messaging Service; mobiiliviestintä. |
| MIDP | Mobile Information Device Profile; Java-profiili, jonka avulla mobiililaitteisiin voi liittää esim. kuvia tai ääntä. |
| NAI | Network Access Identifier; tunnistusprotokolla. |
| OCSP | Online Certificate Status Protocol; varmenteen tilanneprotokolla. |

| | |
|---|---|
| OMA | Open Mobile Alliance; Allianssi, joka tekee mobiilistandardeja. |
| ORB | Object Request Broker; ohjelma, joka mahdollistaa ohjelman yhteyden ottamisen toiseen ohjelmaan verkon välityksellä. |
| OWSER | OMA Web Service Enabler release; OMA Nettipalvelujen tarjoaja. |
| PCU | Packet Control Unit; Paketinohjausyksikkö. |
| PDP | policy decision point; abstraktin valtuutusjärjestelmän kohta, jossa päätös käyttäjän pyynnön hyväksymisestä tai hylkäämisestä tehdään. |
| PEAP | Protected Extensible Authentication Protocol; Suojattu tunnistusprotokolla. |
| PGP | Pretty Good Privacy; Verkkomainen varmennearkkitehtuuri. |
| PIC | Pre-IKE Credential Provisioning Protocol; Protokolla joka integroi legacy-tunnistuksen IPSec-protokollaan. |
| PIN | Personal Identification Number; Henkilökohtainen tunnusnumero. |
| PKI | Public Key Infrastructure; julkisen avaimen järjestelmä. |
| PPP | Point to Point Protocol; Verkkolaitteiden yhteysprotokolla. |
| PSO-ID | Provision Service Objekt Identifier; Globaalitunniste (SPML:ssä). |
| PSP | Provisioning Service Provider: ohjelmisto, joka vastaa RA:n lähettämään SPML-pyyntöön. |
| PST | Provisioning Service Target: Ohjelman varauksen hoitava taho (SPML:ssä). |
| PSTD-ID | Provisioning Target Data Identifiers; yksittäisen PSP:n liitettävät tunnisteet. |
| RA | Requesting Authority; varauspyynnön tekijä (SPML:ssä). |
| RADIUS | Remote Authentication Dial-In User Service: palvelin/käyttäjä protokolla, jolla hoidetaan käyttäjän tunnistus, valtuutus ja tilastointi. |
| REK | Rights Encryption Key; Oikeuksien suojausavain. |
| REL | Rights Expression Language; Kieli, joka määrittää OMA DRM materiaalin oikeudet. |
| RMI | Java Remote Method Indicator; hakemistopalvelu, joka tarjoaa pohjan asiakas-palvelin arkkitehtuurille verkossa. |
| RNC | Radio Network Controller; UMTS-verkon tukiasemaohjain. |
| ROAP | Rights Object Acquisition Protocol; Oikeuksien hankintaprotokolla. |
| SA | Security Association; Turvallisuusyhteistyö. |
| SAD | Security Association Database; Turvallisuusyhteistyötietokanta. |
| SAML | Security Assertion Markup Language; XML-standardi, joka vaihtaa tunnistus- ja valtuutustietoja eri turvallisuus domainien kesken. |
| SATU | Sähköinen asiointitunnus. |
| SGSN | Serving GPRS Support Node; Operointisolmu. |
| SIM | Subscriber Identity Model; SIM-kortti. |
| SIP | Session Initiation Protocol: Istunnon aloitusprotokolla. |

| | |
|---|---|
| SMS | Short Message Service; tekstiviestipalvelu. |
| SOAP | Simple Object Access Protocol; XML-pohjaisten viestien lähettämisen tietoliikenneprotokolla. |
| SPD | Security Policy Database; Turvallisuusmenettelytietokanta. |
| SPI | Security Parameter Index; Turvallisuusparametrihakemisto. |
| SPML | Service Provisioning Markup Language; XML-pohjainen kieli hankinta pyyntöjen ja vastausten välittämiseen. |
| SQL | Structured Query Language; tietokantojen hallitsemiseen tarkoitettu ohjelmointikieli. |
| SSL | Secure Socket Layer; Kuljetuskerroksen turvallisuus protokolla. |
| SSO | Single Sign-On; kertakirjautuminen. |
| SSOS | Single Sign-On Service; kertakirjautumispalvelu. |
| TCP | Transmission Control Protocol; Yhteyden muodostamisprotokolla kahden tietokoneen välille. |
| TLS | Transport Layer Security; Kuljetuskerroksen turvallisuusprotokolla. |
| UDP | User Datagram Program; Yhteyskäytäntö, jolla sovellus lähettää tietoa koneelle. |
| UMTS | *Universal Mobile Telecommunication System*; kolmannen sukupolven matkapuhelin tekniikka. |
| URI | Uniform Resource Identifier; merkkijono, jossa kerrotaan tietyn tiedon paikka (URL). |
| URL | Uniform Resource Locator; www-sivujen osoitin. |
| USB | Universal Serial Bus; sarjaväyläarkkitehtuuri. |
| USIM | Universal Subscriber Identity Model; UMTS verkossa käytettävä SIM-kortti. |
| VoIP | Voice over Internet Protocol; äänen ja videokuvan siirtäminen reaaliaikaisesti internetin välityksellä. |
| XHTML | eXtensible Hypertext Markup Language; web-sovellusten kuvauskieli. |
| XML | eXtensible Markup Language; merkintäkieli. |
| XrML | XML-based rights managment language; XML-pohjainen käyttöoikeuksien myöntämiseen tarkoitettu kieli. |
| WiMAX | Worldwide Interoperability for Microwave access; Kehitteillä oleva langaton laajakaistatekniikka. |
| WLAN | Wireless Local Area Network; langaton verkko. |
| WSDL | Web Service Defination Language; verkkopalvelujen kuvauskieli. |
| WSP | Web Service Provider; verkkopalveluiden tarjoaja. |
| WSR | Web Service Requester; verkkopalvelun pyytäjä |
| WWW | World Wide Web; maailmanlaajuinen Internet-verkko. |

# 1 JOHDANTO

Tässä työssä käsitellään eDWardenia, joka on sähköisen asioinnin menetelmä, sekä tutkitaan, onko eDWardenin patenttihakemuksessa esitetyt toiminnat mahdollista toteuttaa integroimalla eDWarden Open Mobile Allianssin standardeihin. Tässä työssä keskitytään lähinnä eDWardenin mobiilipuoleen.

Työn aluksi tutkitaan erilaisia standardeja ja protokollia, jotka liittyvät eDWardeniin sekä tutkitaan, miten sähköisiä palveluita voidaan käyttää moobiililaitteilla. Lisäksi tutustutaan Open Mobile Allianssin (OMA) kahteen standardiin, joihin eDWarden olisi tarkoitus integroida. Kappaleessa 2 tutkitaan nykyisiä sähköisiä asiointipalveluja sekä niiden ongelmia. Kappaleessa 3 tutustutaan päällisin puolin mobiiliverkkoihin jotka mahdollistavat internetpalvelut kännyköihin. Kappaleessa 4 käsitellään mobiiliverkkojen tietoturvariskejä sekä tutustutaan muutamaan erilaisiin suojaustekniikoihin. Kappaleessa 5 tutkitaan, kuinka sähköiset palvelut tarjotaan mobiiliverkon kautta ja kappaleessa 6 tutustutaan Single Sign-On:n eli kertakirjautumiseen. Kappaleessa 7 käydään läpi kahta Open Mobile Allianssin standardia (OWSER ja DRM). Kappaleessa 8 tutustutaan pikaisesti Citrixin tarjoamiin mobiilisovelluksiin. Kappaleessa 9 tutustutaan eDWardenin arkkitehtuuriin ja toimintaan sekä tutkitaan, voidaanko eDWarden integroida OMA:n standardeihin ja voidaanko toteuttaa eDWardenin patenttihakemuksessa mainitut toiminnat OMA:n standardien avulla. Kappaleessa 10 käydään läpi muutamia mahdollisia mobiilipalveluita ja kappaleessa 11 esitellään eDWardenin demo, joka tehtiin osana tätä työtä. Kappaleessa 12 ovat työn ja tutkimuksen pohjalta tehdyt johtopäätökset. Koska eDWardenin patenttihakemuksen esimerkeissä on käytetty yksinomaan Java-alustaa, on myös tässä työssä olevien eDWarden esimerkkien oletettu toimivan Java-ympäristössä. Todellisuudessa eDWarden voidaan kuitenkin toteuttaa myös muussa kuin Java-ympäristössä.

## 2 SÄHKÖISET ASIOINTIPALVELUT

Nykyään sähköisissä palveluissa kansalainen tunnistetaan sähköisen henkilökortin tai muun sähköisen tunnistusmenetelmän avulla. Kansalaisille tarjotaan yksityisiä asiointipalveluja esim. sähköisten lomakkeiden täyttöä ja allekirjoitusta. Kansalainen ei voi ottaa vastaan tietoa tai asiakirjoja siten, että ne olisivat käypiä jossain toisessa yhteydessä, koska jokainen palveluntarjoaja tarjoaa kansalaiselle palveluja oman tiedonkäsittelytarpeensa tai velvoitteensa pohjalta. Kahden kansalaisen välistä keskinäistä todennettavaa asiointia ei ole. Viranomaisten kanssa asioidessa kansalainen on pelkkä passiivinen tunniste ja tiedon luovuttaja, eikä hän saa mitään vastinetta omalle työpöydälleen.

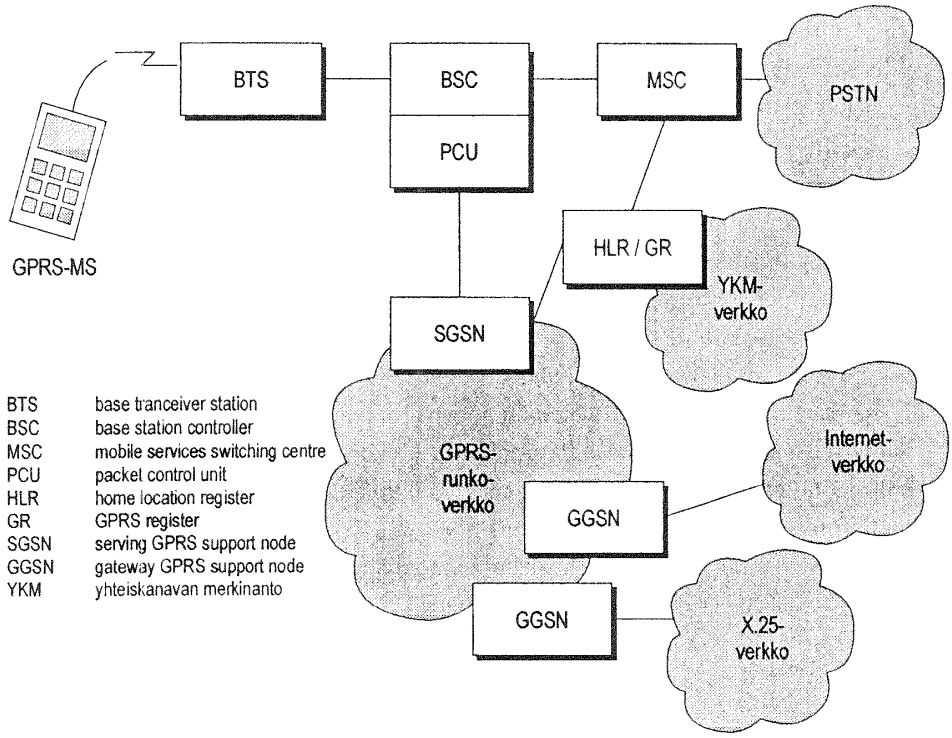
Ongelmaa on alettu ratkaista kehittämällä viranomaisten ja yritysten hallussa olevien kansalaista koskevien tietojen keskinäistä vaihtoa. Tämä tehostaa asioiden hoitoa, mutta kansalaisella ei kuitenkaan ole minkäänlaista mahdollisuutta puuttua, valvoa tai tarkastaa hänestä verkon kautta kerrottavista tiedoista. Ongelman perimmäinen syy on se, että kansalainen on vain passiivinen tunniste verkossa, ei aktiivinen toimija eikä tiedon käsittelijä niin kuin fyysisessä maailmassa. Lisäksi kansalaista käsittelevät tiedot ovat hajallaan verkossa, koska jokaisella palveluntarjoajalla on kansalaisesta vain palveluntarjoajan kannalta tärkeät tiedot. Ratkaisuna näihin ongelmiin on tehdä kansalaisesta itsenäinen, verkossa tietoa käsittelevä prosessi. Juuri tämä on yksi eDWardenin perusajatuksista. eDesktop-järjestelmä tekee myös kansalaisesta itsenäisen kokonaisuuden verkkoon, jolloin kaikki kansalaista koskevat tiedot ovat samassa paikassa. [1.]

## 3 MOBIILIVERKON INTERNET-PALVELUT

Tässä kappaleessa tutustutaan pääpiirteittäin mobiiliverkkoihin, joitten kautta Internet-palveluita voi käyttää kännykällä.

## 3.1 GPRS ja muut mobiiliverkot

GPRS (*General Packet Radio Service*) on GSM (*Global System for Mobile communication*) -verkon laajennus, joka on tarkoitettu purskeisen Internet-protokollan (IP) mukaiseen tiedonsiirtoon, eli fyysistä yhteyttä ei ole jatkuvasti varattu, vaan se on aktiivinen ainoastaan tiedonsiirron aikana. GPRS-verkosta voidaan yhdistyä suoraan internet-verkkoon ja se näkyy ulkopuolisille dataverkoille yhtenä normaalina Internet-aliverkkona. [2, s. 49 - 50.]




*Kuva 1: GPRS-arkkitehtuuri [2, s. 52].*

GPRS toi GSM-verkkoon seuraavat lisäosat:

- Paketinohjausyksikkö (PCU, *Packet Control Unit*) huolehtii GSM-tukiasemajärjestelmän (BBS) ja GPRS-runkoverkon välisestä yhteydestä. PCU erottelee GPRS-paketit piirikytkentäisistä yhteyksistä ja lähettää ne SGSN-elementille.

- Operointisolmu (SGSN, *Serving GPRS Support Node*) toimii samalla hierarkkisella tasolla kuin matkapuhelinkeskus (MSC), ja se seuraa yksittäisten päätelaitteiden sijaintia, joiden sijainnin se tietää solun tai reititysalueen tarkkuudella riippuen päätelaitteen liikkuvuuden hallintatilasta. SGSN huolehtii myös käyttäjän tunnistamisesta ja suojaukseen liittyvistä toiminnoista sekä verkkoon pääsystä. Lisäksi SGSN-elementin kautta voidaan kerätä laskutustietoja GPRS-yhteyksistä. Tärkein laskutukseen vaikuttava seikka on radiorajapinnan resurssien käyttö eikä yhteyden kesto niin kuin GSM:ssä.

- Yhdyskäytäväsolmu (GGSN, Gateway GPRS Support Node) mahdollistaa tiedonsiirron ulkoisten tietoverkkojen ja GPRS-verkon välillä. GGSN on kytketty SGSn:n IP-pohjaisen GPRS-runkoverkon kautta. GGSN vastaa periaatteessa normaalin runkoverkon reititintä paitsi, että sen täytyy pystyä reitittämään datayhteydet liikkuvasta ympäristöstä (koska kännykän käyttäjä saattaa liikkuessa siirtyä toisen tukiaseman (BTS) alueelle, jolloin myös yhteydestä vastaava SGSN vaihtuu toiseksi.) GGSN tietää käyttäjän sijainnin SGSN:n tai solun tarkkuudella riippuen päätelaitteen ja GGSN:n toimintatilasta. [2, s. 53 – 55.]

UMTS (*Universal Mobile Telecommunication System*) on kolmannen sukupolven pakettikytkentäinen matkapuhelintekniikka joka perustuu GSM- ja GPRS-verkkoon. UMTS-verkko kehittyi pikkuhiljaa ja sillä on nykyään useita eri spesifikaatioita. Ensimmäisessä spesifikaatiossa (Release 99) UMTS-verkon radiojärjestelmä osa muutettiin GSM/GPRS-verkosta, mutta sen runkoverkko-osa oli samanlainen kuin GPRS-verkolla. Myöhemmissä määrityksissä näitä verkkomäärityksiä on muutettu ja verkkoon on lisätty uusia toimintoja.

Tulevaisuudessa puhelinverkoissa siirryttäneen kokonaan IP-pohjaisiin verkkoihin, jolloin puheluidenkin reititys hoidetaan reitittimien ja palvelimien avulla. Tällä hetkellä kehitteillä on niin sanotun neljännen sukupolven eli 4G matkapuhelin tekniikat.

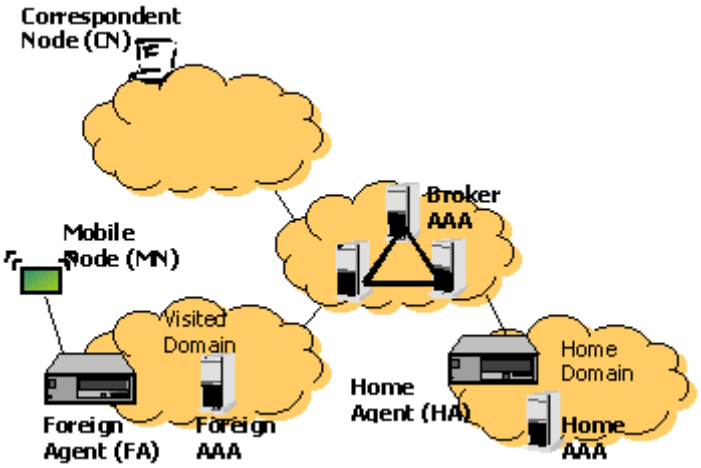
Tällaisia tekniikoita ovat mm. LTE (*Long Term Evolution*) ja WiMAX (*World-wide Interoperability for Microwave access*.). LTE on tekniikka, joka tukeutuu vahvasti nykyisin 3G-verkkoihin ja pyrkii parantamaan niiden palveluja sekä nopeuttamaan tiedonsiirtoa. WiMAX puolestaan on kehityksen alla oleva langaton laajakaistatekniikka, jonka tarkoituksena on tarjota käyttäjille liikennöintinopeuksiltaan nykyisiä kaapelimodeemiyhteyksiä vastaava langaton verkkoyhteys. [3.]

### 3.2 Mobiili-IP

Normaalissa IP-verkossa IP-osoitteet tunnistetaan laitteen sijainnin mukaan, samalla tavalla kuin normaalissa puhelinverkossa löydetään numeroa vastaava puhelin. Tämä helpottaa reitittimien toimintaa, sillä niiden täytyy vain katsoa reititystaulustaan oikea liitäntä (interface), josta IP-paketit lähetetään eteenpäin seuraavalle reitittimelle. Käyttäjän päätelaitteen siirtyessä toiseen verkkoon se saa uuden IP-osoitteen. IP-osoitteen muuttuessa lähetyksen aikana TCP (*Transport Control Protocol*) -yhteys katkeaa, ja se täytyy yhdistää uudelleen. Tällainen aiheuttaa ongelmia sellaisten ohjelmien kanssa, jotka olettavat yhteyden pysyvän päällä koko yhteydenpidon ajan. Vanhoissa IP-verkoissa tämä ei aiheuttanut ongelmia, mutta mobiilikäyttäjät saattavat siirtyä verkosta toiseen kesken lähetysten, mikä saattaa aiheuttaa ongelmia joissain IP-sovelluksissa.

Mobiili-IP-protokollan tarkoituksena on tarjota mobiilikäyttäjille samat palvelut kuin normaaliverkon käyttäjille, ilman että verkon sovelluksia pitäisi muuttaa. Mobiili-IP mahdollistaa käyttäjän liikkumisen verkon solmujen välillä ja pitämään silti TCP-yhteyden toiminnassa koko ajan. Mobiili-IP:ssä jokaisella mobiilipäätteellä (esim. kännykkä) on kaksi IP-osoitetta. Pysyvä kotiosoite (Home Address), joka päätelaitteella on kotiverkossa, sekä Care-of Address (CoA) joka määritellään päätteelle sen käydessä muissa verkoissa. Käyttäjän kotiverkossa sijaitseva koti-agentti (Home Agent) päivittää käyttäjän sijainnin verkossa ja vieras-agentti (Foreign Agent) tai DHCP (*Dynamic Host Configuration Protocol*) antaa Care-of-Address IP-osoitteen verkossa vierailleville käyttäjille.

Päätelaite kuuntelee mobiiliagenttien (koti- ja vieras-agentit) verkkoon lähettämiä yleislähetysmainostuksia (advertisements broadcast), jotka ilmaisevat, minkä verkon alueella käyttäjän päätelaite on. Käyttäjän siirtyessä toiseen verkkoon päätelaite havaitsee liikkumisen ja yrittää saada väliaikaisen IP-osoitteen verkon DHCP:ltä. [4.]




*Kuva 2: Mobiili-IP:n esimerkkiarkkitehtuuri, kun käytetään Diameter-protokollaa [5].*

Kuvassa 2 näkyy esimerkki AAA- (*Authentication, Authorization and Accounting*) ja Diameter-protokollaa käyttävän Mobiili IP -verkon arkkitehtuurista. Diameter on RADIUS (*Remote Authentication Dial-In User Service*) -protokollasta kehitetty protokolla, joka mm. yksinkertaistaa mobiilikäyttäjien hallintaa verkossa ja käyttää NAI:hin (*Network Access Identifier*) pohjautuvaa tunnistustekniikkaa. Käyttäjän rekisteröimiseksi Foreign agent (vieras agentti) kysyy ensiksi Foreign AAA-palvelimelta (vieras AAA-palvelin) apua käyttäjän rekisteröinnin ajaksi. Vieras AAA-palvelin etsii päätelaitteen NAI-tunnuksen ja ottaa yhteyttä käyttäjän koti-palvelimeen (Home AAA). Käyttäjän AAA koti-palvelin autentikoi käyttäjän päätelaitteen Mobile-IP:n rekisteröimisviestissä olleen NAI-tunnuksen perusteella ja käynnistää käyttäjän koti-agentin (home agent). Kuvassa oleva Broker AAA tarkoittaa luotetun kolmannen osapuolen palvelimia, joiden kautta tieto saattaa tarvittaessa kulkea kotiverkon ja vieraanverkon välillä. [5.]

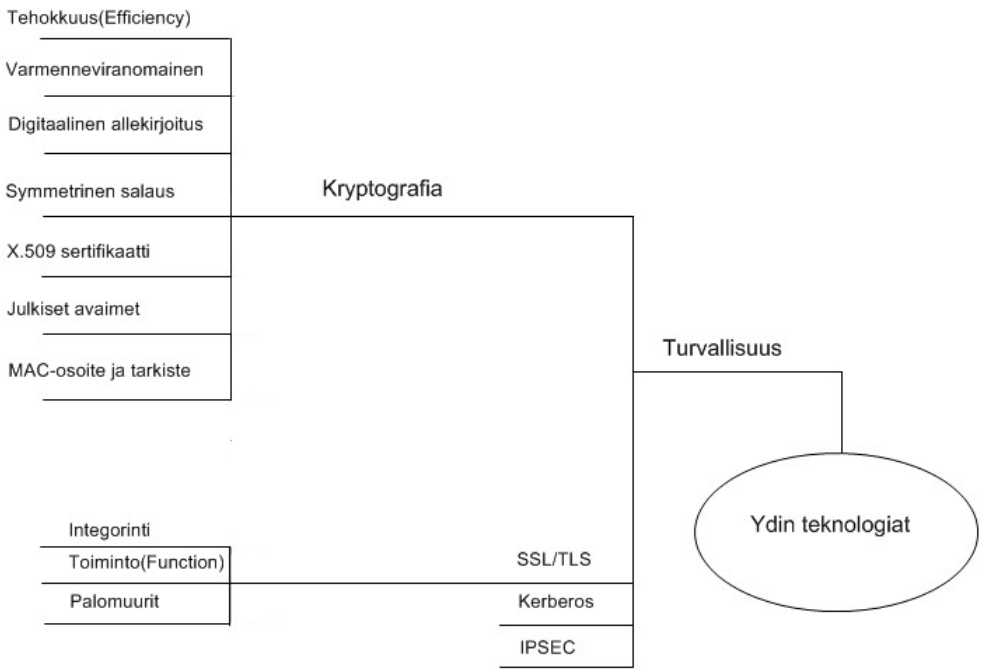
Rekisteröinnin päättymisen jälkeen kotiagentti pysäyttää päätelaitteelle lähetetyt paketit ja käyttää IP-in-IP–salausta luomaan tunnelin (suojattu yhteys) käyttäjän väliaikaiseen osoitteeseen, jonne se lähettää pysäyttämänsä paketit tekemänsä tunnelin kautta. [4.]

# 4 TIETOTURVALLISET MOBIILI PALVELUT

Sähköisen asioinnin perusedellytykset ovat turvallisuus ja todennettavuus. Tässä kappaleessa käydään läpi useita eri turvallisuusprotokollia ja tekniikoita sekä mobiiliverkon tietoturvariskejä.

AAA (*Authentication, Authorization and Accounting*) tarkoittaa suomeksi tunnistusta, valtuutusta ja kirjanpitoa (tunnistukseen perustuva resursointi). AAA on tunnistuspalvelu, joka mahdollistaa käyttäjän tunnistuksen, käyttäjien saamien palvelujen profiloimisen sekä tilastotietojen keräämisen kuten esimerkiksi yhteysaikoja, joka mahdollistaa esim. laskutuksen. AAA-protokollaa käyttäessä palvelu koostuu yleensä kolmesta eri komponentista: asiakkaasta, liityntäpisteestä (kytkimenportti tai langattoman verkon tukiasema) ja AAA-palvelimesta. Asiakkaan ja liityntäpisteen välillä on yleensä käytössä jokin alemman tason protokolla, joka tarjoaa tunnistuspalvelun (esim. EAP eli Extensible Authentication Protocol, joka käsitellään kappaleessa 4.2). Liityntäpiste välittää autentikointiprotokollan parametrit AAA-palvelimelle, joka suorittaa varsinaisen käyttäjäntunnistuksen. [6.]

Niin sanottuja AAA-protokollia eli protokollia, joiden avulla tunnistus, valtuutus ja tilastointi voidaan hoitaa, ovat RADIUS, Kerberos ja TACACS+. Tähän työhön liittyy lähinnä RADIUS. RADIUS -protokollaa käytetään mm. EAP-protokollissa. RADIUS on palvelin/käyttäjä–protokolla, eli sen käyttöön tarvitaan sekä erillinen palvelin että tietty käyttäjä. RADIUS:ta käytetään pääosin operaattorien tai yritysten sisäverkoissa, jolloin RADIUS:ta hallinnoi yksi ja sama taho. Tällöin palvelua voidaan pitää melko luotettavana. RADIUS käyttää yhteyden ottamiseen UDP (*User Datagram Program*) -protokollaa, sillä se takaa suurimman yhteysnopeuden. [7.]




*Kuva 3: Turvallisuustekniikat*

### 4.1 Tietoturvariskit mobiilissa verkossa

Mobiiliverkkojen siirtyessä piiripohjaisesta pakettikytkentäiseen verkkoon tuli mobiiliverkkoon myös uusia uhkia. Pakettipohjaisessa televerkossa ulkoiset rajapinnat kytkeytyvät IP-verkkojen kokonaisuuteen GGSN-yksiköiden kautta, joiden suojauksena käytetään yleensä palomuuria. Palomuurien suojausta pystytään kuitenkin horjuttamaan mm. pommittamalla palomuuria tahallisen suurella signaalikuormalla. Verkossa tehtyjä tietohyökkäyksiä ei myöskään välttämättä huomata heti, vaan hyökkäyksen aiheuttamat ongelmat saattavat näkyä esim. verkkojen tai palvelujen epämääräisenä toimintana. Laitteistojen, palvelujen ja ohjelmien monipuolistuminen tarjoaa uusien mahdollisuuksien lisäksi myös uusia haastavampia turvallisuuskysymyksiä.

Laittomilla tunkeutumisilla ja hyökkäyksillä voi olla monenlaisia tarkoituksia, kuten käyttöoikeuksien varastaminen ja niiden käyttäminen omiin tarkoituksiin, väärennettyjen viestien lähettäminen järjestelmään (esim. roskaposti) tai vahingollisten keinojen kehittäminen, jotka saavat verkon toimimaan virheellisesti (tällaista kutsutaan palvelunestohyökkäykseksi). Nykyaikaiset suojaustekniikat vaikeuttavat tällaisia hyökkäyksiä, mutta toisaalta Internetin välityksellä kaikki löydetyt tietoturva-aukot leviävät nopeasti yleisön tietouteen. Mikäli kaikkiin uhkakuviin halutaan varautua tehokkaasti, tarkoittaa se todennäköisesti käytettävyyden heikkenemistä aiempiin tietoliikenneratkaisuihin verrattuna, sillä yksinkertaistettu ja helppokäyttöinen käyttäjäympäristö saattaa mahdollistaa laajan väärinkäytön.

Verkkohyökkäykseen on monenlaisia tapoja. Yksi mahdollinen hyökkäyksen muoto on viruksien ajaminen päätelaitteille Javaa tai vastaavia ympäristöjä käyttämällä. Virukset voivat puhelimeen päästyään mm. käynnistää puheluita ja aiheuttaa siten haittaa käyttäjälle. Myös automaattiset SIM- (*Subscriber Identity Model*) ja USIM (*Universal Subscriber Identity Model*) -korttien päivitykset, kuten ensisijaisten operaattorilistojen päivitykset verkkovierailutarpeisiin saattavat avata mahdollisuuksia väärinkäytöksiin. Hyökkääjä voi myös pyrkiä tunkeutumaan palveluntarjoajien palvelimiin ja yrittää sitä kautta mm. hyökätä hyvin suojattuun verkkoinfrastruktuuriin tällaisen heikommin suojatun palvelimen kautta tai tehdä aktiivisia hyökkäyksiä valetukiasemien kautta. Palvelunestohyökkäykset tehdään usein hyödyntämällä ohjelmistojen virheitä tai puutteita, joiden avulla aiheutetaan mm. verkkokapasiteetin ylimääräistä kulumista, yritetään kaataa palveluja tai verkkoa tai hidastetaan niiden toimintaa. Yksi ongelma on myös se, ettei GSM- ja UMTS-salaus toimi kaikilla rajapinnoilla. GSM:ssä salaus toimii vain tilaajan päätelaitteen ja tukiaseman välillä, GPRS:ssä tilaajan päätelaitteen ja SGSN:n välillä ja UMTS:ssa tilaajan päätelaitteen ja RNC:n (*Radio Network Controller*) välillä. Jos näiden yhteyksien ulkopuolella käytetään radiolinkkiä, niin se on silloin suojaamaton, ellei linkissä ole erillistä salausjärjestelmää. Yleisiä tietoturvauhkia voivat olla mm.

- salakuuntelu tai dataliikenteen huomaamaton seuranta.

- oman identiteetin naamioiminen, jolloin väärinkäyttäjä hämää tilaajaa, uskottelemalla olevansa esim. operaattori saadakseen luottamuksellista tietoa käyttäjältä.
- luvaton tietokantojen selailu, jonka avulla väärinkäyttäjä voi löytää arkaluontoisia tietoja tilaajasta.
- tietosisällön manipulointi, jolloin hyökkääjä saattaa poistaa, lisätä tai muokata viestejä.
- häirintä tai väärinkäyttö, joilla hyökkääjä voi estää luvallisten käyttäjien liikenteen tai merkinannon. [8, 185 – 192.]

Mobiiliverkkojen uhat kuten virukset ovat nykyään arkipäivää myös kännykän käyttäjille, joten nykyään mobiilipuhelimiin on saatavilla myös erilaisia torjuntaohjelmia, kuten F-Secure ja Gold Lock. F-Secure Mobile Security on F-Securen mobiiliversio, jossa on mm. palomuuri, muistikortin automaattinen tarkastus, reaaliaikainen laiteen sisäinen suoja ja automaattiset päivitykset, jotka yhdessä suojaavat kännykkää erilaisilta hyökkäyksiltä haittaohjelmista aina tietomurtoihin saakka (lisätietoja osoitteessa http://www.f-secure.fi/tuotteet/fsms). Gold Lock on puolestaan Nokian N73-, N80-, E61- ja E70-mallien kanssa toimiva, puhelimesta toiseen ulottuva suojaus joka suojaa sekä tekstiviestejä että puhetta (lisää tietoa Gold Lockissa osoitteessa http://www.gold-lock.com/index.htm).

## 4.2 EAP

EAP (Extensible Authentication Protocol) on tunnistukseen käytettävä liikennöintiprotokolla. EAP ei ole autentikointimetodi tai autentikointiprotokolla vaan standardi, joka kertoo, miten tunnistusviestit vaihtuvat asiakkaan, autentikaattorin ja autentikointipalvelimen välillä. EAP tukee useita eri tunnistusprotokollia, joista osa on laitevalmistajakohtaisia. EAP on alun perin kehitetty toimimaan PPP:n (*Point to Point Protocol*) kanssa lähinnä piirikytkentäisiin puhelinverkkoihin. Uudempi versio EAP over LAN (EAPoL) on kehitetty toimimaan pakettikytkentäisissä verkoissa. EAPiä käytetään nykyään myös langattomien ja langallisten verkkojen suojaamiseen.

Eri EAP-protokollat toimivat hieman eri tavalla, mutta kaikki liikennöivät kuitenkin linkkitasolla, jolloin ne eivät tarvitse IP-osoitteita kuljettaakseen viestejä laitteelta toiselle. EAPin etuja ovat muuteltavuus, automaattinen tuki monille eri tunnistusprotokollille sekä se, että autentikaattori voi tunnistaa paikallisia asiakkaita samalla, kun se toimii läpikulkukäytävänä protokollille, joita se ei tue. [7.]

EAP-koostuu useista pyyntö/vastaus–pareista. EAPin suorittaminen alkaa käyttäjän lähettämästä EAP-pyynnöstä ja päättyy EAP-hyväksyntään tai EAP-hylkäykseen, joilla käyttäjä joko päästetään verkkoon tai estetään hänen verkkoon pääsynsä [11]. Kuten aikaisemmin mainittiin, EAP-protokollia on monia erilaisia kuten EAP-TLS joka käyttää autentikoinnissa PKI (*Public Key Infrastructure*)-digitaalisia varmenteita sekä PEAP:a (*Protected EAP*), joka on EAP-TLS:n sovellus. EAP-TLS:ssä sekä palvelimella että käyttäjällä pitää olla varmenne, mutta PEAPissä ainoastaan palvelin tarvitsee varmenteen. PEAP-protokollassa käytetään hyväksi tunnelointia ja autentikointi tapahtuu kahdessa vaiheessa. Aluksi palvelimen päässä olevan varmenteen avulla vaihdetaan salattuja avaimia, joiden avulla luodaan tunneli. Tämän jälkeen asiakaslaite suorittaa varmennetun toisen kättelyn palvelimen kanssa (Avainten vaihto tapahtuu SSL:llä, *Secure Socket Layer*, joka käydään läpi kappaleessa 4.9). Toinen autentikointi tehdään toisella EAP-keskustelulla. PEAPin etuna on se, että se suojaa käyttäjän tietoja lähettämällä ne toista tunnelia pitkin. [7.]

EAPin ongelmana on se, ettei EAP-protokollilla ole yhtenäistä standardia käyttäjän tunnistukseen ja istunnossa tarvittavien avainten kuljetukseen vaan jokainen standardi toteutetaan omalla tavallaan. Yleensä istunnon avaimet kuljetetaan jonkin suojaus protokollan avulla esim. TLS:n (*Transport Layer Security*). Edellä mainittu PEAP-protokolla käyttää juuri TLS-tunnelia avainten välittämiseen, jolloin EAP-protokolla suoritetaan TLS-tunnelin sisällä. Toinen vaihtoehtoinen EAP-suojaus on PIC (*Pre-IKE Credential Provisioning Protocol*), jossa käytetään internetin avaintenvaihto-protokollaa (IKE, *Internet Key Exchange*) ja Layer 3:n (verkkokerroksen) tunnelointia. PIC-protokolla tunnistaa verkon käyttäjälle, sekä tarjoaa yhtenäisen tavan valtuutusten siirrolle.

Kummallakin näistä suojauksista ovat haavoittuvia silloin kuin niitä käytetään muiden EAP-protokollien kanssa. Ongelmana ovat passiiviset salakuunteluyritykset sekä man-in-the-middle-hyökkäykset. Tunnelointi tarjoaa käyttäjän henkilöllisyydelle jonkinlaisen suojan passiivisten salakuunteluhyökkäysten varalta, mutta väliaikainen henkilöllisyys, jota hallitaan kotiverkossa, suojaisi sekä passiivisia että aktiivisia hyökkäyksiä vastaan. Man-in-the-middle-hyökkäyksissä tunnelointiprotokollat eivät pysty suojaamaan käyttäjän henkilöllisyyttä ja hyökkääjä pystyy esim. kaappaamaan verkkoyhteyden. Lisäksi mikäli EAP-SIM-protokollaa käytetään suojaukseen tunneloinnissa, se olisi heikompi kuin jos käytettäisiin pelkästään EAP-SIM:ä ilman tunnelointia. Ratkaisuna näihin olisi luoda salattu sitominen tunnelointiprotokollan ja käyttäjäntunnistusprotokollan välille. [9.]

### 4.3 Digitaalinen henkilöllisyys

Digitaalinen henkilöllisyys koostuu tiedoista, jotka kuvaavat yksityisen henkilön tai asian, niin ettei häntä voi sekoittaa toiseen henkilöön tai asiaan. Lisäksi digitaalinen henkilöllisyys kertoo kohteen suhteista muihin tahoihin. Digitaalisen henkilöllisyyden valvomiseen kuuluu henkilöllisyyden luonti, valvominen, käyttö ja lopuksi tuhoaminen, kun niillä ei enää ole käyttöä. Yksi digitaalisen henkilöllisyyden tärkeimmistä tehtävistä on valtuuttaa tietyt toimitteet kuten vaikka valtuutuksen antaminen jonkin tiedoston muokkaamiseen. Yksi digitaalisen henkilöllisyyden tärkeimmistä asioista on luottamus. Loppujen lopuksi jokainen valtuutuksen anto, joka tehdään digitaalisen henkilöllisyyden avulla, vaatii luottamusta siihen että käyttäjän henkilöllisyys on oikea. Digitaalista henkilöllisyyttä käytettäessä luottamus esiintyy monissa paikoissa. Luotetaan esim. siihen, että taho, jonka kanssa asioidaan, on juuri se taho kuka hän väittääkin olevansa tai että saadut henkilöllisyysvakuutukset on myönnetty oikean tahon toimesta. Digitaalisen henkilöllisyyden tapauksessa luottamus on yleensä yhdistetty tiettyihin asiakirjoihin, jotka vahvistavat henkilöllisyyden, sekä näihin asiakirjoihin sidottuihin ominaisuuksiin (attributes). Toinen tärkeä asia on yksityisyys eli digitaalisten henkilöllisyyksien täytyy olla hyvin suojattu. Seuraavissa kappaleissa käydään läpi turvallisuustekniikoita (mm. PKI, XML ja SOAP), joita käytetään myös digitaalisessa henkilöllisyydessä. [10, s. 8 – 9, 15, 17, 21.]

### *4.3.1 Nimet ja tiedostot*

Tunnistuksesta puhuttaessa nimet ovat yksi ensimmäisistä asioista, joilla tunnistus voidaan tehdä. Nimi onkin yksi yleisimmistä tiedoista, jotka henkilöllisyystietoihin tallennetaan. Nimien avulla pystytään viittaamaan monimutkaisiin kokonaisuuksiin lyhyen merkkijonon avulla. Nimiavaruudeksi kutsutaan perusjoukkoa (universe), jossa nimi on varmasti ainutlaatuinen ja jossa määritellään, mitä nimi tarkoittaa. Tästä syystä nimiavaruutta kutsutaan joskus domainiksi. URI (*Uniform Resource Intication*) on maailmanlaajuinen nimiavaruus, ja se on yksi verkon tärkeimmistä ominaisuuksista. URI on URL:n (Uniform Resource Locator) yleisempi versio, ja se tunnistaa uniikin paikan (esim. verkkosivun) sijainnin verkossa. URL puolestaan on verkkosivun osoite. URL kertoo jonkin sivun tai muun olemassa olevan resurssin sijainnin verkossa, kun taas URI:a voidaan käyttää nimeämään asioita yksittäisellä globaalilla merkkijonolla, vaikka nimelle ei olisikaan määritelty sijaintia verkossa. URL:n ja URI:n rakenne on kuitenkin sama, jolloin URI:t toimivat usein myös URL:na. Ilman URI:a monet verkon käytössä itsestään selvänä pidettävät asiat eivät toimisi. URI:n avulla voidaan esim. viitata dokumentissa toiseen dokumenttiin ilman että dokumenttien tekijöiden tarvitsee sopia käytettävästä ohjelmistosta tai palvelimesta. URI liittää myös verkon ulkopuoliset resurssit kuten tietolähde kirjastojen tiedostot osaksi maailmanlaajuista nimiavaruutta sekä varmistaam, että ne voidaan erottaa muista lähteistä (eli että niillä on uniikki nimi). URL ja URI koostuvat kolmesta osasta:

- Protokollan tunnistuksesta jonka perässä on kaksoispiste esim. http:.
- Domainin nimi ilmoittaa uniikin domainin verkossa esim. www.esimerkki.fi.
- Polku komponentti ilmoittaa mitä erityisiä (specific) resursseja domainissa täytyy tunnistaa esim. /llp?ln=esimerkki&lang=fin.

Yhdessä nämä kolme osaa muodostavat domainin URI http://www.esimerkki.fi/llp?ln=esimerkki&lang=fin. Yksi URI:n tärkeimmistä asioista on se, ettei se saisi koskaan vaihtua, joten URI kannattaa valita tarkoin.

Nimen lisäksi hakemistot ovat yksi tärkein osa tiedostojenhallintaa ja siten myös henkilöllisyyden hallintaa. Tiedostojenhallinta on täynnä erilaisia hakemistoja ja useilla järjestelmillä on useita eri hakemistoja osoitekirjoille, salasanatiedostoille jne. Hakemistopalvelu on verkkotietoinen (network-aware) hakemisto, joka mahdollistaa hakemiston keskitetyn hallinnan ja samanaikaisesti hakemiston tietojen jakamisen eri puolilla verkkoa oleville ohjelmille. Hakemistopalvelu sisältää rakenteisen säilytyspaikan tiedoille, joilla on usein monimutkaisia keskinäisiä suhteita. Rakenne on määritelty sisältä käsin kaavion avulla. Kaavio määrittää myös, mitä ominaisuuksia merkintään (entry) voidaan yhdistää, sallitut ulkoasut tai ominaisuuden tyypin sekä sen, onko merkintä pakollinen vai vapaaehtoinen. Jokainen merkintä määritellään objektiksi hakemistoon ja objekti sisältää merkintään liitetyt ominaisuudet. Hakemistot ovat yleensä hierarkkisia, ja hakemiston hierarkkinen rakenne tallennetaan hakemistopuuhun. Hakemisto-objekteja jotka ovat puun solmukohdassa, nimitetään säiliöobjekteiksi (container objects). Säiliöobjektit voivat sisältää toisia säiliöobjekteja, lehtiä (leaves, tarkoittaa päätepistettä esim. tulostin, ihminen tai toimisto) tai pääteobjekteja. Hakemistopalvelu tarjoaa myös mahdollisuuden etsiä tiedostoja ja valvoa merkintöjä. Erilaisia hakemistopalveluja on satoja, mutta muutamat niistä ovat levinneet melko laajalti. Seuraavaksi käydään pikaisesti läpi neljä erilaista laajalti levinnyttä hakemistopalvelua. [10, s. 73 – 81.]

*Domain Name System*

DNS (*Domain Name System*) on hakemisto, joka kartoittaa domainien nimet IP-osoitteisiin. DNS on rakennettu hierarkkiseen domain nimiavaruuteen, ja se on hajautettu hakemisto, joka mahdollistaa yksittäisen, globaalin domain nimien hakemiston arkkitehtuurin. Hakemisto rakennetaan tuhansista palvelimista, joita eri yritykset omaistavat. DNS:n avulla eri yritysten palvelimet voivat yhteistyössä hoitaa tehokkaasti domainien nimien kartoituksen IP-osoitteisiin. [10, s. 81.]

*RMIRegistery*

RMI eli Java Registery Method Invocation tarjoaa pohjan asiakas-palvelin arkkitehtuurille verkon yli. RMIRegistry on RMI hakemisto, joka tarjoaa hakemiston nimetyille viittauksille etäisiin objekteihin (remote object) ohjelmistoympäristön sisällä.

Kun palvelin käynnistyy, se rekisteröityy RMiRegisteryyn antamalla nimensä ja viittauksen palvelimen metodiin. Myöhemmin käyttäjän kirjautuessa metodiin, palvelin ei käytä tiettyä viittausta metodiin, vaan pyytää RMIRegisteryä palauttamaan viitteen palvelusta, jolla on tietty nimi (rekisteröityessä annettu nimi). Kun RMIRegistery palauttaa viittauksen, käyttäjä voi sen avulla yhdistää itsensä palvelimeen ja herättää (invoke) metodin. Nimettyjen hakemistojen etäviittausten (remote reference) käytöllä on muutamia etuja. Ensinnäkin käyttäjän ei tarvitse olla tietoinen toteutuksesta tai levittämiseen (deployment) liittyvistä tiedoista. Lisäksi nimen luomaa epäsuoruutta (indirection) voidaan käyttää skaalaamaan palveluja, koska useampi kopio palvelimesta voi olla palveleva asiakasohjelma. [10, s. 82.]

*X.500*

X.500 on toiminut pohjana useille muille tiedostojärjestelmille. X.500 on itse asiassa määritelmäperhe, johon kuuluu mm. X.509-tunnistuskehysmääritelmä, jota käytetään julkisen avaimen järjestelmässä. X.500 määrittää hierarkisen hakemistopalvelun, joka toimii normaalissa nimiavaruudessa. palvelu on suunniteltu hyvin skaalautuvaksi ja laajaksi. X.500:a käsitellään hieman tarkemmin julkisen avaimen järjestelmän yhteydessä. X.500-standardissa on niin paljon vaihtoehtoja määrittelyjen rakentamiseen, että se on melko monimutkainen käyttäjän näkökulmasta. [10, s. 83.]

*LDAP*

LDAP eli lightweight directory access protocol luotiin alun perin tarjoamaan käyttäjille yksinkertaisempaa pääsyä joihinkin X.500:n tarjoamiin toimintoihin. LDAP määrittää myös erityisen rajapinnan (API, *Aplication Program Interface*) asiakkaille, mitä X.500 ei tee. API-rajapinta mahdollistaa ohjelmistojen kehityspakkausten luonnin. Kehityspakkaukset sisältävät hakemistopalveluissa käytettävää koodia. LDAP on kehittynyt X.500-yhdyskäytävästä omaksi toimivaksi hakemisto palveluksi ja nykyään useat kaupalliset hakemistot tukevat LDAP:a. [10, s. 84.]

### *4.3.2 Yhteenkootut tiedostotiedot*

Kertakirjautumisesta on tullut monille yrityksille tärkeä osa henkilöllisyyden valvontaa. Hajallaan olevat henkilötiedot aiheuttavat ongelmia käyttäjille ja kaupankäynnille. Henkilötietojen yhteen kokoaminen ja eri henkilöllisyystietojen välisten vuorovaikutussuhteiden löytäminen on tärkeää sekä asiakkaan että palveluntarjoajan kannalta. Henkilötietojen kokoamiseksi yrityksillä on neljä eri vaihtoehtoa:

- Kootaan kaikki tiedot yhteen säilytyspaikkaan.
- Luodaan metahakemisto, joka synkronisoi muista henkilötiedostovarastoista (identity data store) saadut tiedot.
- Luodaan virtuaalinen hakemisto, joka tarjoaa yhden integroidun näkymän muista henkilötietovarastoista.
- Liittää yhteen eri yritysten hakemistoja sitomalla henkilötietohakemistoja yhteen (Federaatio). [10, s. 86 – 87.]

## 4.4 Hyvän tunnistusjärjestelmän ominaisuuksia

Tunnistusjärjestelmillä on monia tärkeitä ominaisuuksia, joita niiden tulisi pystyä täyttämään. Tunnistusjärjestelmien tulisi olla mm. käytännöllinen, riittävän turvallinen, sen tulisi olla yhteensopiva useiden protokollien kanssa, sen tulisi tukea liikkuvuuden hallintaa (mobiili-IP) sekä olla luotettava. Lisäksi järjestelmän valvojan tulisi voida lisätä, poistaa ja muuttaa käyttäjien tietoja mahdollisimmin vaivattomasti. Pääsynhallinta on prosessi, jossa käyttäjille myönnetään pääsy tiettyihin palveluihin ja kielletään pääsy toisiin palveluihin.

Pääsynhallintaa tarvitaan lähes aina silloin, kun asiakkaille annetaan pääsy johonkin palveluun. Esimerkkinä pääsynhallinasta voidaan käyttää sähköpostia, jossa pääsynhallinta myöntää käyttäjälle oikeuden käyttää omaa sähköpostiaan ja toisaalta kieltää muiden sähköpostien käytön.

Pääsynhallinta on lähinnä poliittinen (policy) kysymys ja pääsynhallinta on suunniteltu vahvistamaan yhtiön tai ryhmän valitsemaa linjaa. Pääsynhallintapolitiikassa tärkein kysymys on se, kenellä on vastuu politiikan toimivuudesta. Vastuullisuus jaetaan usein kolmeen kategoriaan: omistajiin, valvojiin ja käyttäjiin. Omistaja voi olla esim. tarjottavan resurssin luoja tai resurssia tarjoavan organisaation pääjohtaja. Omistajalla on aina päätösvalta ja perimmäinen vastuu tarjotusta resurssista, joten pääsynhallinta on tehokkaampaa silloin kuin omistaja on selvästi määritelty. Valvojat ovat henkilöitä, jotka valvovat ja hoitavat resurssia päivästä toiseen. Valvojien vastuulla on pääsynhallintapolitiikkojen toimeenpano sekä se, että resurssi on tarjolla sallituille käyttäjille. Valvojilla on näin myös tärkeä osa pääsynhallinnassa, koska he valvovat ja kehittävät pääsynhallintapolitiikat, joissa käyttäjät määritellään. Käyttäjä on henkilö, ryhmä, yritys, ohjelmisto tai jokin muu taho, joka käyttää tarjottua resurssia. Käyttäjät voivat olla vastuussa siitä, että resurssi on suojattu silloin kun he käyttävät sitä, jolloin käyttäjästä tulee hetkellisesti valvoja. Usein pääsynhallintajärjestelmät eivät pysty suojaamaan tarjottavaa resurssia täydellisesti, jolloin käyttäjän vastuulla on toimia omistajan resurssin käytöstä antamien ohjeiden mukaan.

Yksi pääsynhallinnan perusajatuksista on se, että käyttäjille ei pidä antaa enempää resursseja kuin he tarvitsevat haluamansa tehtävän suorittamiseen. Käytännössä se kuitenkin on yleensä ongelmallista, koska johonkin tiettyyn tehtävään tarvittavat resurssit saattavat vaihtua ajan myötä ja henkilö saattaa tarvita eritasoisia valtuuksia riippuen senhetkisistä tehtävistään. Lisäksi mahdollisimman tarkkojen käyttäjäoikeuksien tekeminen vaatii tarkempaa lupaa, koska jokainen käyttäjän mahdollisesti tarvitsema resurssi pitäisi erikseen määritellä sallituksi. Koska mahdollisimman pienien käyttöoikeuksien tekeminen on melko työlästä, on vaihtoehto kirjanpito, joka on huomattavasti helpompi, halvempi ja nopeampi toteuttaa.

Kirjanpidon avulla valvotaan käyttäjien toimia ja vaikka sillä ei pystytä estämään rikkeitä, ne kuitenkin huomataan kirjanpidon avulla, jolloin samanlaiset rikkeet voidaan jatkossa estää korjaamalla rikkeen aiheuttanut ongelma (esim. kieltämällä rikkeen tehneen henkilön pääsy resurssiin).

Käytännöllisen pääsynhallinta järjestelmän täytyy jollain tasolla luottaa siihen, että valvojat ja käyttäjät noudattavan annettuja politiikkoja. [10, s.59 – 67.]

### 4.5 Tunnistusmenetelmät mobiilipalveluissa

Yleisimmin käyttäjän kirjautuessa johonkin järjestelmään hän kirjautuu sisään käyttäen omaa käyttäjänimeään sekä salasanaa. Salasana ja käyttäjänimen hyvänä puolena on sen yksinkertaisuus ja se, että se on kaikille tuttu kirjautumiskeino. Salasanan ja käyttäjänimen käytöllä on kuitenkin huonojakin puolia. Ensinnäkin ihminen muistaa luotettavasti vain rajallisen määrän salasanoja (n. kahdeksan). Parhaita salasanoja ovat pitkät ja satunnaiset merkkijonot, mutta ihmiset eivät muista sellaisia, joten käytännössä salasanat ovat lyhyitä ja helposti muistettavia. Lyhyet helposti muistettavat salasanat ovat myös helposti murrettavissa, koska hyökkääjän on helppo keksiä ne. Lisäksi ihmiset kirjoittavat salasanoja muistiin, joko paperille ja koneelle ja ne saattavat sitä kautta joutua vääriin käsiin. Ihmisiä ja myös koneita voidaan myös huijata antamaan oma salasana esim. esiintymällä järjestelmän valvojana tai vale kirjautumisikkunan avulla. Koska pelkkä salasana on helppo murtaa, kutsutaan pelkän salasanan käyttöä heikoksi tunnistamiseksi. [10, s. 53.]

Vahvassa tunnistamisessa käytetään vähintään kahta tunnistuskeinoa esim. salasanaa ja SIM-korttia. Kaiken kaikkiaan ihmisen tunnistuskeinot voidaan jakaa kolmeen eri ryhmään.

- Käyttäjän hallussa olevat, tunnistuksessa käytettävät esineet: esim. eID (*electronic personal Identification Card*) –kortti, kännykän SIM-kortti tai autentikointitunniste (authentication token).
- Käyttäjän tietämä tunniste esim. salasana, PIN (*Personal Identification Number*) -koodi tai henkilökohtainen tunnusluku.
- Ihmisen olemukseen perustuva tunnistus esim. biometrinen tunnistus, käsialan tai äänen tunnistus. [11.]

Kahden tekijän tunnistus (Two-factor Authentication) eroaa vahvasta tunnisteesta siinä, että siinä on käytettävä kahta edellä mainituista kolmesta ryhmästä, kun taas vahvaan tunnistukseen riittää kahden saman ryhmän tunnistusmetodin käyttö (esim. Useamman kysymyksen kautta tehty tunnistus lasketaan vahvaksi tunnistukseksi, mutta ei kahden tekijän tunnistukseksi) [11]. Seuraavissa luvuissa käsitellään mobiiliverkon yleisempiä tunnistustapoja.

### *4.5.1 Haaste-vastausjärjestelmä*

Haaste-vastaus -järjestelmässä palvelin generoi satunnaisen merkkijonon ja käyttäjän täytyy tiettyjen hänelle ennalta annettujen algoritmien mukaan muokata merkkijonoa ja lähettää se takaisin palvelimelle. Algoritmit voivat sisältää myös salaisten avainten käytön merkkijonon muokkauksessa. Yleensä merkkijonojen muokkaukset tapahtuvat koneilla, jotta algoritmit voidaan tehdä monimutkaisemmiksi ja siten vaikeammiksi arvata. Haaste-vastausjärjestelmän etu salasanaan verrattuna on se että algoritmeista voidaan tehdä kohtuullisen monimutkaisia. Huonona puolena haaste-vastaus–järjestelmässä on se, että algoritmit pitää koodata juuri tätä tarkoitusta varten olevalla laitteelle tai käyttäjän koneelle. Lisäksi algoritmit voivat hävitä tai ne voidaan varastaa samalla tavalla kuin salasanatkin.

Haaste-vastausjärjestelmästä on myös muunnelma, jossa käyttäjälle annetaan erikoistarkoitukseen tehty laskenta laite, jota kutsutaan tunnisteeksi (token), joka laskee sarjan näennäissatunnais- (pseudorandom) lukuja. Sekä tunnisteen että tunnistuksesta huolehtivan palvelimen kello on tarkasti synkronoitu samaan aikaan, ja tunniste näyttää palvelimelle uuden näennäissatunnaisluvun tietyin väliajoin (vaikkapa joka 60:s sekunti). Haaste-vastaus järjestelmässä voidaan myös käyttää tunnistustapaa, jossa asiakas lähettää vastaukseksi oman käyttäjänimensä, salasanansa ja tunnisteelta saamansa numeron. Tällöin palvelin hakee käyttäjänimen perusteella oikean numeron ja salasanan ja toteaa ne oikeiksi.

Tässä järjestelmässä palvelin ei lähetä haastetta, mutta sekä tunniste että palvelin käyttävät ovat tarkasti synkronoitu samaan aikaan ja kummatkin käyttävät monimutkaista algoritmia laskemaan yhteisen salaisen vastauksen.

Näennäissatunnaislukutunnistus lasketaan kahden tekijän tunnistukseksi, koska siinä käytetään sekä jotain, mitä henkilöllä on (tunniste), että jotain, minkä henkilö tietää (algoritmi tai salasana). Tällöin mahdollisen hyökkääjän pitää tietää kummatkin osat (tunnisteen ja algoritmin/salasanan), ennen kuin hän voi murtautua järjestelmään. Haaste-vastausjärjestelmä voidaan myös yhdistää käyttäjätunnukseen ja salasanaan niin, että salasanaa ei tarvitse siirtää verkon yli. Silloin sekä palvelin että käyttäjä tietävät salasanan ja palvelin generoi haastelauseen ja lähettää sen käyttäjälle. Käyttäjä luo viestitiivisteen (message digest) käyttämällä salasanaa haastelauseeseen ja lähettää saamansa tiivisteen takaisin palvelimelle. Palvelin käyttää samaa salasanaa haastelauseeseen saadakseen oman tiivisteen ja vertaa sitten omaa tiivistettään ja käyttäjältä saamaansa tiivistettä toisiinsa. Kirjautuminen onnistuu, jos tiivisteet ovat samat. [10, s. 54 – 55.]

### *4.5.2 Tunnistus älykortin avulla*

Älykortilla tarkoitetaan muovikorttia, johon on integroitu mikropiiri, joka tallentaa tietoa kortin haltijasta ja käyttötarkoituksesta. Älykortit tunnetaan kahdesta pääpiirteestä: erittäin hyvät tietoturvaominaisuudet ja kyky toimia kannettavana sovellusalustana mitä monipuolisimmissa käyttötarkoituksissa. Älykortti on salaisen tiedon turvallinen säilytyspaikka. Älykortteja ovat mm. SIM-kortti, maksukortti, sähköinen henkilökortti tai eri korttien yhdistelmä. Kortille voidaan tallentaa esimerkiksi henkilötietoja, rahaa tai muuta sellaista tietoa, jonka muuttuminen tai paljastuminen olisi vaarallista. Lisäksi kortille voidaan tallentaa salausavaimia, joita kortin käyttäjä voi käyttää esimerkiksi avaimen vaihtoon, verkossa tunnistamiseen tai digitaaliseen allekirjoitukseen. [12.]

Älykortin avulla tapahtuva tunnistus perustuu julkisen avaimen menetelmään (kappale 4.7). Tunnistus tapahtuu käyttäjän henkilökohtaisella älykortilla sekä vain käyttäjän tietämällä PIN-koodilla, joka on tallennettu älykortille.

PIN-koodin lisäksi älykortille on tallennettu käyttäjän digitaalisia varmenteita sekä käyttäjän yksityiset avaimet. Älykortilla tunnistaminen tapahtuu siten, että käyttäjä asettaa älykortin tietokoneeseen asennettuun lukulaitteeseen ja syöttää PIN-koodin joko tietokoneen tai lukulaitteen näppäimistöltä.

Mikäli syötetty arvo täsmää kortille tallennetun PIN-koodin kanssa, älykortti aktivoituu siten, että lukulaite pystyy lukemaan kortin julkisia tietoja. Tämän jälkeen lukulaite hakee kortilta käyttäjän varmenteen ja lähettää sen tunnistusta pyytävälle palvelulle. Palvelu varmistaa käyttäjän varmenteen oikeellisuuden ja myöntää käyttöoikeuden. Varmenne tarkistetaan varmentajan digitaalisesta allekirjoituksesta (kts. kappale 4.6). [13.]

Älykorteilla voidaan myös toteuttaa haaste-vastausjärjestelmä. Älykortilla toteutetussa haaste-vastausjärjestelmässä haasteen muokkaamiseen käytettävät algoritmit on tallennettu älykortille. Joissain älykorteissa on myös sisäänrakennettu sormenjälkiskanneri, jonka avulla voidaan tehdä käyttäjän biometrinen tunnistus. Tällä biometrisellä tunnisteella pystytään varmistamaan, että kortin käyttäjä on myös kortin oikea omistaja. Älykorteissa olevat sormenjälkitunnistimet mahdollistavat kaikkien kolmen eri tunnistuskäytännön yhtäaikaisen käytön eli kolmen tekijän tunnistuksen (three-factor authentiation), jossa käytettäisiin sekä sormenjälkeä (ihmisen olemukseen perustuva tunnistus), älykorttia (käyttäjän hallussa oleva tunniste) että salasanaa (käyttäjän tietämä tunniste). Todennäköisesti yleisin älykorteilla toteutettu tunnistusjärjestelmä on matkapuhelimien käyttämä SIM-tunnistus. SIM-kortilla ovat käyttäjän tiedot, joten kun SIM-kortti siirretään puhelimesta toiseen, niin mukana siirtyvät käyttäjän tiedot, kuten puhelinnumero, tiedot verkosta sekä useimmiten myös osoite- ja yhteydenottotiedot. Suurin ongelma älykorteilla toteutetuissa tunnistuksissa on se, että älykorttien käyttö vaatii erillisen kortinlukijan, jota täytyy myös huoltaa ja valvoa asennuksen jälkeen. [10, s. 58 – 59.]

### *4.5.3 SIM-kortilla tapahtuva tunnistus*

SIM-kortilla tapahtuva tunnistaminen voi tapahtua joko kortinlukijalla, USB (*Universal Serial Bus*) -liittimellä tai suoraan kännykästä. Tunnistusasetuksista riippuen kirjautumisen yhteydestä saatetaan kysyä myös SIM-kortin PIN-koodia. GSM-tunnistus perustuu haaste-vastausmekanismiin.

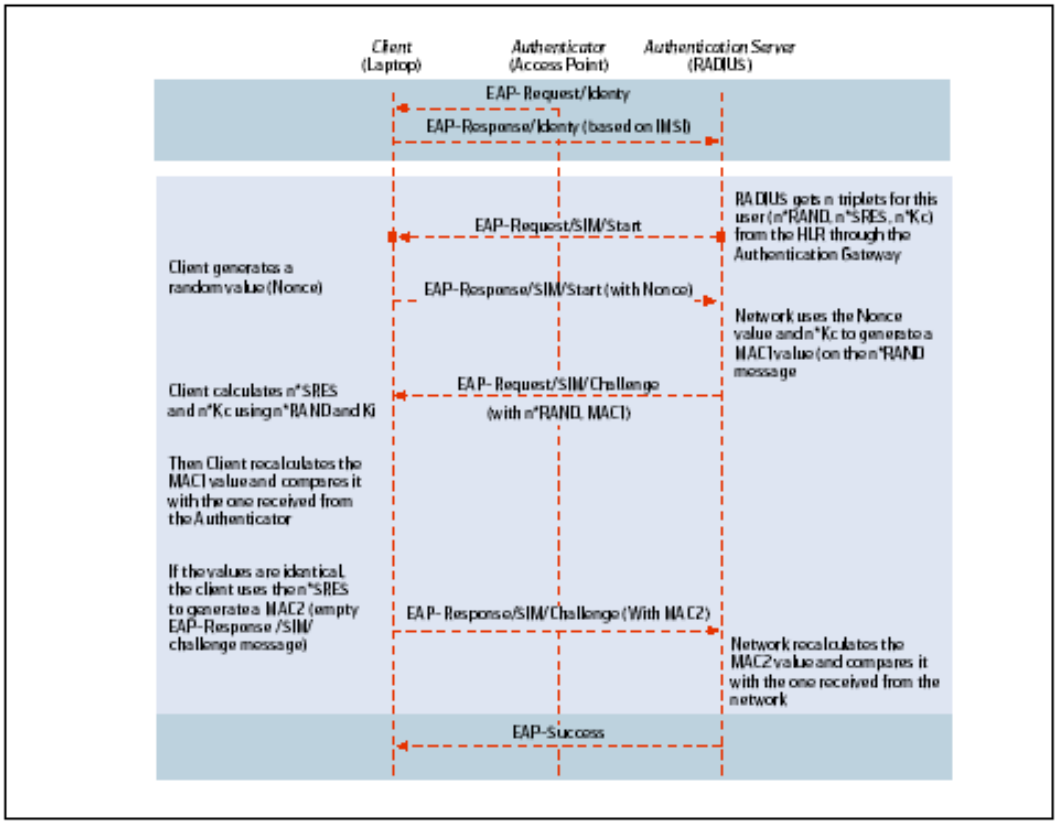
SIM-kortin käyttämä tunnistusmetodi muodostaa 128-bittinen satunnaisluvun (RAND), joka muodostetaan tunnistuksen yhteydessä haasteeksi, RAND ja SIM-kortilla oleva salausavain (Ki) lähetetään tunnistuspalvelimelle, joka tuottaa 32-bittisen vastauksen (SRES) ja 64-bittisen avaimen (Kc).

Kuvassa 4 näkyy esimerkki siitä, mitä EAP (*Extensible Authentication Protocol*) -SIM tunnistuksen aikana tapahtuu. Tunnistustapahtuma alkaa, kun tunnistaja (Authenticator) lähettää havaitsemalleen käyttäjälle käyttäjätunnistuspyynnön (EAP-Request/Identity). Käyttäjäympäristön ohjelmistoprosessia, joka suorittaa EAP-SIM neuvottelun, kutsutaan anomiseksi. Anomisprosessin vastaus tunnistuspalvelimelle sisältää joko käyttäjän IMSI:n (International Mobile Subscriber Identity) tai väliaikaisen tunnisteen. Tämän jälkeen tunnistaja toimii ainoastaan relay-agenttina, joka kuljettaa viestejä käyttäjän anomisprosessin ja Tunnistuspalvelimen välillä. Seuraavaksi käyttäjä saa EAP-SIM/aloituspyynnön, (EAP-Request/SIM/Start), johon käyttäjä lähettää vastaukseksi viestin, jossa on käyttäjän anomisprosessin valitsema satunnainen numero (Nonce). Palvelin saa käyttäjän GSM-tripletin (RAND, SRES ja Kc) käyttäjän GSM-verkon operaattorin AuC:lta (*Authentication Centre*). Saaduista tiedoista (triplets, Nonce, käyttäjän henkilöllisyydestä ja EAP-versiosta) palvelin johtaa käyttäjälle seuraavat avaimet:

- K_aut eli tunnistusavaimen
- K_encr eli salausavaimen
- myöhemmin saatetaan myös lähettää yleisavain (master key) tai muita ohjelmakohtaisia avaimia, jos niitä tarvitaan.

Tunnistusavainta (K_aut) käytetään laskemaan MAC (Message Authentication code), jota käytetään seuraavissa EAP-viesteissä. Salausavainta (K_encr) käytetään puolestaan salaamaan ENCR_DATA-ominaisuudet (attributes). Tämä salaus käyttää myös aloitusvektoria (*Initialization Vector*, IV), joka on pakollinen osa kaikissa EAP-viesteissä, joissa salausta käytetään. Yleisavainta voidaan puolestaan suojausasetuksista riippuen käyttää suojaamaan radiotietä.

Kun avaimet on luotu, tunnistuspalvelin voi lähettää käyttäjälle EAP Request/SIM Challenge -viestin, johon sisältyy GSM-tripletin RAND (jota käytetään MAC1:n laskemiseen), salattu uusi käyttäjätunnus sekä MAC1, johon sisältyy NONCE (aikaisemmin lähetetty satunnainen numero). Kun käyttäjän anomisprosessi saa tähän haasteviestin, se ajaa GSM-algoritmin saadakseen GSM-tripletin, sitten prosessi johtaa avaimet samalla tavalla kuin palvelimella tehtiin aikaisemmin sekä laskee MAC1:n ja vertaa sitä palvelimen laskemaan MAC1:n. Jos MAC1:t täsmäävät, niin verkko on tunnistettu siksi verkoksi, jolle GSM-tripletti sekä käyttäjän luoman Nonce on lähetetty. Tällöin anomisprosessi laskee MAC2-arvon ja lähettää EAP Responce SIM/Challenge-viestin, joka sisältää MAC2-arvon, joka puolestaan sisältää käyttäjän SRES-vastauksen (jolla MAC2 lasketaan) arvon. Tunnistuspalvelin laskee MAC2:n ja tarkistaa, että saatu MAC2 ja palvelimen laskema MAC2 täsmäävät ja lähettää käyttäjälle EAP-Success-viestin, joka kertoo tunnistuksen onnistuneen. [14.]




*Kuva 4: SIM-kortilla tapahtuva tunnistus [14]*

### *4.5.4 Mobiilikansalaisvarmenne*

Mobiilikansalaisvarmenteella tarkoitetaan väestörekisterikeskuksen henkilöille myöntämän SATU:n eli sähköisen asiointitunnuksen liittämistä sirukorteille esim. SIM-kortille, pankkikortille tai sähköiseen henkilökorttiin. Mobiilivarmenne syntyi kuin HST (Henkilön Sähköinen tunnistus ryhmä) kokosi useita eri tunnistusteknologioita (mm. SIM-tunnistus, pankkikorttitunnistus ja sähköisellä henkilökortilla tapahtuva tunnistus) ja määritteli niille yhteisen arkkitehtuurin ja ekosysteemin, joiden pohjalta operaattorit ovat tuomassa Väestörekisterikeskuksen kanssa mobiilikansalaisvarmennepalveluja markkinoille.

Mobiilikansalaisvarmenteen avulla teleoperaattorit ovat pystyneet toteuttamaan sähköisen henkilön tunnistamisen ja siihen liittyvän varmentamisen SIM-kortin avulla. Mobiilivarmenteen käyttöönotto edellyttää kansalaisilta uuden SIM-kortin hankkimista operaattorilta. Uuteen SIM-korttiin on asennettu asiointitunnus, joka tulee rekisteröidä poliisiasemalla. Mobiilikansalaisvarmenne perustuu PKI (Public Key Infrastructure) -teknologian käyttöön. Varmenteella varustettu matkapuhelin on tunnistautumis- ja allekirjoitusväline sähköisessä asioinnissa ja kaupankäynnissä. Käyttäjän tunnistaminen ja sähköinen allekirjoitus voidaan tehdä matkapuhelimella, vaikka asiointipalvelua käytettäisiin tietokoneen internetselaimella. Siksi se on vaihtoehto tietokoneen kortinlukijalle. Varmennus voidaan tehdä myös puhelun aikana vaikkapa puhelinasiakaspalveluun. [15.]

## 4.6 Sähköinen allekirjoitus

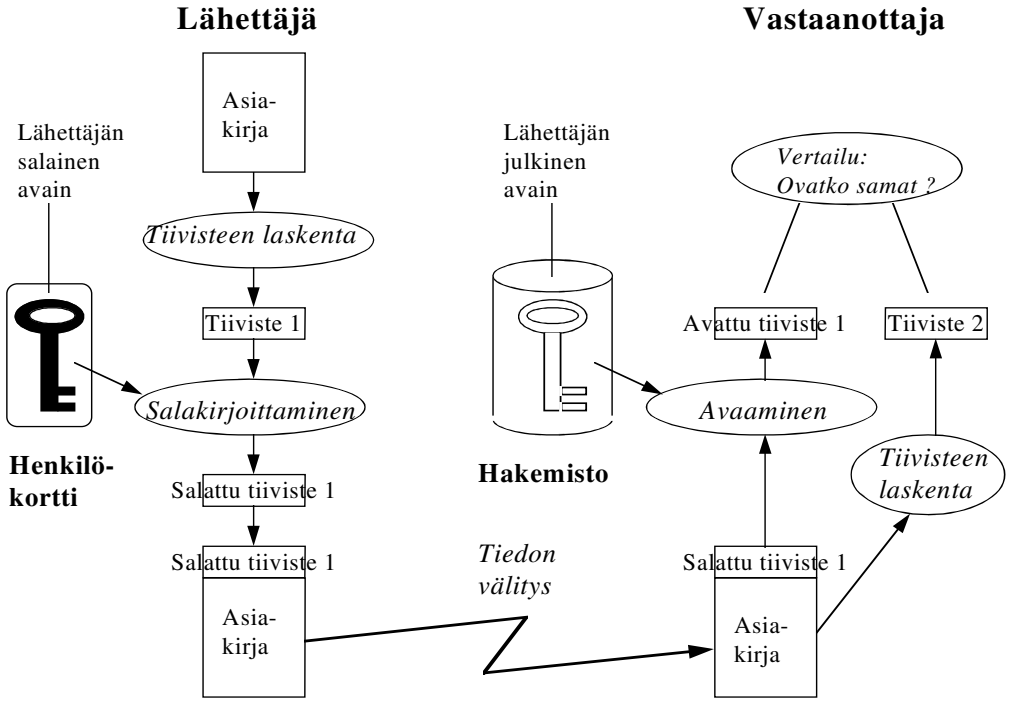
Jotta elektroninen tiedonsiirto olisi luotettavaa, täytyy viestin välittäjän ja vastaanottajan välillä olla luottamus viestin muuttumattomuudesta, sekä varma tieto siitä, että viestii oikean henkilön kanssa. Lisäksi täytyy olla varmuus siitä, että elektronisesti tehty sopimus on pitävä. Tämä toteutetaan sähköisellä allekirjoituksella ja julkisen avaimen järjestelmällä (kappale 4.7). Sähköisellä allekirjoituksella tarkoitetaan yleensä eri teknologioita, jotka sallivat henkilön tai koneen merkitä dokumentti elektronisesti.

Elektronisen merkinnän välityksellä dokumenttiin sisältyy tunnistusmekanismi. Sähköinen allekirjoitus ei salaa viestin sisältöä, vaan varmistaa sen koskemattomuuden. Sähköinen allekirjoitus on siis elektroninen tapahtuma, jolla tunnistetaan lähettäjä sekä varmistetaan dokumentin sisällön luotettavuus ja koskemattomuus. Allekirjoitus on myös todistusoikeudellinen funktio eli tiedon vastaanottajan tulee voida vedota allekirjoitukseen, mikäli myöhemmin syntyy ristiriita sopimusosapuolten välille. Sähköinen allekirjoitus voi olla esimerkiksi nimi sähköpostin vartalossa, digitaalikuva käsin kirjoitetusta allekirjoituksesta tai uniikki biometrinen tunniste kuten sormenjälki tai silmän verkkokalvokuva.

Sähköiseltä allekirjoitukselta vaaditaan seuraavia ominaisuuksia: tiedon eheyttä, kiistämättömyyttä sekä lähettäjän tunnistamista. Tiedon eheys eli muuttumattomuus varmistetaan esim. erilaisilla salausmetodeilla kuten salaisen avaimen menetelmällä, jossa sopimuksen osapuolilla on yksi salausavain, jolla sekä salataan että puretaan tiedosto tai julkisen avaimen menetelmä (kappale 4.7), jossa on eri avain salaamista ja purkamista varten. Tiedon kiistämättömyys tarkoittaa sitä, ettei kumpikaan osapuoli voi jälkeenpäin kiistää sähköisen viestinnän tapahtumista tai omia toimiaan. Tärkeää on myös viestin lähettämisen ajankohta. Luotettavin tapa lähetysajan todentamiseen on kolmannen osapuolen tarjoama aikaleimapalvelu. Tämä palvelu vahvistaa sähköisen viestin sisällön ja olemassaolon sekä esimerkiksi digitaalisen allekirjoituksen laatimisen ajankohdan. Aikaleimaan perustuvan näytön luotettavuus riippuu kolmannen osapuolen luotettavuudesta. Tiedon lähettäjän tai lähteen tunnistaminen eli autentikointi, liittyy puolestaan oleellisena osana informaation suojaamiseen. Yleisimmin tunnistustapana käytetään salasanaa, eli järjestelmä tunnistaa käyttäjän salasanan perusteella. [16.]

Sähköisen allekirjoituksen toimintaperiaate näkyy kuvasta 5. Koko dokumentin salaaminen salausavaimella on raskas ja aikaa vievä tapahtuma joten yleensä digitaalisessa allekirjoituksessa käytetään tiivistealgoritmia, jossa dokumentista lasketaan tiivistealgoritmi, joka salataan salausavaimella [17]. Salattu tiiviste toimii sähköisenä allekirjoituksena, ja se liitetään välitettävän aineiston mukaan.

Vastaanottaja purkaa tiivisteen lähettäjän julkisella avaimella ja laskee vastaanottamastaan aineistosta tiivisteen samalla hajautusalgoritmilla, jolla lähettäjä tiivisteen alun perin teki. Sen jälkeen vastaanottaja vertailee julkisella avaimella purettua ja itse laskemaansa tiivistettä toisiinsa. Mikäli tiivisteet ovat samat, voi vastaanottaja olla varma siitä, että aineisto on lähettäjän tekemä eikä se ole muuttunut matkalla. Jos koko viestin sisältö halutaan salata, täytyy silloin käyttää vastaanottajan julkista avainta salaamiseen. Tällöin viestin saa takaisin selväkieliseen muotoon vastaanottajan yksityisellä avaimella. [18.] Vaihtoehtona tiivistealgoritmille on se että julkisen avaimen järjestelmän avulla osapuolet neuvottelevat istunnon alussa salaisen avaimen, jota käytetään loppu istunnon ajan. Tällaista salausjärjestelmää kutsutaan hybridisalausjärjestelmäksi. Avaimenvaihtoalgoritmit sallivat kahden eri toistensa yleiset avaimet tietävän osapuolen laskea salaisen avaimen käyttäen toisen osapuolen salaista ja toisen yleistä avainta. Ongelmana on tilanne, jossa joku kolmas henkilö onnistuu pääsemään avaimenvaihdon väliin, jolloin kaikki viestit kulkevat hänen kauttaan. Tällaisen tapauksen varalta tarvitaan luotettavaa avaintenvaihto järjestelmää, joka toteutetaan digitaalisella varmenteella. Yleisin hybridiavainten järjestelmä on TLS (Transport Layer Protocol), joka tunnetaan yleisesti SSL:nä (*Security Socket Layer*), jonka toimintaa käydään läpi kappaleessa 4.9. [10, s. 37 - 38.]




*Kuva 5: Sähköisen allekirjoituksen periaate [18].*

Digitaalinen allekirjoitus on yksi sähköisen allekirjoituksen muotoja, jota käytetään julkisen avaimen järjestelmässä [16]. Digitaalisella allekirjoituksella pystytään varmistamaan viestin lähettäjän henkilöllisyys. Tämä tapahtuu digitaalisen varmenteen avulla. Digitaalinen varmenne voidaan myöntää henkilölle, yritykselle, ryhmälle tai verkkopalvelimelle [10, s. 42]. Digitaalinen varmenteen tulee sisältää vähintään lähettäjän henkilötiedot sekä julkisen avaimen, varmentajan yksilöivät tiedot, varmentajan digitaalinen allekirjoitus, varmenteen voimassaoloaika, varmenteen yksilöivä tunnus, sekä tieto varmenteen mahdollisista käyttörajoituksista.

Kuka tahansa pystyy myöntämään varmenteen OpenSSL:ä tai jotain muuta varmenteen ohjelmoimisrajapintaa, mutta on olemassa luotettuja kolmansia osapuolia (turvallisuusviranomainen tai tämän valtuuttamaa taho, Suomessa varmentajana toimii väestörekisterikeskus), jotka tarjoavat mm. julkisen avaimen rekisteröintiin, varmentamiseen ja jakeluun liittyviä palveluja [10, s. 44]. Tällaisia luotettuja kolmansia osapuolia kutsutaan CA:ksi (*Certificated Authority)*. Varmentajan julkisen avaimen on oltava laajasti tunnettu, etteivät huijarit voi väittää oman julkisen avaimensa olevan varmentajan avain. Vain tällöin digitaalinen allekirjoitus on luotettava. Kun lähettäjä haluaa varmenteen henkilöllisyydestään, hän osoittaa ensin varmentajalle henkilöllisyytensä esim. henkilöllisyystodistuksen tai sormenjäljen avulla, jonka jälkeen varmentaja luo varmenteen. Varmenteeseen sisältyy varmenteen pyytäjän henkilötiedot ja hänen julkinen avaimensa sekä varmentajan allekirjoitus. Varmenne muodostetaan varmentajan julkista avainta vastaavalla yksityisellä avaimella. Lähettäjä liittää tämän varmenteen digitaalisesti vastaanottajalle lähettämäänsä viestiin. Vastaanottaja avaa sen käyttämällä varmentajan yleistä avainta. Kun vastaanottaja avaa varmenteen, tietää hän myös viestin allekirjoittajan henkilöllisyyden varmenteen perusteella. Varmentajia voi olla useita, ja ne voivat muodostaa hierarkioita. Hierarkkisessa järjestelmässä ylempänä oleva varmentaja varmentaa hierarkiassa seuraavaksi alempien varmentajien luotettavuuden. [17.]

Varmenne on tietorakennelma (data structure), joten se on binäärimuodossa. Varmenteen useat käyttötarkoitukset vaativat varmenteen siirtoa paikasta toiseen verkon sisällä. Jotta varmenteen siirto verkossa olisi mahdollista, tietorakenne sarjoitetaan (serialized) käyttämällä DER (Distinguished Encoding Rules) -salausalgoritmia. Sarjoitetussa muodossa varmenne voidaan siirtää verkossa. [10, s. 44.]

### 4.7 Julkisen avaimen järjestelmä

Julkisen avaimen järjestelmä eli PKI (*Public Key Infrastructure*) on ohjelmistojen, salausteknologian ja palvelujen yhdistelmä. Julkisen avaimen salaus on epäsymmetrisen salauksen muoto, joka perustuu yksityisen ja julkisen avainparin hyödyntämiseen. Salausavaimet vastaavat toisiaan niin, että julkisella avaimella salattu viesti voidaan avata vain avainparin yksityisellä avaimella ja päinvastoin. Esimerkiksi salattava viesti salataan vastaanottajan julkisella avaimella ja puretaan vastaanottajan yksityisellä avaimella. Taulukko 1 kertoo, mitä avainta käytetään missäkin tilanteessa. PKI:n avulla voidaan selvittää, kenen hallussa julkista avainta vastaava yksityinen avain on. Näin voidaan valmistautua viestin lähettäjän henkilöllisyydestä.

Salausavain on pitkä numerosarja (Suomessa 1024-bittinen avain katsotaan turvalliseksi), joka on sekä salaus- että purkualgoritmin parametri. Salausalgoritmi puolestaan muuttaa viestin muotoon, jota vastaanottajan on mahdotonta lukea. Purkualgoritmi muuttaa viestin jälleen lukijalle ymmärrettävään muotoon. Purku- ja salausalgoritmin lisäksi viestin salaukseen ja purkuun käytetään avainta, joka on siis joko yksityinen tai julkinen. Vaikka avaimet ovat toisistaan matemaattisesti riippuvaisia, toisen avaimen avulla ei pysty päättelemään mitään toisesta avaimesta, eli vaikka saisi tietää käyttäjän julkisen avaimen, ei silti pysty päättelemään hänen salaista avaintaan [18]. Yksityinen avain on haltijansa henkilökohtainen avain, joka tulee pitää salaisena. Julkinen avain on puolestaan sovituilta tahoilta yleisesti saatavilla oleva avain, jota voidaan säilyttää esimerkiksi yleisessä hakemistossa, luotetulla kolmannella osapuolella tai avaimen haltija voi säilyttää sitä itse. Henkilön julkisen avaimen voi saada joko henkilöltä itseltään (esim. puhelimen tai sähköpostin välityksellä) tai luotetulta kolmannelta osapuolelta.

Kolmannen osapuolen tehtävä on lisäksi varmistaa julkisen avaimen aitous ja eheys. [17.]

*Taulukko 1: PKI:n salaisen ja julkisen avaimen käyttö [19].*

| Mitä halutaan tehdä | kenen avainta käytetään | Mitä avainta käytetään |
|---|---|---|
| Lähetetään salattu viesti | vastaanottajan | julkista |
| Lähetetään viesti, jossa salattu allekirjoitus | lähettäjän | yksityistä |
| Puretaan salattu viesti | vastaanottajan | yksityistä |
| Puretaan salattu allekirjoitus viestistä | lähettäjän | julkista |

Julkisen avaimen järjestelmän varmennusarkkitehtuurit voidaan jakaa kahteen ryhmään: puumaiseen arkkitehtuuriin, joista yleisin on X.500/X.509 ja verkkomaiseen arkkitehtuuriin eli PGP:hen (*Pretty Good Privacy*). X.500 on julkisen avaimen infrastruktuurissa käytetty hakemisto, joka pystyy taltioimaan tietoja ihmisistä ja olioista eripuolille verkkoa sijoitettuihin palvelimiin. X.509 puolestaan on osa X.500-standardia, ja X.509 määrittää tietorakenteen muodon, jossa varmennetiedot sijaitsevat [12, s. 43]. X.509 kutsutaan yleisesti julkisen avaimen varmenteeksi. X.500/X.509 on arkkitehtuuriltaan hierarkinen, ja sen lähtökohtana on juuri varmenneorganisaatio, josta luottamus luodaan ja josta se jaetaan hierarkiamaisesti kaikille verkon käyttäjille eri varmenneorganisaatioiden kautta. X.509-varmennetta käytetään julkisen avaimen liittämiseen tiettyyn yksilöön tai tahoon. PGP eli luottamusverkko on toinen julkisen avaimen arkkitehtuuri. PGP:ssä ei ole erillistä varmennusorganisaatiota, vaan käyttäjät luovat itse luottamusverkkonsa. [16.]

Digitaalista allekirjoitusta voidaan myös käyttää tunnistuksessa esim. verkkosivuille kirjautuessa. Yksinkertaisin tapa käyttää digitaalista varmennetta tunnistukseen on haaste-vastausjärjestelmä, jossa digitaalinen allekirjoitusalgoritmi korvaa salaisen algoritmin. [10, s. 49.]

## 4.8 SOAP

SOAP (Simple Object Access Protocol) on protokolla, jonka tehtävänä on lähettää XML (*eXtensible Markup Language*) -pohjaisia viestejä tietoliikenneverkoissa. SOAP käyttää viestien lähettämisessä yleensä HTTP (*Transfer Protocol over Secure Socket Layer*) tai HTTPS (*Hypertext Transfer Protocol over Secure Socket Layer*) -protokollia, mutta muitakin tiedonsiirtoprotokollia voidaan käyttää. SOAP muodostaa perustan verkkopalveluprotokollapinolle tarjoten perusviestintäkehyksen, jonka päälle voidaan rakentaa muita kehittyneempiä ja erikoistuneempia palveluja. Tiedonsiirtoprotokollana SOAP käyttää Internetin ohjelmakerroksen protokollaa. [20.]

SOAP-viesti koostuu kolmesta osasta: kuoresta, otsikosta ja sisällöstä. Kuori on nimensä mukaisesti SOAP-viestin uloin osa ja SOAP-viestejä kutsutaankin joskus myös kirjekuoriksi (envelope). Kirjekuori on aina XML-muotoinen dokumentti, ja sen on alettava envelope-elementillä. Kuori sisältää korkeintaan kaksi elementtiä: vapaaehtoisen otsikko-elementin (header) ja pakollisen sisältö-elementin (body). SOAP-otsikko on kokoelma yhdestä tai useammasta otsikkokentästä (header block), joiden avulla tietoa kuljetetaan SOAP-reitillä. SOAP-otsikkokenttä rajaa otsikon sisällä olevan tiedon loogisesti muodostettuun, yhteen laskennalliseen yksikköön, ja otsikkokenttä osoittaa aina yhteen SOAP-reitin solmuun. Kaikissa otsikkomerkinnöissä on käytettävä XML:n nimiavaruuksia nimiavaruuksien törmäämisten (namespace collision) välttämiseksi. SOAP-otsikko kenttä voi sisältää kolme erilaista SOAP-määritystä: actorin, mustUnderstantin tai encodingStylen. Actor kertoo, mille sovellukselle otsikko on tarkoitettu. MustUnderstant kertoo että viesti on pakko ymmärtää jokaisessa solmussa. EncodingStyle määrittää, kuinka tietoja tulisi välittää lankaa pitkin (serializion). Attribuutin arvo on yleensä yksi URI, *http://schemas. xmlsoap.org/soap/encoding/*, mutta sen arvona voi olla yksi tai useampi URI-osoite välilyönnillä erotettuna.

SOAP-viestin tärkein osa on runko-osa (body), jossa välitetään halutunlaista XML-pohjaista tietoa. SOAP-runkosisältö (*SOAP body*) sisältää kokoelman nollasta tai useammasta elementin tieto-osasta, jotka on tarkoitettu vastaanottajalle.

SOAP tarjoaa jaetun prosessointimallin, jossa oletetaan, että SOAP-viesti luodaan SOAP-lähettimessä ja lähetetään SOAP-vastaanottimeen nollan tai useamman SOAP-välittäjän kautta. SOAP-vastaanottimia, lähettäjiä ja välittäjiä kutsutaan yleisesti SOAP-solmuiksi. SOAP-lähetin lähettää SOAP-viestin, SOAP-vastaanotin ottaa vastaan SOAP-viestin ja SOAP-välittäjä toimii sekä lähettäjänä että vastaanottajana. Viestin luovaa SOAP-lähetintä kutsutaan alkuperäiseksi SOAP-lähettimeksi (Initial SOAP sender) ja viestin lopullista kohdetta lopulliseksi SOAP-vastaanottimeksi (ultimate SOAP receiver). SOAP-solmu, joka käsittelee SOAP-viestiä, toimii yhden tai useammin SOAP-roolin mukaan. Solmujen rooleista tunnistetaan kyseinen solmu ja sen tehtävä verkossa (onko solmu esim. lähettäjä tai vastaanottaja). Roolit tunnistetaan URI:sta (*Uniform Resoure Identifier*), jota kutsutaan SOAP-roolinimeksi. Solmujen roolinimiä on mm. next, none ja ultimateReceiver, joista ultimateReceiver on SOAP-viestin lopullisen kohdesolmun rooli, nextiä käyttävät niin välittäjäsolmut kuin lopullinen kohdesolmu, kun taas none tarkoittaa sitä, ettei solmu saa toimia silloin kuin se on tässä roolissa.

Kuten aikaisemmin mainittiin, SOAP-otsikkokenttä saattaa sisältää `role` *attribute information item:n* eli roolin ominaisuuden tieto-osion, jota käytetään kohdistamaan tiedonkulku tietyn SOAP-solmun kautta, silloin kun otsikkokentän SOAP-rooli on sama kuin kyseisen solmun SOAP-rooli. Otsikkokenttiä, joilla on none-rooli, ei koskaan muodollisesti käsitellä. Tällaiset SOAP-otsikkokentät voivat kuljettaa tietoa, jota tarvitaan muiden SOAP-otsikkokenttien käsittelyyn. Mikäli toistimet eivät poista kyseisiä otsikkokenttiä verkosta, ne välitetään viestissä lopulliselle vastaanottajalle. Otsikkokenttä, joka sisältää `mustUnderstand` attribute information item:n, on pakko käsitellä solmussa, mikäli kyseisen ominaisuuden tieto-osio on true.

Pakollisen SOAP-otsikkokentän oletetaan muuttavan muiden SOAP-otsikkokenttien tai SOAP-sisällön merkitystä SOAP-solmun täytyy joko käsitellä otsikkokenttä tai hylätä koko viesti. Jos viesti hylätään, niin solmu luo SOAP-virheen (SOAP fault, ominaisuuden tieto-osio, jossa on tiedot virheestä).

SOAP-viestin käsittelyyn kuuluu seuraavat vaiheet:

1. Solmun rooli/roolit määritetään.
2. Tunnistaa kaikki solmuun kohdistuvat pakolliset otsikkokentät.
3. Jos jotain pakollisista otsikkokentistä ei ymmärretä, generoidaan SOAP-virhe. Jos virhe luodaan niin silloin seuraavia vaiheita ei toteuteta.
4. Käsitellään kaikki solmuun kohdistuvat pakolliset otsikkokentät, ja lopullisen vastaanottajan tapauksessa myös SOAP-sisältö.
5. SOAP-välittäjän tapauksessa viesti lähetetään eteenpäin.

SOAP MEP (Message External Pattern) on malli, joka osoittaa mallin solmujen väliselle viestien vaihdolle. Eri MEP-malleja on mm. pyyntö/vastaus malli, yksisuuntainen malli ja peer-to-peer-keskustelumalli. SOAP-viestejä voidaan vaihtaa useiden eri alusta protokollien (esim. muiden sovelluskerroksen protokollien) avulla. Määrityksiä sille, kuinka SOAP-viestit kulkevat solmulta toiselle aliprotokollan avulla kutsutaan SOAP-sitomiseksi (SOAP binding). SOAP-sitominen tarjoaa keskitetyn toteutuksen SOAP-solmujen vaihtamille SOAP-tietopaketeille sekä tarjoaa mekanismin, joka tukee ominaisuuksia, joita SOAP-sovellukset tarvitsevat. Yleisimmin SOAP sidotaan HTTP-protokollaan. [21.]

## 4.9 SSL ja TLS

SSL (*Secure Socket Layer*) on kuljetuskerroksen suojausprotokolla, joka toimii TCP (*Transmission Control Protocol*) -kerroksen yläpuolella. SSL tarjoaa turvallisen tiedonsiirtopalvelun ohjelmistokerroksen protokollille.

SSL on käytössä monissa verkkopalveluissa esim. pankkisovelluksissa, ja se tukee yhdessä HTTP:n (*Hypertext Transfer Protocol*) kanssa turvallista tiedonsiirtoa käyttäjän ja verkkopalvelimen välillä, jotta käyttäjät voisivat lähettää tärkeitä tietojaan kuten luottokortin numeron palvelimelle. Internet-osoitteissa SSL:n käytön näkee siitä, että Internet-osoitteessa http:n tilalla on https. SSL:llä on neljä aliprotokollaa, jotka ovat:

- SSL-kättelyprotokolla, jonka avulla verkkopalvelin ja käyttäjä aloittavat istunnon. Kättelyn aikana palvelin ja käyttäjä tunnistavat toisensa, sekä sopivat tiedonsiirtoon liittyvistä suojauksista.
- SSL-hälytysprotokolla, joka lähettää epänormaalin tapauksen yhteydessä hälytysviestejä palvelimen ja käyttäjän välillä.
- SSL-koodin vaihtoprotokolla (SSL change cipher specprotocol), joka vaihtaa koodia sen hetkiselle yhteydelle. Tätä käytetään kättelyvaiheen lopussa.
- SSL-tallennusprotokolla, joka tarjoaa salauksen sekä tiedoneheys palvelut ohjelma kerroksen tiedonsiirrolle. [22, s. 242 – 244.]

SSL oli alun perin Netscape-selaimeen kehitetty suojausprotokolla. Kun Netscape luovutti kehityksen IETF (*Internet Engineering Task Force*) -standardointiorganisaatiolle, joka kehitti siitä oman versionsa, jota kutsutaan TLS eli Transport Layer Securityksi. Käytännössä TLS ja SSL ovat pieniä eroja lukuun ottamatta samanlaisia protokollia. [23.]

## 4.10 XML

XML-kieli on merkintäkieli, jolla tiedon merkitys on kuvattavissa tiedon sekaan. XML-kieltä käytetään sekä formaattina tiedonvälitysjärjestelmien välillä että dokumenttien tallennukseen. XML-kieli on rakenteellinen kuvauskieli, joka auttaa jäsentämään laajoja kuvausmassoja paremmin. Se muistuttaa melko paljon HTML (*Hypertext Markup Language*) -kieltä, jolla verkkosivuja tehdään, mutta sitä ei ole tarkoitettu sivunkuvauskieleksi. Sen sijaan XML-kielellä kuvataan tiedon rakennetta ilman ennalta määrättyjä koodeja.

XML:n tuomia etuja on mm. sisältövirheiden välttäminen, sisällön yhdenmukaisempi tallennusmuoto, tiedon haun helpottaminen, tiedon pitkäaikaissäilyvyyden parantaminen sekä käsittelyvaiheiden automatisointi XML-kielellä voi muodostaa uusia koodeja, joiden avulla voidaan luoda dokumentteja hyvinkin erilaisiin ja erityisiin tarkoituksiin. XML-dokumentti koostuu elementeistä, jotka koostuvat alku- ja loppumerkeistä. Alkumerkkiä merkitään <elementti> ja loppumerkkiä </elementti>. Elementtejä voi olla sisäkkäin rajaton määrä. [24.]

Seuraavaksi on yksinkertainen esimerkki XML dokumentista.

<? XML version = ”1.0”?>

< ! DOCTYPE VBS SYSTEM ”book.dtd”>

<kirja>

<otsikko> XML aloittelijoille </otsikko>

<kirjailija> Elizabeth Chang </kirjailija>

<tiivistelmä > XML:n perusteet </tiivistelmä>

<aihe>

<avainsana = ”internet ohjelmointi”>

<avainsana = internet tietokanta>

</aihe>

<hinta> 25 euroa </hinta>

</kirja>

Ensimmäinen rivi kertoo käytetyn XML:n versionumeron. Toisella rivillä on DTD (Document Type Definition) -tiedostonimi, jossa dokumentin elementeille ja attribuuteille sallitut ilmenemismuodot määritellään. Jokaisella dokumentilla on juuri osa ja tässä esimerkissä se on <kirja>. Lopuilla riveillä määritellään juurielementin lapsielementit (otsikko, kirjailija, tiivistelmä, aihe ja hinta). Viimeisellä rivillä määritellään juurielementin loppu </kirja>. XML-dokumentin sanotaan olevan hyvin muodostettu (Well Formet), jos se täyttää XML 1.0 suosituksissa annetut ohjeet. Näiden ohjeiden mukaan XML-dokumenttiin kuuluu mm. juurielementti, käytetyn XML:n versionumero sekä alku- ja loppumerkki jokaiselle elementille.

Samassa suosituksessa annetaan myös ohjeet siitä millainen on pätevä ”valid” XML-dokumentti. Pätevä dokumentin ehdot täyttyvät kun <! DOCTYPE> (esimerkin toinen rivi) on määritelty. [22, s. 269 – 270.]

### *4.10.1 XML Signature*

XML-allekirjoitusta käytetään koko XML-viestin tai jonkin viestin osan koskemattomuuden varmistamiseen. XML-allekirjoitus standartissa ei määritellä uusia allekirjoituskeinoja vaan sen sijaan siinä määritellään kuinka digitaalinen allekirjoitus toimii XML-dokumenteissa. XML-allekirjoitus standardi toimii pohjana muille standardeille kuten SAML:lle (*Security Assertion Markup Language*). XML-allekirjotusstandardissa on useita satunnaisuuksia (contingencies) sen käyttöön liittyen, ja ne saavat standardin näyttämään monimutkaiselta, mutta sen tärkeimmät osat ovat melko helposti ymmärrettäviä. XML-allekirjoitus sijaitsee <Signature/>-elementissä, joka koostuu kolmesta osasta:

- <SignetInfo/>-elementti sisältää viittauksen allekirjoitettuun tietoon sekä tiedot mm. siitä mitä allekirjoitustapaa on käytetty ja miten viestiä on muokattu.
- <SignatureValue/>-elementti sisältää varsinaisen allekirjoituksen.
- <KeyInfo/>-elementti sisältää avaintiedot, joita tarvitaan allekirjoituksen vahvistamiseen. [10, s. 99.]

### *4.10.2 XML-salaus*

Kuten kaikki kommunikointitavat Internetissä, myös XML-dokumentit voidaan salata kokonaan ennen kuin ne lähetetään verkon yli. Ongelmana kuitenkin on se, että osa XML-dokumentista pitää olla selväkielisessä (salaamattomassa) muodossa. Esimerkiksi XML-pohjaisen SOAP-viestin otsikko-osan täytyy olla selväkielisessä muodossa, koska välittäjien pitää nähdä reititystiedot sekä muu tärkeä informaatio, joka sijaitsee SOAP-otsikossa. Vaihtoehtona koko viestin salaamiselle olisi lähettää vaikkapa SSL-suojatun yhteyden kautta viestilaitteelta toiselle. Ongelmana tässä tapauksessa on se, että viesti näkyisi kokonaisuudessaan selväkielisenä kaikille välittäjälaitteille.

XML-salauksen tarkoituksena on korjata nämä ongelmat tarjoamalla dokumentin osittaisen salauksen. Kuten XML-allekirjoituksessa myös XML-salauksella on useita eri satunnaisuuksia, mutta sen perusidea on helposti ymmärrettävissä. Salatut tiedot XML-dokumentissa tunnistetaan <EncryptedData/>-elementistä, joka koostuu kahdesta osasta.

- <EncryptionMethod>-elementti antaa tiedot <KeyInfo/>-elementistä, joka on sama elementti kuin XML-allekirjoituksessa. <EncryptionMethod>-elementti ei ole pakollinen <EncryptedData/>-elementissä.
- <CipherData>-elementti sisältää, joko salatut tiedot <CipherValue/>-elementin sisällä tai viittauksen salattuun tietoon, jolloin viittaus on <CipherReference>-elementin sisällä. [10, s. 101.]

### *4.10.3 SAML*

SAML (*Security Assertion Markup Language*) on suunniteltu XML-standardin tunnistus- ja valtuutustietojen vaihtamiseen eri turvallisuus-domainien kesken (palveluntarjoajan ja henkilöllisyyden tarjoajan välillä). SAML:n pääasiallinen tarkoitus on ratkaista kertakirjautumisen (SSO, *Single Sign-On*, joka käydään läpi kappaleessa 6) ongelmat. Kertakirjautumisratkaisuja on useita erilaisia ratkaisuja sisäverkoissa esim. evästeiden (cookies) käyttö, mutta niiden laajentaminen sisäverkosta ulkoverkkoon on ollut ongelmallista. SAML:sta on tullut määrittävästandardi, joka luo pohjan monille verkkopohjaisille SSO-ratkaisuille. SAML olettaa, että käyttäjä on kirjautunut vähintään yhden henkilöllisyyden antajan (identity provider) palveluun. Henkilöllisyyden antajan odotetaan tarjoavan paikallisia tunnistuspalveluja käyttäjälle ja palveluntarjoaja luottaa henkilöllisyyden tarjoajaan käyttäjän tunnistuksessa. Käytännössä palveluntarjoajan käyttäjäntunnistus tapahtuu niin, että henkilöllisyyden tarjoaja lähettää käyttäjän pyynnöstä SAML-vakuutuksen (SAML-assertion) palveluntarjoajalla, ja tämän vakuutuksen pohjalta palveluntarjoaja tekee päätöksen siitä, hyväksytäänkö vai hylätäänkö käyttäjän tunnistus. SAML on rakennettu seuraavien standardien varaan:

- XML Schema
- XML Signature

- XML encryption
- Hypertext Transfer Protocol (HTTP)
- SOAP.

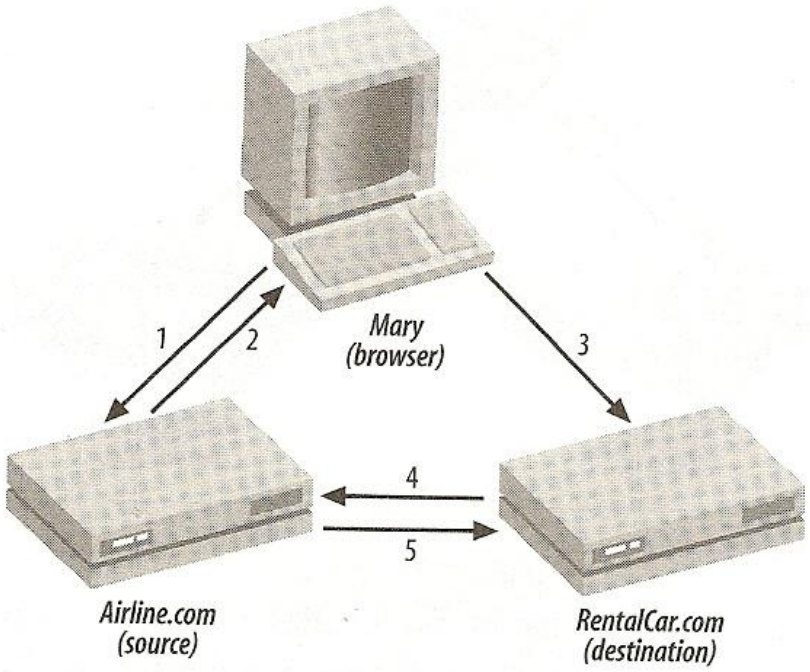
SAML on siis XML-pohjainen ja käyttää XML-pohjaista vakuutusta, protokollaa, sitomista ja profiilia. SAML Core ilmoittaa SAML-vakuutuksen yleisen syntaxin ja merkityksen (semantic) sekä vakuutuksen pyynnössä ja siirrossa käytettävän protokollan. SAML protocol ilmoittaa mitä siirretään ja SAML-binding ilmoittaa kuinka SAML-pyyntö ja SAML–vastaus kuljetetaan normaaleille viestintä- ja kommunikointiprotokollilla. Tärkeä SAML-binding eli SAML-sitominen on sitominen SOAP:iin. SAML-profiili keskittyy ilmaisemaan määritellyt käyttötapaukset silloin kuin käytetään tiettyjä kombinaatioita vakuutuksista, protokollista ja sidoksista. [25.]

SAML-viranomaiseksi kutsutaan online-palvelua, joka vastaa käyttäjän SAML-pyyntöihin. SAML-viranomaisia on kolmea eri tyyppiä: tunnistus viranomainen, ominaisuus viranomainen (*attribute authority*) ja politiikan päättämispisteet (*policy decision point, PDP*). Käytännössä yksi ja sama taho voi tuottaa kaikki kolme erilaista vakuutusta. SAML-viranomaistyyppien antavat vakuutukset ovat:

- SAML authentication assertion, eli SAML-tunnistusvakuutus, jolla vastataan silloin, kun pyydetään jonkin henkilön valtuutustietoja. Tunnistusvaltuutus kertoo, että henkilö A on tunnistettu keinolla X kello xx.xx. Esim. Henkilö Pekka yhtiöstä XCOM on tunnistettu salasanan avulla 6.2.2009 klo 13.45.
- SAML attribute assertion, eli SAML-ominaisuusvakuutusta voidaan pyytää sen jälkeen kuin SAML-tunnistusvakuutus on lähetetty. Ominaisuusvakuutuksella vastataan silloin, kun kysytään tiettyyn käyttäjään liittyviä ominaisuuksia. Ominaisuusvaltuutus kertoo että henkilö A:n liitetään palvelut X ja Y, joilla on arvot B ja C. Esim. Henkilöllä Pekka on kyky Email arvolla pekka@xcom.fi ja kyky osasto arvolla tietokonetekniikka.

- SAML authorization assertion eli SAML-valtuutusvakuutus on PDP:n palauttama vastaus pyyntöön, jossa tiedustellaan käyttäjän lupaa päästä käsiksi tiettyyn resurssiin. Valtuutusvakuutus ilmoittaa että henkilö A:lla on (tai ei ole) myönnetty oikeudetta Resurssin X toimintoon Y, kuten voidaan todeta tiedostosta B. Esim. Käyttäjällä http://a.com/palvelut on myönnetty oikeus lukea tiedostoa http://b.com/tiedot, kuten on todistettu vakuutuksissa A1, A2 ja A7.

SAML:lla on neljä pääasiallista käyttötapaa, kaksi verkkoselain profiilia ja kaksi SOAP profiilia. Verkkoselaimen profiilit ovat pull- ja push-profile. Pull eli vetomallissa SAML:a käytetään muodostamaan Single Sign-on kahden verkkosivun välille. Veto-malli käyttää SAML-artefaktia, jotka käytännössä ovat sama asia kuin tunniste (token). Artefaktit lähetetään sivulta toiselle URL tiedustelumerkkijonoa (query string) käyttäen. Vakuutuksen (assertion) tekevä sivu (lähde) muodostaa linkin kohdesivuun, joka sisältää kyseisen URL:n artefaktin. Käyttäjän klikatessa hiirellä linkkiä kohdeosoite saa artefaktin osana http get-pyyntöä. Kuvassa 6 näkyy esimerkki veto-profiilista. Esimerkissä oletetaan että sivustot ovat sopineet yhteistyöstä ja että Mary on antanut sivustoille luvan tarkistaa hänen henkilöllisyytensä.




*Kuva 6: Selaimen veto-profiili [10, s. 105.]*

1. Käyttäjä Mary on kirjautunut ja tunnistettu Airline.com sivulla (lähdesivu) ja hän ostaa sivulta lentolipun.

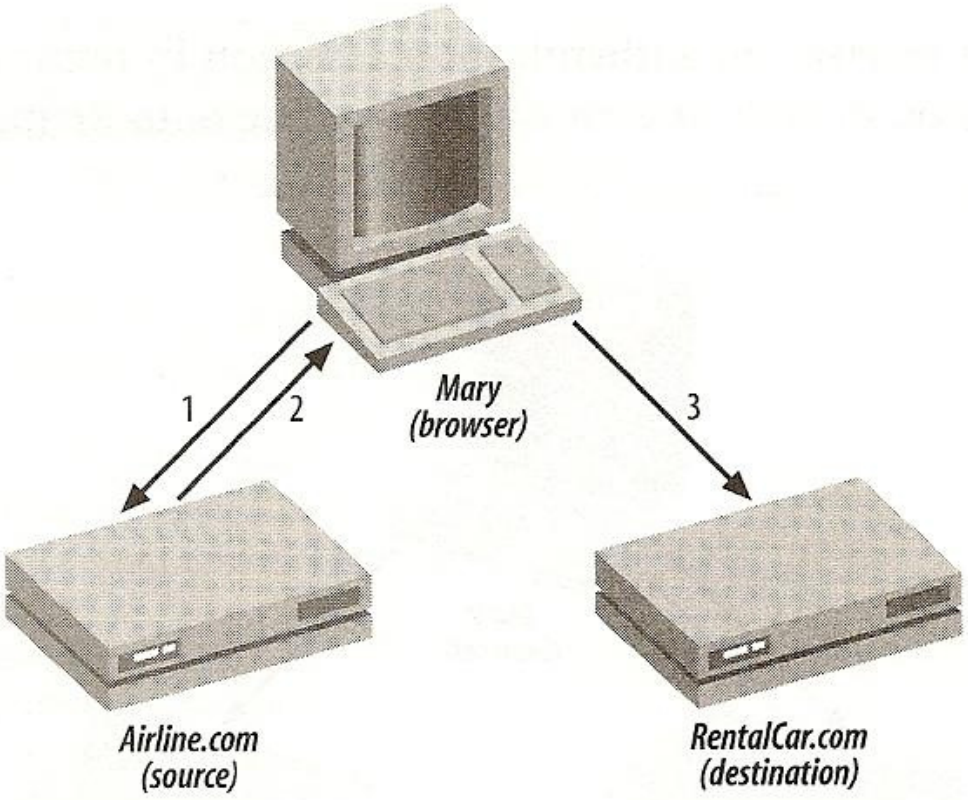
2. Oston aikana Airline.com suosittelee Marylle auton vuokraamista RentalCar.com sivulta (kohdesivu) ja tarjoaa SAML-artefaktin sisältävän linkin kyseiselle sivulle.

3. Maryn klikatessa linkkiä artefakti siirtää hänet RentalCar.com sivulle.

4. RentalCar.com tekee SAML-pyynnön Maryn tunnistamisesta käyttäen SAML-artifaktia.

5. RentalCar.com vastaanottaa tunnistusvakuutuksen Airline.comilta. Tunnistusvakuutuksen avulla Mary voi asioida RentalCar.comissa, ilman että hänen täytyy kirjautua sinne erikseen.

Toinen selainpohjainen profiili on työntö-profiili, jossa lähdesivu luo verkkokaavakkeen, joka sisältää vakuutuksen käyttäjän henkilöllisyydestä. Käyttäjän lähettäessä kaavakkeen se lähetetään kohdesivulle HTTP POST:n avulla, jolloin vakuutus työnnetään lähdesivulta kohdesivulle HTML kaavakemekanismia käyttäen. Työntö-profiilissa lähdesivu allekirjoittaa vakuutuksen digitaalisella allekirjoituksella käyttäen hyväksi XML-allekirjoitusta. Allekirjoituksen avulla kohdesivu pystyy varmistamaan vakuutuksen antajan henkilöllisyyden ja näin ollen tarkistamaan, että vakuutuksen antaja on luotettu lähde. Kuvassa 7 näkyy esimerkki työntömallin toiminnasta käytännössä.



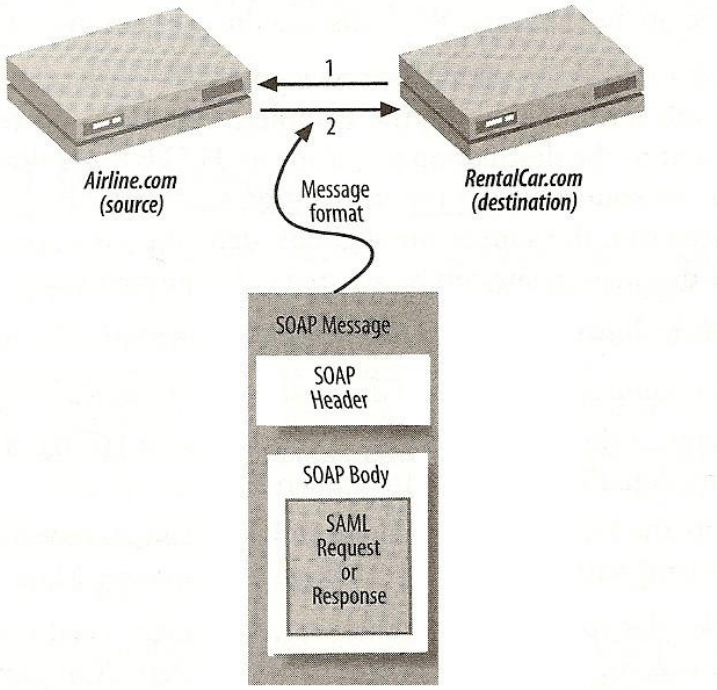

*Kuva 7: Selaimen työntö-profiili. [10, s. 106.]*

1. Mary käy Airline.com sivulla, tunnistautuu ja ostaa lentolipun.

2. Jossain välissä asiointia Airline.com palauttaa HTML-kaavakkeen, joka sisältää digitaalisesti allekirjoitetun vakuutuksen.

3. Mary lähettää saamansa kaavakkeen RentalCar.comiin, joka tarkistaa digitaalisen allekirjoituksen. Tarkistettuaan allekirjoituksen aitouden RentalCar.com käsittelee Maryn tekemän pyynnön.

Mikäli Mary haluaa vuokrata auton RentalCar.comista, RentalCar.com saattaa tarvita lisätietoja Airline.comilta. Saadakseen tarvitsemansa tiedot Airline.com voi tehdä ominaisuuspyynnön (Attribude request) kysyäkseen tarvitsemiaan tietoja esim. Maryn pankkikortin numeroa tai hänen osoitettaan. Airline.com vastaus ominaisuuspyyntöön lähetettäisiin SOAP-viestin sisältöosassa. Tällaista tapaa kutsutaan suoraksi pyynnöksi ja vastaukseksi ja sen toimintaperiaate näkyy kuvassa 8.

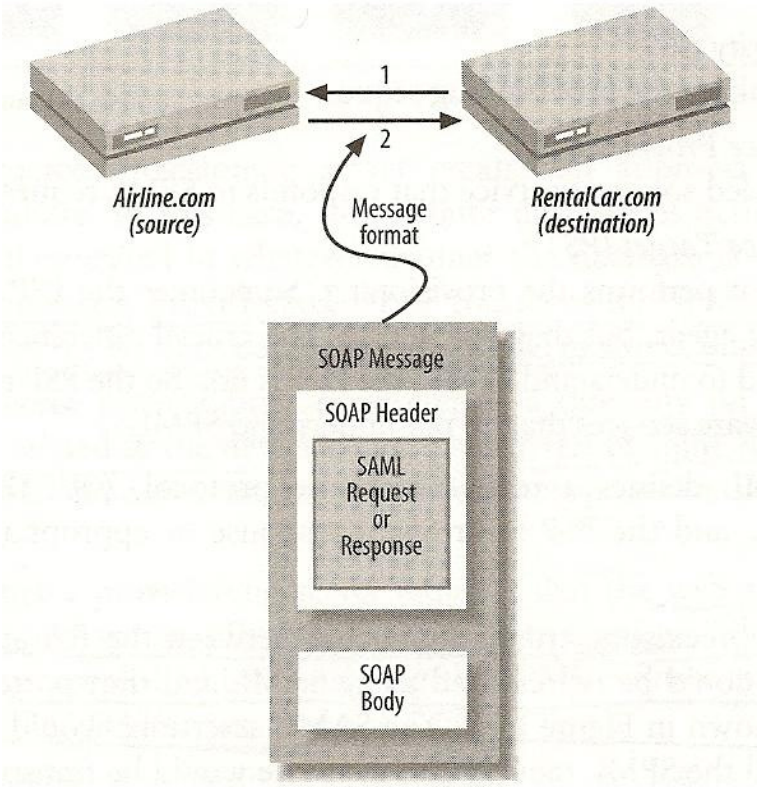



*Kuva 8: Suora vastaus ja pyyntö. [10, s. 106.]*

1.RentalCar.com tekee pyynnön Marya koskevista tiedoista

2. Airline.com vastaa RentalCar.comin tekemään pyyntöön

Edellisen esimerkissä SAML-pyyntö ja –vastaus ovat SOAP-viestin sisältöosassa, koska niissä on jokin kysytty tieto. Joskus SAML:a käytetään tunnistamaan itse SOAP-viesti, jolloin SAML-vakuutus SOAP-sisällön luotettavuudesta on SOAP-otsikossa. Tällaista mallia kutsutaan verkkopalvelu-profiiliksi ja esimerkki sen toiminnasta näkyy kuvasta 9.




*Kuva 9: Verkkopalvelupyyntö. [10, s. 107.]*

Esimerkki toimii samalla tavoin kuin kuvan 8 esimerkki. Ainoana erona on se, että SAML-vakuutus on SOAP-viestin otsikko-osassa eikä sisällössä niin kuin aikaisemmassa esimerkissä. SAML:n pääasiallinen hyöty on se, että käyttäjän turvallisuusviite (security context) kulkee käyttäjän mukana, jolloin uutta turvallisuusviitettä ei tarvitse tehdä aina uudelle sivulle mentäessä. Tämä vähentää tarvetta varastoida ja synkronisoida tunnistus-, todennus- ja valtuutustietoja jokaisella sivulla. Tiedot ovat näin paremmassa turvassa ja kertakirjautumisen ja ominaisuuksiin perustuvan tunnistuksen käyttö mahdollistetaan. Lisäksi useita palveluntarjoajia voidaan yhdistää (federated) käyttämällä samaa kieltä ja protokollaa. [10, s. 102 – 107.]

### *4.10.4 SPML*

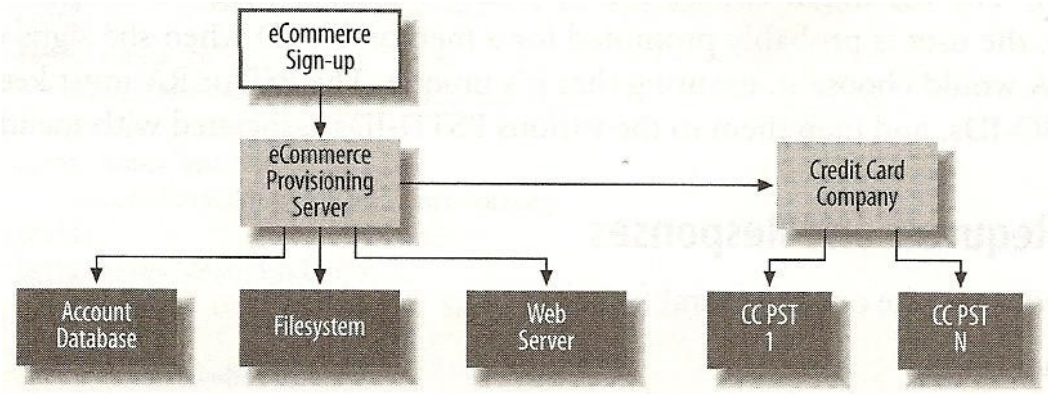
SPLM (*Service Provisioning Markup Language*) on XML-pohjainen kieli varaus (Provisioning) pyyntöjen ja vastausten vaihtoon. SPML:n avulla luodaan tunnukset ja palvelut, joita tunnistuspalvelut käyttävät. SPLM on vielä melko uusi, joten sitä ei tällä hetkellä tue kuin muutama tavarantoimittaja, mutta sitä tai jotain sen tapaista kieltä tarvitaan luomaan automaattinen tunnistusjärjestelmä, jota tarvitaan yksittäisissä tunnisteissa (identity).

SPML:n tarkoituksena on tukea kaikkia identiteetin valvomisessa tarvittavia toiminteita sen koko elinkaaren ajan. SPML:ssä määritellään kolme pääasiallista roolia:

- Requesting Authority (RA) on varauspyynnön tekevä taho.
- Provisioning Service Provider (PSP) on SPML-pohjainen ohjelmisto, joka vastaa RA:n tekemään SPML-pyyntöön.
- Provisioning Service Target (PST) on taho, joka suorittaa varauksen.

Joskus PSP ja PST ovat sama ohjelmistoagentti, mutta se ei ole tarpeellista. Tärkein ero PST:n ja PSP:n välillä on se, että PSP:n täytyy ymmärtää SPML:ä, jotta se voi olla ennen ohjelmistopalvelua, joka ei ymmärrä SPML:ä. PST:n ei puolestaan tarvitse ymmärtää SPML:ä. SPML määrittää SAML:n tapaan pyyntö-vastausprotokollan, jossa RA tekee pyynnön SPML-muodossa ja PSP palauttaa vastauksen tai virheilmoituksen niin ikään SPML-muotoisena. Jotta SPML toimisi, RA:n ja PSP:n välillä on oltava etukäteen luotu luottamussuhde, joka voidaan luoda vaikkapa käyttämällä SAML:a kuvan 9 mukaisesti (eli SAML-vakuutus SOAP-otsikossa ja SPML-pyyntö ja vastaus kuljetettaisiin SOAP-viestin sisältö osassa). PSP voi myös tehdä pyyntöjä suoraan toiselle PSP:lle. Käytetään jälleen esimerkkiä selventämään SPML:n eri osien toimintaa ja suhteita. Kuvassa 10 nähdään esimerkki varaustoiminnoista, silloin kuin luodaan uusi käyttäjätunnus sähköiseenkauppajärjestelmään. Sähköisen kaupan sisäänkirjautuminen (eCommerce Sign-up), joka toimii tässä tapauksessa RA:na, lähettää SPML-pyynnön uuden käyttäjätilin luomisesta sähköisenkaupan varauspalvelimelle (eCommence Provisioning Server), joka toimii tässä tapauksessa PSP:nä. Kuten aikaisemmin mainittiin, SPML-pyyntö voidaan siirtää SOAP-viestissä, jonka otsikko-osassa on SAML-vakuutus.

Varauspalvelin (PSP) käyttää SAML-vakuutusta päättääkseen, onko RA:lla valtuudet luoda uusi käyttäjätili. Mikäli RA:lla on valtuudet tilin luomiseen, palvelin luo pyydetyn tilin.




*Kuva 10: SPML:n toiminta sähköisenkaupankäynnin yhteydessä [10, s. 109]*

Sähköisenkaupan varauspalvelin suorittaa neljä eri toimintoa, neljän eri PST:n kanssa, jotta uudentilin luominen saataisiin valmiiksi:

1. Varauspalvelin luo rekisteritiedoston kirjanpitotietokantaan (accounting database), joka toimii tässä tapauksessa PST:nä. Tämä toiminto autentikoitaan tietokantaan ohjelmoidulla tavalla, ja varauspalvelin kääntää SPML-pyynnön SQL (*Structured Query Language*) muotoon ja lähettää sen kirjanpitotietokantaan JDEC (*Java Databese Connectivity*) tai jotain muuta vastaavaa rajapintaa käyttäen. Muutos on pakko tehdä, koska tietokanta ei ymmärrä SPML-kieltä.

2. Varauspalvelin luo tietojärjestelmään (Filesystem) hakemiston käyttäjällä JNDI:tä (*Java Naming and Directory Language*) tai jotain vastaavaa rajapintaa. Luotuun hakemistoon voidaan tallentaa uuteen käyttäjätiliin liittyviä tietoja. Tunnistus ja tarvittavat komennot suoritetaan jälleen PST:n (tietojärjestelmän) asetusten mukaisesti.

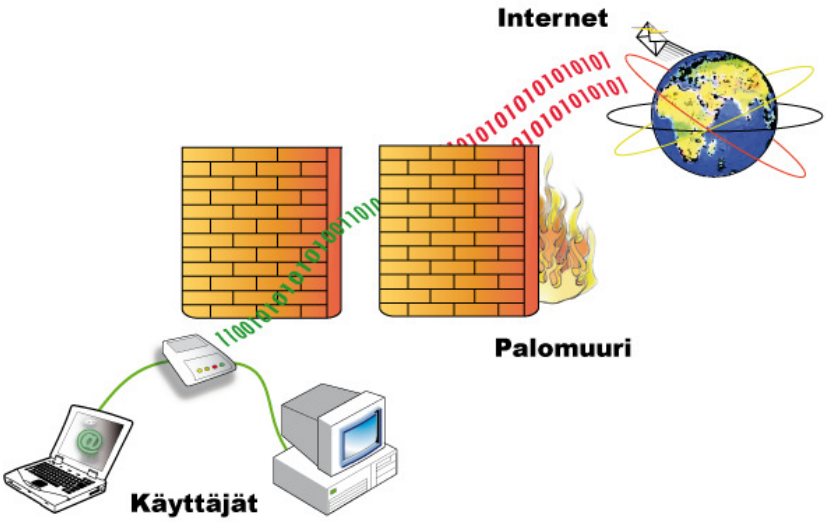
3. Varauspalvelin pyytää jonkin verkkopalvelimen autentikointi järjestelmää esim. LDAP (*Lightweight Directory Access Protocol*) luomaan käyttäjätilin. Tämä toteutetaan samalla tavalla kuin kohdissa 1 ja 2, eli SPML-pyyntö käännetään LDAP:n ymmärtämään käskymuotoon vaikkapa JNDI:n.

4. Varauspalvelin lähettää uuden SPML-pyynnön luottokorttiyhtiön SPS:lle, jossa pyydetään kauppa käyttäjätilin (Merchant account) luomista (provisioning). Luottokorttiyhtiön PSP puolestaan lähettää omia SPML-pyyntöjään toteuttaakseen halutun palvelun.

SPML siis luo ja valvoo tunnisteita. Luodun tunnisteen täytyy olla ainutkertainen (unique) ainoastaan kyseisen PSP:n nimiavaruudessa (namespace). Tunnisteita on kahta tyyppiä. Toisia tunnisteita käytetään PSP:hen tai kohdejärjestelmän alueella ja toisia globaalissa ympäristössä. Yksittäiseen PSP: yhdistettyjä tunnisteita kutsutaan PSTD-ID:ksi (Provisioning Target Data Identifiers). Jotta PSTD-ID:t olisivat hyödyllisiä, täytyy niiden olla yhteydessä johonkin globaaliin tunnisteeseen. SPML-standardissa tätä globaalia tunnistetta kutsutaan PSO-ID:ksi (Provision Service Objekt Identifier). RA voi valita PSO-ID:n tai PSP voi huolehtia siitä. Joko PSP:n tai RA:n pitää silmällä PSO-ID:itä ja kartoittaa (map) ne useisiin PSTD:hin, jotka ovat yhteydessä henkilötunnisteeseen. [10, 108 – 110.]

### 4.11 Palomuuri

Palomuurilla tarkoitetaan tietotekniikassa eristävää moniosaista järjestelmää, jonka tehtävänä on suodattaa suojattavan verkon (sisäverkko, intranet) ja ulkopuolisen verkon (maailmanlaajuinen verkko, internet) välisiä yhteyksiä. Kaikki sisäverkolta tuleva ja lähtevä tieto kulkee palomuurin läpi. Palomuureja tarvitaan useimmiten internetyhteyksiltä tulevien hyökkäysten torjumiseksi. Palomuuri suoriutuu tästä sääntöjen avulla, joiden avulla suodataan pois tarpeettomat tiedot. Palomuurijärjestelmä koostuu usein kahdesta komponentista, pakettisuodattamista ja yhdyskäytävästä.




*Kuva 11: Palomuurin toiminta [26].*

Palomuuritekniikoita on kolme eri tyyppiä: pakettisuodatin, sovelluspalomuuri sekä piiritason palomuuri. Näistä pakettisuodatin on kaikkein yksinkertaisin. Se toimii verkkokerroksessa ja seuloo pakettivirrasta paketteja lähde- ja kohdeosoitteen sekö porttien perusteella [22, s. 237]. Pakettisuodatinpalomuureja on kahden tyyppisiä: tilallisia (stateful) ja tilattomia (stateless). Tilattomassa palomuurissa paketti verrataan säännöstöön ja vain säännöstössä sallitut paketit pääsevät läpi ja muut paketit hylätään. Tilallinen palomuuri puolestaan pitää kirjaa TCP ja UDP (*User Datagram Program*) -yhteyksistä ja sallii vain yhteyksiin kuuluvat paketit. Näistä kahdesta mallista tilallinen mahdollistaa tarkemman valvonnan. Pakettisuodatinpalomuurien ongelmana on se, että paluu pakettien portteja ei voida kaikissa tilanteissa tietää tarkasti, jolloin kaikki portit yli portti numeron 1024 on avattava paluuyhteydelle, jotta tärkeät tiedot pääsisivät läpi palomuurista. Porttien avaaminen tarkoittaa myös sitä, että näistä porteista saadaan silloin yhteys sisäverkkoon ilman että palomuuri tulee väliin. Tilallisissa palomuureissa voidaan tarvittaessa lisätä sääntöjä tunnettuja protokollia varten, jolloin yli portin 1024 olevat portit voidaan estää, jos kaikki tarpeelliset portti numerot (yli 1024 menevät) määritellään palomuurille erikseen.  Sovelluspalomuuri tarkkailee paketin sisältämää dataa tarkkaillen, sisältävätkö ne laittomia komentoja (tarkkailu tehdään porttinumeron perusteella). [26.]

Sovelluspalomuuri toimii sovelluskerroksessa, jossa se toimii samalla tavalla kuin proxy-palvelin. Sisäänpääsyn kontrollointi tapahtuu tällöin sovelluskerroksessa eikä verkko kerroksessa. Jotta jotain tiettyä palvelua voidaan käyttää, täytyy palvelua vastaava proxy-palvelu asentaa sovellussuodattimeen. Käyttäjä ottaa ensin yhteyttä sovelluspalomuuriin, joka ottaa sen jälkeen yhteyttä käyttäjän haluamaan palveluun. Tällöin tunnistus jakaantuu kahteen eri vaiheeseen käyttäjän ja palomuurin väliseen tunnistukseen ja palomuurin ja sovelluksen väliseen tunnistukseen. Sovelluspalomuurit ovat turvallisempia kuin pakettisuodattimet, koska ne pystyvät sekä valvomaan ohjelma tason käskyjä, että piilottaa sisäisen käyttäjän (host) ulkopuolisilta verkoilta. Tämä johtuu siitä, että kun pakettia lähetetään Internetiin, vain sovelluspalomuurin lähde IP-osoite sisältyy IP-pakettiin (sovelluksen/käyttäjän omaa IP-osoitetta ei tässä tapauksessa löydy IP-paketista). Sovelluspalomuurin haittapuolena on viiveen kasvu, joka saattaa aiheuttaa ”pullonkauloja” sisäverkkoon.

Kolmas palomuurityyppi on piiritason yhdyskäytäväpalomuuri, joka toimii sovelluspalomuurin tapaan lähettäjä ja vastaanottajan välisenä ”agenttina” (eli käyttäjä ottaa ensin yhteyttä palomuuriin, joka ottaa yhteyttä vastaanottajaan/sovellukseen). Piiritason palomuuri toimii samalla tavalla kuin piirikytkentäinen puhelinverkko, eli käyttäjän täytyy tehdä kytkentä kohteeseen ennen kuin tiedonsiirto voi alkaa. Kytkennän tekeminen saattaa vaatia oman ohjelmistonsa. Yhteyden muodostamisen jälkeen tietopaketit kuljetetaan sisä- ja ulkoverkon välille muodostetun yhteyden yli. [22, s. 238 – 239.]

### 4.12 IPSec

IPSec eli IP-turvallisuusprotokolla on verkkokerroksen turvallisuusprotokolla, joka tukee tunnistus- ja salauspalveluja. Tämä toteutetaan lisäämällä IP-kehykseen IPSec-otsikko, IP-otsikon ja hyötykuorman väliin. Tämä IPSec-otsikko lisätään IP-kehykseen, joka lähettäjän tietokoneella tai IPSec:n sallivalla yhdyskäytävällä (IPSec-enabled gateway). IPSec-otsikoita on kahta tyyppiä, tunnistusotsikko (AH, *Authentication header*) sekä hyötykuorman turvallisuuskapselointi (ESP, *Encapsuling Security Payload*). Tunnistusotsikko vahvistaa IP-paketin tunnistuksen ja varmistaa paketin sisällön eheyden.

Eli toisin sanoen tunnistusotsikon avulla vastaanottaja tarkistaa, ettei IP-paketin sisältöä ole muutettu lähettämisen jälkeen. Mikäli paketin sisällön havaitaan muuttuneen, lähetetään lähettäjälle uudelleenlähetyspyyntö, jolloin lähettäjä lähettää saman paketin uudestaan. Tunnistusotsikon käyttö ei kuitenkaan riitä paketin tietojen pitämiseen luottamuksellisena, koska pakettia ei ole salattu. ESP eli paketin turvallisuuskapselointi tarjoaa salauspalvelun sekä valinnaisen tunnistepalvelun. Niin ESP:llä kuin AH:kin on kaksi toimintatilaa (mode), kuljetustila ja tunnelitila. Kuljetustilassa ylemmän kerroksen tieto on suojattu, kun taas tunnelitilassa kaikki tieto on suojattua. Kuljetustilassa vastaanottajan laitteiston pitää tukea IPSec:ä, kun taas tunnelitilassa se ei ole välttämätöntä. Jotta IPSec:ä voidaan käyttää kahden koneen välillä, täytyy ensin muodostaa SA (*Security Association*) eli turvallisuusyhteistyö, jossa määritellään mm. vaadittava suojaus (AH vai ESP), salaus ja tunnistustavat sekä salaus funktioiden suorittamiseen tarvittavat avaimet. Mikäli halutaan tehdä kaksipuolinen yhteys (eli koneelta A koneelle B sekä koneelta B koneelle A), niin silloin täytyy tehdä kumpaankin suuntaan oma SA. Tällöin on mahdollista, että kahden koneen väliset SA:t voivat olla erilaiset eri suuntiin mentäessä.

SA voidaan tehdä joko manuaalisesti tai dynaamisesti. Manuaalisessa tapauksessa järjestelmänvalvoja voi asentaa SA:n järjestelmän alkuasetusten aikana. Dynaamisessa tapauksessa kaksi tietokonetta neuvottelee SA:n käyttäen internetavaimenvaihto (IKE, *Internet Key Exchange*) -protokollaa. SA:n luomisen jälkeen siihen liitetään SPI (*Security Parameter Index*) eli turvallisuusparametrihakemisto, joka määritellään kaikkien SA:n kuuluvien IP-pakettien IPSec-otsikossa. SPI ei ole täysin yksilöllinen, joten IP-paketin SA tunnistetaan SPI:n ja IP-otsikosta löytyvän kohdeosoitteen avulla. IPSecin käyttöön tarvitaan kaksi tietokantaa. SA-tietokanta (SAD, *Security Assertion Database*) sekä turvallisuus menettelytapa tietokanta (SPD, *Security Policy Database*). SA-tietokanta varastoi SA:n tiedot ja SPD puolestaan määrittää turvallisuus menettelytavat sekä ylläpitää IP-liikenteen ja AP:n välisiä yhteyksiä (mapping). [22, s. 230 – 232.]

### 4.13 Digital Rights Management

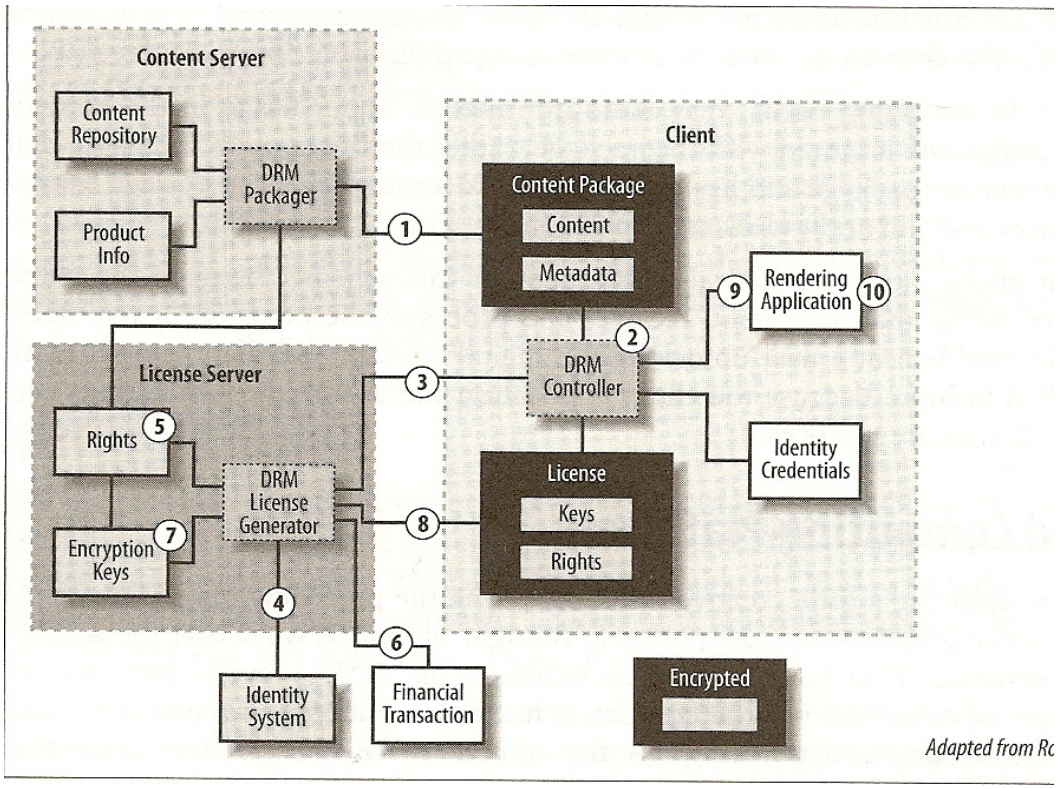
Pääsynhallinta tunnistusta ja valtuutusta käyttäen toimii hyvin silloin kuin ympäristöä pystytään hallitsemaan. Silloin kuin ihmiset tai resurssit ovat yrityksen suoran hallinnan ulkopuolella, ei normaaleista pääsynhallintakeinoista kuitenkaan ole apua. Ongelmana on se, että resurssin esim. cd-levyn tai dokumentin saatuaan käyttäjä voi kopioida sitä rajattomia määriä, lähettää sen yrityksen kilpailijoille tai omille ystävilleen jne. Salauksista tai salasanoista ei ole hyötyä tässä tilanteessa, koska käyttäjän pitää voida käyttää saamaansa resurssia. Tällaisia tilanteita varten on luotu Digital Rights Managment (DRM) eli digitaalisten oikeuksien hallinta. DRM tarjoaa mahdollisuuden hallita, mihin käyttöön digitaalista resurssia voidaan käyttää. Tämä ei kuitenkaan ole yksinkertainen asia, sillä vaikka DRM:stä on huomattavaa hyötyä yrityksille, niin se on myös yksi suurimmista kiistakysymyksistä. Tämä johtuu siitä, että esim. musiikkilevyn ostanut asiakas haluaa jakaa ostamansa musiikin ystäviensä kanssa (esim. kopioimalla sen ystävilleen), kun taas yritys haluaa, että jokainen ostaisi oman levyn yritykseltä ja DRM:ä käytetään mm. musiikkilevyjen kopioinnin estämiseen. Tämän takia asiakkaat suhtautuvat yleensä kielteisesti DRM-tuotteisiin.

DRM:llä on paljon muitakin käyttöjä kuin pelkästään musiikin ja videoiden suojaaminen ja siitä olisi joissain tilanteissa hyötyä meille kaikille, sillä DRM:n avulla voitaisiin esim. antaa pankkikortin numero verkkokauppaan ja antaa verkkokaupalla oikeudet veloittaa tilistä yksittäisiä maksuja, mutta samalla estää verkkopankkia tallentamasta tai lähettämästä pankkikorttisi numero minnekään. DRM-järjestelmillä tulisi olla mm. seuraavanlaisia ominaisuuksia:

- hyvä turvallisuus
- oikeuksien erottaminen varsinaisista tiedoista
- oikeuksien valvonta (katselu, tulostus, muuttaminen)
- tuki sekä online että offline töille.
- dynaaminen oikeuksien jakaminen ja peruminen.

Käytännössä missään olemassa olevassa DRM-järjestelmässä ei ole kaikkia vaadittavia ominaisuuksia, vaan niihin on valittu yrityksen kannalta tärkeimmät ominaisuudet.

DRM-arkkitehtuurissa on kolme pääasiallista toimijaa: asiakas, oikeudet myöntävä lisenssipalvelin ja sisältöpalvelin, jossa haluttu materiaali on. Asiakas (client) on DRM:n tapauksessa sovellus, joka toimii digitaalista resurssia haluavan osapuolen puolesta. Sisältöpalvelin on sovellus, joka toimii digitaalisia resursseja tarjoavan tahon puolesta. Lisenssipalvelin on puolestaan sovellus, joka toimii digitaalisen resurssin oikeuksia hallussa pitävän tahon puolesta. Kuvassa 12 näkyy DRM:n arkkitehtuuri ja toiminta. [10, s. 89 - 92.]




*Kuva 12: DRM-arkkitehtuuri [10, s. 93.]*

Aluksi asiakas (käyttäjän laitteella oleva ohjelmisto) pyytää käyttäjän puolesta tiettyä resurssia sisältöpalvelimelta (kohta 1 kuvassa 12). Sisältöpalvelin luo paketin pyydetystä resurssista käyttäen hyväkseen tavaran säilytyspaikkaa ja tuotteen tietoja (esim. hinta). Tuotteen säilytyspaikka (content repository) voi olla osa sisältöpalvelinta tai sitten oma erillinen tavaran hallintajärjestelmä. Haluttu tuote kuljetetaan suojatusta paketissa, joka sisältää sekä tuotteen että metadatan (ohjelmisto tieto, joka sisältää kuvauksen tallennetuista tiedoista).

Metadata sisältää yleensä seuraavat tiedot: otsikko, tekijä, oikeuksien omistaja sekä muut tarvittavat tiedot tuotteesta sekä tiedot, jotka uniikisti tunnistavat kyseisen tuotteen. DRM-ohjain (2) ottaa yhteyttä lisenssipalvelimeen (3) saadakseen oikeudet hankkimalleen tuotteelle ja lähettää tälle tiedot tuotteen paketista sekä sen toimittaneesta tahosta. Lisenssipalvelin tarkistaa tunnistusvarmenteen (4) käyttäen hyväkseen jotain tarjolla olevaa tunnistusjärjestelmää, jonka jälkeen se kääntyy oikeustietokannan (5) puoleen saadakseen valtuutuksen paketin sisällön käyttöön. Oikeudet tuotteen käyttöön annetaan esim. XML tai XrML (XML-based rights managment language) -tiedostona. Mikäli käyttäjän haluama tuote on maksullinen hoidetaan maksaminen (6) oikeuksien myöntämisen jälkeen. Kun lisenssin myöntäminen on hyväksytty, salausavaimet (7) haetaan paketin tunnistustietojen perusteella ja näitä salausavaimia käytetään lisenssin luomisessa. Avain lähetetään takaisin DRM-ohjaimelle (8) salattuna pakettina, joka sisältää ilmoituksen oikeuksista sekä paketin purkamiseen vaadittavat avaimet. Lisenssi on salattu avainten ja siten koko paketin suojaamiseksi. DRM-ohjain purkaa lisenssin salauksen ja käyttää lisenssistä saamiaan avaimia sisällön purkamiseen. Sisältö voidaan lähettää ohjelmalle (9) sen katselua varten (10). Käyttäjä voi tallentaa sisällön käyttöön liittyvät tiedot. Käyttötiedot tallennetaan paketin kanssa, jotta käyttörajoituksia noudatettaisiin vaikka sisältö siirtyisi yhdeltä käyttäjältä toiselle. Asiakasohjelmistolla on tärkeä osa tässä kokonaisuudessa, koska ohjelmisto (ei käyttäjä) saa ja huolehtii avaimista. Tämä estää osaltaan tuotteiden väärinkäyttöä, sillä tuotteen käyttämiseen tarvitaan tuotteen käyttöoikeudet hankkinut ohjelma. Yksi kaikkein tärkeimmistä DRM:n osista on määritysoikeudet (Specifying Rights), joiden avulla pystytään määrittämään ja valvomaan oikeuksia. Oikeudet ovat erikoistapaus valtuutuksesta ja suurin osa valtuutuksiin liittyvistä asioista pätevät myös DRM:ään. Valtuutuksen ja DRM:n ero on siinä, että DRM rajoittaa oikeuksia paljon hienovaraisemmin (finer-grained) mittakaavassa kuin normaali valtuutusjärjestelmä. [10, s. 92 – 95.]

DRM-järjestelmä ei suinkaan ole aukoton, vaan sitä on melko helppoa kiertää. Esim. kun musiikkitiedosto on laitettu CD-levylle, se voidaan muuttaa toiseen muotoon esim. MP3, ilman DRM:ä, koska DRM:ä ei voi siirtää CD:lle.

Erilaiset suojausongelmat, joita DRM:ssä on, ovat johtaneet lukuisiin pyyntöihin luotettavasta tietokonealustasta (Trusted Platform Module), joka varmistaisi DRM-clientin toiminnan ympäristössä joka estäisi DRM-materiaalin väärinkäytökset. Perusideana on lyhyesti, että tietokoneessa ei pysty käyttämään ohjelmia tai dokumentteja, joita ei ole varmennettu ”luotettavan tahon” toimesta [27]. Tällaisen luotettavan tietokonealustan perusajatuksena olisi suojata jokainen koneenosa (mukaan lukien näppäimistä ja hiiri) väärinkäytöksiltä. Ideana on että kun esim. CD-levyltä yritetään soittaa musiikkia järjestelmä tarkistaa ensin luotettavalta tietokonealustalta, onko käyttäjällä lupa soittaa kyseistä musiikkia annettujen oikeuksien perusteella. Tällainen suojaus vaatisi todella läheistä yhteistyötä laite- ja ohjelmistovalmistajien välillä, sillä kaikkien koneesta löytyvien osien ja ohjelmien pitää tukea luotettavaa tietokone alustaa. Osien luotettavuus tarkistetaan aina koneen käynnistyksen yhdeydessä [27]. [10, s. 95].

# 5 SÄHKÖISTEN PALVELUIDEN TARJOAMINEN MOBIILIVERKON KAUTTA

Mobiiliverkon kautta tarjottavia Internet-palveluja ovat mm. musiikin ja videoleikkeiden lataus, sähköposti, chat (keskustelu sivustot), VoIP (*Voice over Internet Protocol*), paikannus- ja Internet-selainten käyttö. Verkkopalvelut ovat palveluita, jotka käyttävät verkkotekniikkaa ja käyttävät XML-pohjaista kommunikointia HTTP:n yli. Yleisin XML-pohjaisessa HTTP:n yli käytävässä viestinnässä käytetty tekniikkaa lienee SOAP. Yksi verkkopalvelujen perusideoista on käyttää standardoitua kommunikointi ympäristöä, yleensä WSDL:ä (*Web Service Defination Language*), joka mahdollistaa eri käyttäjien pääsyn palveluihin yhtenäisellä tavalla. Käyttäjät voivat käyttää WSDL-kuvausta kommunikoidessaan verkkopalveluja tarjoavan palvelimen kanssa. [28.]

Mobiililaitteille kehitetyt sovellukset voidaan pääpiirteittäin jakaa kahteen luokkaan, web-sovelluksiin ja mobiililaitteisiin ajettaviin sovelluksiin. Näiden lisäksi on joitain sovelluksia, joita käytetään mm. soittoäänien ja mobiilipelien lataamiseen päätelaitteeseen.

Mobiilisovellusten kehittäminen on haasteellista johtuen mm. laitteiden asettamista teknisistä rajoituksista (esim. pieni näyttö, pienempi muistikapasiteetti ja suorituskapasiteetti tietokoneisiin verrattuna, rajoittunut näppäimistö jne.), erilaisista standardeista, tiedonsiirtoprotokollista, verkkoteknologioista, päätelaitteiden nopeasta uusiutumisvauhdista, eri ohjelmistoalustojen tarpeista sekä eri mobiililaitteiden fyysisten ja teknisten ominaisuuksien välisistä eroista. Mobiilit web-sovellukset muistuttavat arkkitehtuuriltaan normaaleja tietokoneen kautta käytettäviä verkkosovelluksia. Mobiililaitteiden selainasiakasohjelma ottaa yhteyttä palvelinkoneessa ajettavaan palveluohjelmaan ja pyytää haluamaansa palvelua, joka sitten näytetään käyttäjälle käyttäjän oman käyttöliittymän välityksellä, jonka joko selainohjelma tai palvelin on muuttanut mobiililaitteella esitettävään muotoon. Nykyään tärkein mobiilien web-sovellusten toteuttamiseen käytettävä sovellusympäristö on todennäköisesti XHTML-kuvauskieli (*eXtensible Hypertext Markup Language*), sillä suurin osa nykyisistä mobiililaitteista tukee XHTML-sivujen selaamiseen soveltuva selainohjelmistoa. XHTML todennäköisesti syrjäyttää aikaisemmat erikoistuneemmat sovellusympäristöt.

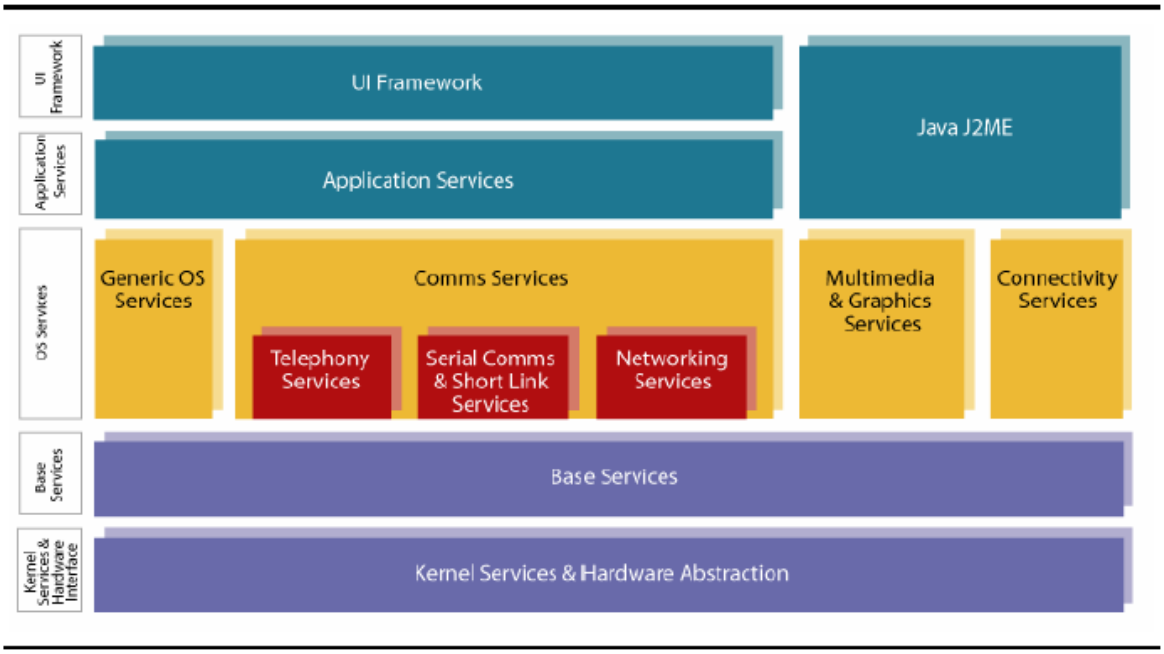
XHTML-kuvauskielelle on kehitetty myös oma mobiiliversio XHTML Mobile Profile, joka sisältää keskeisimmät osat XHTML:stä. Mobiililaitteisiin ajettavat sovellukset ovat puolestaan sovelluksia, joiden ero web-sovelluksiin on se, että ne asennetaan ja ajetaan omina prosesseinaan mobiililaitteessa ja niillä on omat käyttöliittymänsä. Mobiilisovellusten toteuttamiseen on olemassa monia eri ohjelmistoalustoja ja ohjelmistokehyksiä. Tärkeimpiä mobiilialustoja ovat esim. Symbian ja Java. [29, s. 4 – 6.]

## 5.1 Symbian

Symbian on tällä hetkellä johtava älypuhelinten ohjelmistoalusta ja sitä käyttävät tuotteissaan mm. SonyEricsson, Nokia ja NTT DoCoMo. Käyttöjärjestelmänä Symbian tarjoaa monipuolisen kehitysympäristön, ja sen käyttöjärjestelmän perusrakenne noudattaa kerrosarkkitehtuuria ja siinä on viisi kerrosta (kuva 13):

- Ydin- ja laitteistointegraatiokerros (kernel service & hardware integration) huolehtii käyttöjärjestelmän ydintoiminnoista ja tarjoaa laitteistoajurit.
- Perustoimintokerros (base service) sisältää tiedostojärjestelmä rajapinnat (esim. muistikorttiin) ja matalan tason sovelluskehityskehyksen.
- Käyttöjärjestelmäpalvelukerros (operating system service) sisältää käyttöjärjestelmän keskeiset toiminnot esim. verkkoyhteydet sekä grafiikka- ja multimediajärjestelmät.
- Sovelluspalvelukerros (application service) tarjoaa viestien, numeroiden ja yhteystietojen hallintaan liittyvät sovellusmoottorit (application engines).
- Käyttöliittymäkehys (UI framework) sisältää tuen erilaisten käyttöliittymäratkaisujen toteuttamiseen, ja se koostuu kahdesta kerroksesta, sovelluskehyksestä ja työkaluista.

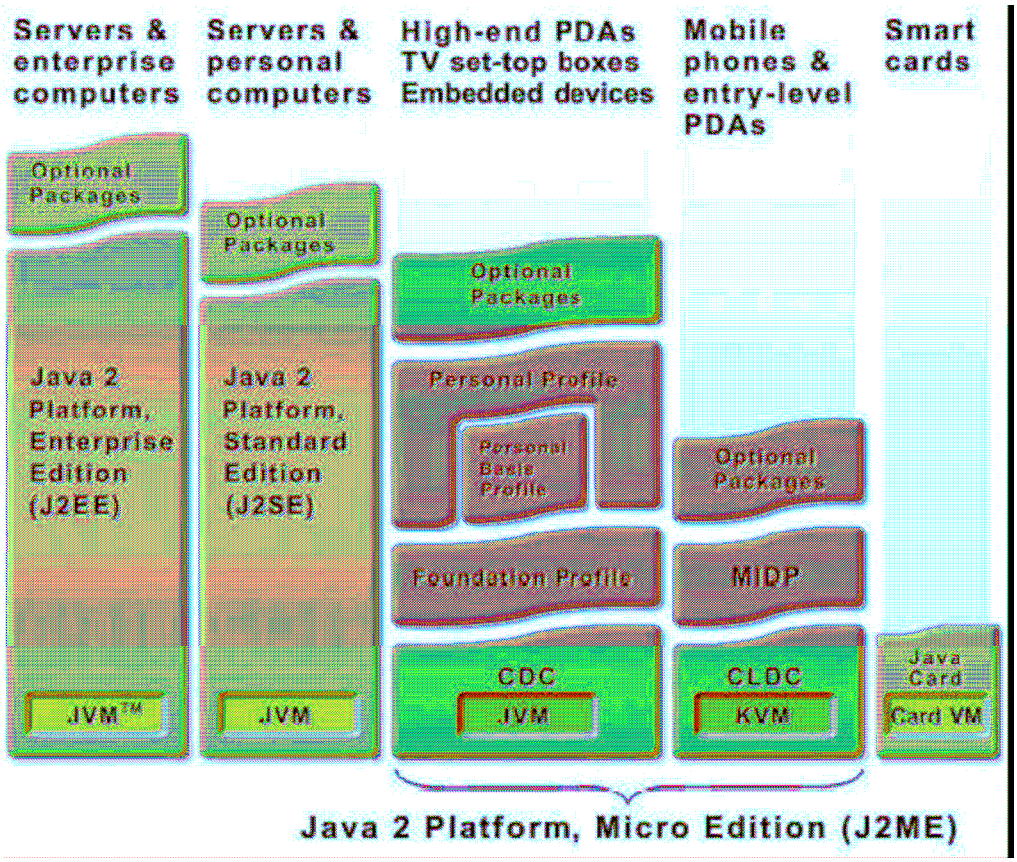
Käyttöjärjestelmään kuuluu myös Java-sovellusten ajamisen mahdollistava virtuaalikoneajoympäristö. Tärkeimmät kerrokset ovat käyttöliittymien sovelluskehityksen kannalta Javan mahdollistava ajoympäristö sekä käyttöliittymäkehys. Käyttöliittymäkehyksen keskeisempiä ominaisuuksia on mm. ikkunointi järjestelmä, sovellusten välinen tehtävävetoinen linkitys ja tapahtumavetoinen käyttöliittymä arkkitehtuuri. [29, s. 6 – 7.]




*Kuva 13: Symbianin käyttöjärjestelmän toiminnallinen kuvaus [29, s.7].*

## 5.2 Java mobiilipalveluissa

Java 2 Micro Edition (J2ME) on J2SE:stä (Java 2 Standard Edition) kehitetty standardi, joka on tarkoitettu Java-alustaksi ensisijaisesti kännyköille, kämmentietokoneille, digisovittimille ja muille resursseiltaan rajallisiin laitteisiin. J2ME koostuu muiden Java-versioiden tavoin Java Community Prosess -ohjelman määrittelemistä standardeista rajapinnoista. J2ME on jaettu erilaisten kohdelaitteiden suuren määrän vuoksi kahteen osaan eli konfiguraatioon. Konfiguraatiot koostuvat virtuaalikoneesta ja pienestä määrästä luokkia, joilla määritellään vähimmäisvaatimukset konfiguraatiota tukeville laitteille (kuva 14).




*Kuva 14: Java-arkkitehtuuri [30].*

Java-käyttöympäristö luodaan konfiguraatioiden, profiilien ja lisäosien avulla, joiden muodostamaa käyttöympäristöä voi joko käyttää sellaisenaan tai luoda itse omia sovelluksia siihen. Konfiguraatioilla luodaan pienimmät yhteiset tekijät laiteryhmien välille. Matkapuhelimissa käytetään konfiguraationa tällä hetkellä yleisemmin CLCD:tä (Connected Limited Device Conficuration), joka on tarkoitettu pieniresurssisille laitteille. CLDC määrää minimikokoonpanon sitä käyttäville laitteille. Lisäksi se määrittelee muun muassa Javan virtuaalikoneen toiminnot, java.lang- ja java.util-kirjastot, verkkoyhteydet, tietoturvamallin ja versioinnin eri kielille. Sen sijaan konfiguraatio ei puutu esimerkiksi sovellusten elinkaareen tai käyttöliittymään. CLDC tarjoaa sovelluksen käyttöön vain niin sanotut perusluokat. Se ei tarjoa omia luokkia, vaan kaikki CLDC:n luokat ovat joko suoraan kopioituja tai karsittuja J2SE:n luokkia. Seuraava mobiili Javan sukupolvi tulee kuitenkin käyttämään CDC (*Connected Device Conficuration*) konfiguraatiota, joka on tarkoitettu kehittyneemmille laitteille ja joka sisältää useampia toimintoja CLCD:n verrattuna. CDC on kuitenkin yhteensopiva myös vanhojen CLCD-sovellusten kanssa.

Tämä sukupolvi tulee keskittymään enemmän palveluihin, jolloin mahdollistetaan samojen palvelujen jakaminen eri ohjelmien kesken sekä uusien palvelutoimintojen tuominen puhelimeen myös puhelimen valmistuksen jälkeen. Lisäksi uusi sukupolvi tarjoaa rajapinnan matkapuhelimen omaan ulkoasuun, jolloin Java ei ota niin paljon kantaa sovellusten ulkoasuun kuin nykyiset versiot. Profiililla täsmennetään samaa konfiguraatiota käytettävien laitteiden toimintaa täysin erityyppisissä tehtävissä. Yleisin käytetty protokolla matkapuhelimissa on MIDP (*Mobile Information Device Profile*), joka sisältää peruskirjastot kannettaville laitteille. MIDP-profiili antaa määritelmät sovelluksen elinkaaren hallinnalle, pysyväismuistille, käyttöliittymille, verkkoyhteyksille, ajastimille, sovellusten jakamiselle ja laskuttamiselle, tietoturvamallille, palvelimelta tehtävällä työntötekniikalle sekä äänille. Lisäpaketeilla puolestaan lisätään laiteeseen toiminnallisuuksia, joita konfiguraatiot tai profiilit eivät tarjoa. [30.]

MIDP-käyttöliittymä arkkitehtuuri koostuu kahdesta rinnakkaisesta päätasosta, korkean- ja matalantason käyttöliittymäkomponenteista. Korkeantason käyttöliittymäkomponentteihin kuuluu mm. syöttökentät, listat ja painikkeet, joita tarvitaan sovelluksen käyttöliittymän toteutukseen sekä tapahtumakäsittelijät, joiden avulla komponentit liitetään sovellukseen. Korkeantason komponenteilla toteutettu sovellus voidaan periaatteessa siirtää suoraan toiseen MIDP-määrittelyä tukevaan laitteeseen. Korkeantason komponentteja käyttäessä tapahtumakäsittely tapahtuu komponentteihin kytkettyjen kuuntelijoiden avulla. Matalantason komponentit tarjoavat puolestaan mahdollisuuden toteuttaa sovelluksen käyttöliittymä itsenäisesti esim. omien käyttöliittymä komponenttien avulla. Matalantason komponenteilla toteutettuja sovelluksia ovat esim. mobiilipelit. Riippumatta siitä, kumman tason komponentteja käytetään, täytyy jokaisen MIDP-sovelluksen käyttää näyttö-luokan metodeja, joilla saadaan yhteys MIDP-käyttöliittymäarkkitehtuurin piirtomekanismeihin. [29, s. 13.]

## 5.3 Mobiilipalveluita

Mobiiliverkon kautta voidaan tarjota useita erilaisia sähköisiä palveluita. Tässä kappaleessa käydään läpi kaksi eri sähköisen verkon kautta tarjottavaa mobiilipalvelua.

### *5.3.1 VoIP*

VoIP eli IP-puheeksi kutsutaan tekniikkaa, jonka välityksellä ääntä ja videokuvaa voidaan siirtää reaaliaikaisesti verkossa. Puhe ja videokuva muutetaan analogisesta digitaaliseen muotoon ja lähetetään paketteina verkon yli. VoIP-yhteiskäytöntä koostu merkinanto (käytetään puhelun muodostukseen) ja puheprotokollista (äänen siirrossa käytetyt protokollat), ja puhelun muodostamista varten on kaksi erilaista standardia, jotka eivät ole keskenään yhteensopivia: H.323, joka on vanhempi ja laajempi sekä SIP (*Session Initiation Protocol*), joka on yksinkertaisempi, mutta laajenee koko ajan. Nykyisen kehityksen valossa SIP-protokollan käyttö on yleistymässä ja syrjäyttämässä H.323:n. [31.]

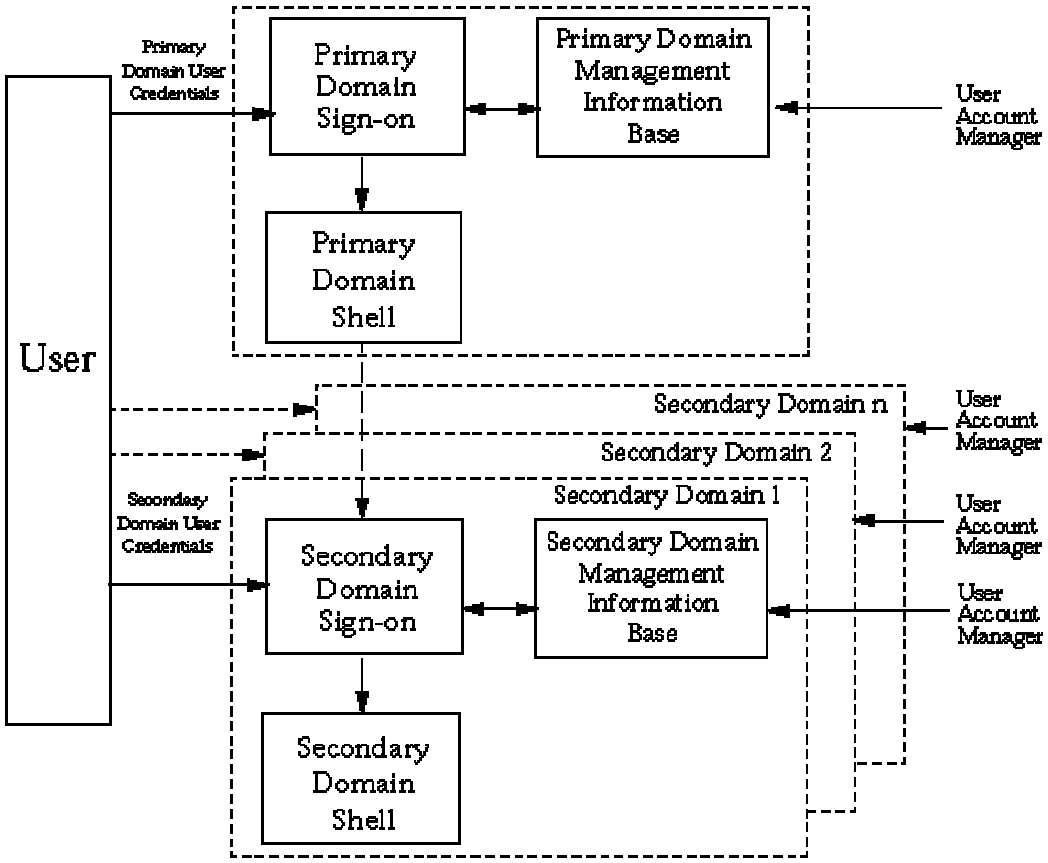
### *5.3.2 Paikannukseen perustuvat palvelut*

Paikannukseen perustuvat palvelut ymmärretään ensisijaisesti erilaisten mobiilipäätelaitteiden (esim. kännyköiden) kautta käytettäviksi palveluiksi. Tyypillisiä esimerkkejä paikallispalveluista ovat reittiopastus ja sijaintiin liittyvän tiedon välittäminen. Paikannukseen perustuville palveluille yhteistä on se, että automaattisesti ja mahdollisimman reaaliaikaisesti tapahtuva mobiililaitteen paikan määritys kuuluu oleellisena osana palveluun ja palvelut perustuvat paikannustietojen hyödyntämiseen. Mobiileissa paikannusmenetelmistä voidaan erottaa kaksi keskeistä menetelmää, satelliittipaikannus ja matkapuhelinverkkopaikannus. Satelliittipaikannus perustuu maata kiertävien satelliittien muodostamaan järjestelmään. Tunnetuin käytössä oleva satelliittipaikannus on Yhdysvaltojen puolustushallinnon ylläpitämä GPS (*Global Position System*). Matkapuhelinverkkopaikannus puolestaan perustuu käyttäjän sijainnin määrittämiseen matkapuhelinverkon tukiasemien perusteella. Käyttäjältä ei puhelimen lisäksi vaadita muita lisälaitteita vaan sijainnin määritteleminen tapahtuu teleoperaattorin toimesta. [32]

# 6 SINGLE SIGN ON

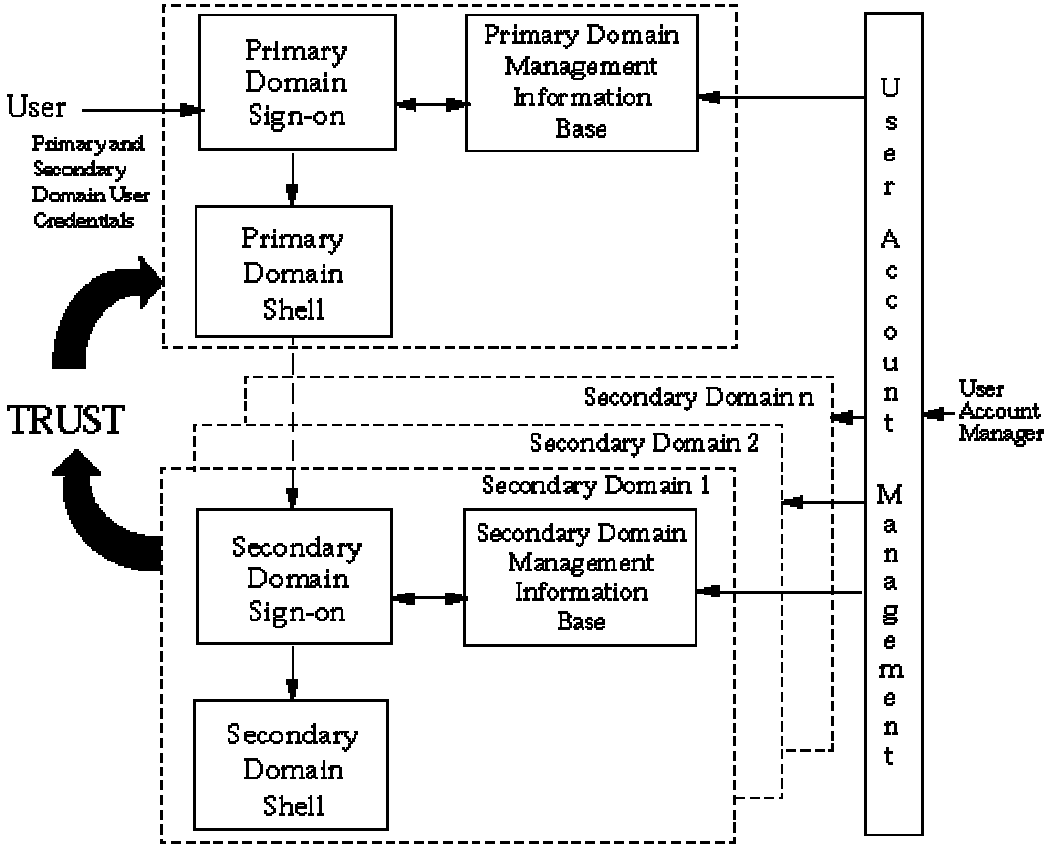
## 6.1 Yleistä kertakirjautumisesta

Tavallisesti tietoliikennejärjestelmässä jokainen palvelu toimii erillisenä domainina, jolloin käyttäjät joutuvat aina erikseen kirjautumaan kuhunkin palveluun. Kuvassa 15 näkyy, miten kirjautuminen hoidetaan tällaisessa järjestelmässä. Aluksi kirjaudutaan main domainiin, jonka jälkeen kirjaudutaan erikseen subdomaineihin aina, kun kyseisen domainin palvelua halutaan käyttää. Tavallisesti tällainen kokonaisuus on koottu eri osista, joissa jokainen laite toimii omana yksikkönään ja joista jokaisella on myös oma turvallisuus vaatimus. Tämä aiheuttaa käyttäjille ja järjestelmänvalvojille tarpeettomia vaikeuksia. Käyttäjillä on todennäköisemmin monta eri käyttäjänimeä ja salasanaa, joitten kaikkien muistaminen voi osoittautua melko hankalaksi, etenkin harvemmin käytettyjen tunnusten kohdalla. Lisäksi erillinen kirjautuminen palveluihin aiheuttaa väkisinkin ajanhukkaa. Järjestelmänvalvojien puolestaan täytyy pitää yllä kaikista eri käyttäjätunnuksista ja salasanoista koostuvaa kirjastoa, jotta he pystyisivät huolehtimaan verkon turvallisuudesta sekä pitämään käyttäjätunnustiedot ajan tasalla. [33.]




*Kuva 15: Perinteinen kirjautumistapa moneen eri palveluun [33].*

Kyseisten ongelmien välttämiseksi on Liberty Alliance kehittänyt single sign on -kirjautuminen, eli suomeksi kertakirjautumisen. Kertakirjautumisessa käyttäjä kirjautuu vain päädomainiin, joka kerää kirjautumisen yhteydessä tarvittavat tiedot, jotta subdomaineihin kirjautuminen onnistuisi. Päätaso välittää sitten tunnistustiedot automaattisesti subdomaineille ilman että käyttäjän täytyy itse kirjautua erikseen jokaiseen palveluun. Tässä tapauksessa subdomainien pitää luottaa päädomainilta saamiinsa tietoihin. Single sign on -kirjautumisen periaate näkyy kuvasta 16. [33.]




Kuva 16: Kertakirjautuminen useampaan palveluun [33].

Kertakirjautuminen hyötyjä ovat:

- Käyttäjällä menee vähemmän aikaa kun ei tarvitse kirjautua jokaiseen palveluun erikseen.
- Turvallisuus paranee, kun käyttäjän ei enää tarvitse muistaa ja käsitellä montaa eri salasanaa ja käyttäjätunnusta.
- Järjestelmänvalvojan tarvitsee pitää yllä pienempää käyttäjänimi ja salasana kirjastoa, jolloin käyttäjien lisääminen ja poistaminen sekä käyttäjien oikeuksien muuttaminen nopeutuu.
- Turvallisuus paranee, kun järjestelmänvalvoja pystyy paremmin valvomaan käyttäjätilejä sekä tarvittaessa rajoittamaan tai estämään yksittäisen käyttäjän pääsyn palveluihin. [33.]

Kertakirjautumisen haittapuolia ovat:

- Kertakirjautuminen on vaikeaa, kallista ja aikaa vievää toteuttaa jo toiminnassa oleviin sovelluksiin, joissa on perinteinen kirjautumistyyli.
- Jos ulkopuolinen saa tietoonsa jonkin käyttäjänimen ja salasanan, niin hän pääsee käsiksi kaikkiin käyttäjän käyttämiin palveluihin.
- Koska kaikki käyttäjätiedot on koottu samaan paikkaan siitä tulee oivallinen kohde hakkereille. Eli hakkerointiyritykset kohdistuvat kaikki samaan tiedostoon. [34.]

## 6.2 Kertakirjautumisluokat

Kertakirjautumistapoihin kuuluu neljä luokkaa:

• Local pseudo SSO (paikallinen luonnoskertakirjautuminen)

• Proxy-based pseudo SSO (Välityspalvelin perustuva luonnoskertakirjautuminen)

• Local true SSO (Paikallinen tosikertakirjautuminen)

• Proxy-based true SSO (Välityspalvelimeen perustuva tosikertakirjautuminen).

Pseudo SSO -järjestelmät käyttävät komponentteja, jotka automaattisesti suorittavat eri toimittajien tunnistusmekanismit esim. käyttäjätunnuksen, salasanan tai varmenteen. Istunnon alussa käyttäjä tunnistautuu kerran pseudo SSO -komponentin kanssa. Pseudo SSO -järjestelmässä komponentit ja palvelut tunnistavat toisensa erikseen aina tarvittaessa. Paikallinen pseudo SSO tarkoittaa sitä, että SSO-komponentit on tallennettu käyttäjän omalle koneelle. Palvelin-pohjaisessa pseudo SSO:ssa puolestaan SSO:ssa tarvittavat tiedot ovat ulkoisella palvelimella, ja tunnistus tapahtuu palvelimen ja palvelun tarjoajan välillä. Pseudo SSO toimii käytännössä paikkana, jonne salasanat, usein salatussa muodossa, säilötään. Tietoihin pääsee käsiksi vain master-salasanan avulla, jonka kirjoittamisen jälkeen muut salasanat näkyvät selkokielellä. Pseudo SSO:n ongelma onkin juuri riippuvuus master-salasanasta. Jos master-salasana unohtuu, ei salaisiin tietoihin päästä enää käsiksi. Toisaalta mikäli hyökkääjä saa selville master-salasanan hän pääsee käsiksi kaikkiin muihinkin salasanoihin.

True SSO:ssa järjestelmässä käyttäjä tunnistautuu palveluihin ASP:n (*Authentication Service Provider*) avulla. Tällöin käyttäjä tunnistautuu kerran ASP:hen, jolla on yhteys kaikkiin palveluntarjoajiin. Käyttäjätunnuksia ja salasanaa ei anneta palveluntarjoajille niin kuin pseudo SSO:ssa vaan palvelun tarjoajia informoidaan tunnistuksesta tunnistusjulistusten (authentication assertion) kautta. Paikallinen true SSO käyttää ASP:nä luotettua komponenttia, joka on käyttäjän koneen sisällä, kun taas palvelinpohjaisessa ASP on palvelimella. Proxy-based True SSO:n vahvuus verrattuna Pseudo SSO:n on tuki muillekin tunnistusmetodeille (esim. älykortin ja PIN-koodin yhdistelmä) kuin master-salasanalle. Proxy-pohjaisen SSO:n etuna on myös se, että koska kaikki tiedot on keskitetty yhdelle verkkopalvelimelle, se pystytään suojaamaan hyvin palomuurin ja muiden suojaustekniikoiden avulla.

Pseudopohjaiset SSO:t ovat läpinäkyviä (transparent), eli ne eivät vaadi muutoksia ohjelmistoon tai palvelun tarjoajan järjestelmään. True SSO puolestaan ei ole läpinäkyvä, eli se vaatii muutoksia järjestelmään tai ohjelmiin toimiakseen. [35.]

# 7 OPEN MOBILE ALLIANCE

OMA (*Open Mobile Alliance*) on eri mobiili alalla olevien yritysten muodostama yhteenliittymä, jonka tarkoituksena on kehittää avoimia standardeja kännykkäteollisuuden käyttöön. OMA:n jäseninä ovat niin suuret kännykänvalmistajat kuten Nokia, Siemens, Sony Ericsson, Ericsson ja Samsung, mobiilioperaattorit esim. Vodafone, Telefónica ja Orange että ohjelmistontuottajat kuten IBM, Microsoft, Symbian sekä Sun Microsystem. [36.]

## 7.1 Single sign on mobiiliverkossa

Mobiilitilaaja voi käyttää monia palveluita, joista kaikki eivät kuulu verkon operaattorin luotettuun domainiin. OMA Web Service Enabler Release (OWSER) on OMA:n kehittämä standardi, joka määrittelee keinot, jolla OMA:n sovelluksia voidaan julkistaa, löytää ja käyttää, verkkopalvelu teknologioita käyttäen.

OWSER perustuu Liberty Allianssin määrittelemiin standardeihin erityisesti seuraaviin kolmeen standardiin: Liberty Protocols and Schema Specification (määrittelee XML-mallin, sekä SSO:n ja muut seuraavassa kappaleessa läpikäydyt palvelut), Liberty Bindings and Profiles Specification (määrittelee Liberty Allianssin protokollien ja viestien sitomisen ja profiilit) ja Liberty Authentication Context Specification (määrittää syntaksit tunnistusyhteys todistuksien (statement) määrittämiseen sekä listaa alustavasti tunnistusyhteysluokat). [37, s. 10.]

Palveluntarjoaja luottaa käyttäjän tunnistuksesta toiseen osapuolen (tunnisteen tarjoajaan), joten palveluntarjoaja saattaa haluta lisätietoja siitä kuinka tunnistus tapahtuu, jotta he voivat paremmin arvioida tunnistukseen liittyviä riskejä ja sen luotettavuutta. Authentication Context Spesifikaatiossa määriteltävät tunnistusyhteys todistukset ja tunnistusyhteysluokat antavat palveluntarjoajalle lisätietoja (esim. kuinka alkuperäinen käyttäjäntunnistus tapahtui tai mitä mekanismejä tiedon tallentamiseen ja suojaamiseen käytetään) siitä kuinka käyttäjän tunnistus tapahtuu. Nämä ylimääräiseet tiedot lähetetään palveluntarjoajalle tunnistusyhteys todistuksella (Authentication Context statement), joka on XML-muotoinen dokumentti. Todistus lähetetään palveluntarjoajalle joko <AuthnResponce> viestin sisällä tai sitten viestissä on viite tunnistusyhteys todistukseen. Erilaisia tunnistusviitteitä (authentication context) on teoriassa ääretön määrä, ja niiden kaikkien analysoiminen on lähes mahdotonta. Libertyn määrittelemät tunnistusyhteysluokat jakavat tunnistusviitteet pienemmiksi luokiksi, joita on helpompi ymmärtää ja analysoida. Tunnistusyhteysluokat myös helpottavat tunnistuksen tarjoajan ja palvelun tarjoajan välistä kanssakäymistä, koska luokkien avulla on helpompaa sopia mitkä tunnistustavat ovat hyväksyttäviä. Tunnistusyhteysluokkia ovat mm. InternetProtocolPassword-luokka, MobileTwoFactorUnregistered-luokka sekä SmartcardPKI-luokka [38. s.6 - 7, 9].

OWSER NI (Oma Web Service Network Identity Enabler) -standardi on jaettu kolmeen osaan OWSER NI FF (OWSER Network Identity Federation Framework), OWSER NI FSF (OWSER Network Identity Web Service Framework) ja OWSER NI AD(OWSER Network Identity Architecture document) [37, s. 10].

## 7.2 OWSER Identity Federation Framework

OWSER NI standardin Federarion Framework osa tarjoaa määrittelyt osista, joita tarvitaan Identity Federationin toimintaan Libertyn sallivassa (Liberty enabled) verkkopalveluympäristössä. Sen tarkoitus on tarjota OMA-verkkopalveluiden suunnittelijoille ratkaisut normaaleihin toimintoihin, käyttäen verkkopalveluja tekniikoita. Ilman tällaista runkoa, jokainen OMA-verkkopalveluiden tarjoaja ratkaisisi kyseiset asiat omalla tavallaan, jolloin minkäänlaista yhteensopivuutta ei olisi. Tällaista normaalia toimivuutta, jota käytetään monissa verkkopalveluissa tarjoamaan yhdenmukaistettua käyttöä tilalle tai tiedolle käyttäjään liittyen, kutsutaan verkkohenkilöllisyydeksi (*Network Identity*). Tällainen palvelu on mm. Single Sign On (SSO). Standardiin kuuluu Identity Federation, Single Sign-On, Name Registration, Authentication Context, Federation Termination sekä Single Sign-Out, ja ne perustuvat Liberty Alliancen 1.1 standardeihin, joissa nämä edellä mainitut protokollat on standardisoitu. Mobiiliverkko operaattorit toimivat OWSER standardissa yleensä henkilötunnuksen tarjoajana, joka luo, ylläpitää ja käsittelee henkilöllisyys tietoja sekä tarjoaa tunnistusvakuuden kyseiseen tahoon luottaville palveluntarjoajille. [39 s. 10.]

*Identity Federation ja SSO*

Identity Federation ja SSO perustuvat pyyntö/vastaus-protokollaan. SSO käynnistyy kun käyttäjä lähettää http-pyynnön palveluntarjoajalle. Palveluntarjoaja lähettää henkilötunnisteen tarjoajalle <lib.AuthnRequest> viestin, eli tunnistuspyyntö viestin, jossa pyydetään henkilötunnisteen tarjoajaa antamaan tunnistusvakuutus palveluntarjoajalle. Henkilötunnisteen tarjoaja vastaa tunnistusvastausviestillä <lib:AuthnResponse>, joka sisältää tunnistusvakuutuksen palveluntarjoajalle. Tunnistusvakuudet saatuaan palveluntarjoaja tekee päätöksen mitä palveluja se sallii käyttäjän käyttää. OWSER Federation Framework-standardissa käsitellään kolme eri protokollaa, joilla Federation Identity ja SSO voidaan toteuttaa: Browser Artifact Profile, Browser POST profile ja Liberty-Enabled Client/Proxy profile. Browser Artifact eli selain artefaktiprofiili luottaa tunnistusartefaktin (SAML artefakti) käyttöön.

Palveluntarjoajan täytyy saada (dereference) tunnistusvakuutus (eli tunistusartefakti) henkilöllisyyden tarjoajalta, jotta palveluntarjoaja voi tunnistaa käyttäjän. Tämän profiilin tukeminen on välttämätöntä henkilöllisyyden ja palvelun tarjoajien kannalta, mikäli he haluavat käyttää verkkohenkilöllisyyttä. Browser POST-profiilissa tunnistus tapahtuu ilman SAML-artefaktia. Browser POST-profiilin tukeminen ei ole palvelun ja henkilöllisyydentarjoajien kannalta välttämätöntä, joskin henkilöllisyyden tarjoajan olisi hyvä tukea sitä, jotta hän voisi keskustella sellaisten palveluntarjoajien kanssa, jotka kyseistä profiilia käyttävät. Liberty-Enabled Client/Proxy profiili määrittelee Libertyn sallivan käyttäjä agentin (Liberty Enabled User Agent), palvelun ja henkilöllisyyden tarjoajan väliset yhteydet. Liberty-enabled user eli Libertynsalliva käyttäjä on käyttäjän käyttäjäagentti, jolla joko on tai joka tietää, kuinka saada tiedot henkilöllisyydentarjoajasta, jota käyttäjä haluaa käyttää palveluntarjoajan kanssa luodakseen verkkohenkilöllisyyspohjaisen palvelun kuten SSO. Browser artifact ja client/Proxy profiilit käyttävät viestinnässään apuna SOAP:a. [39, s. 11 – 13.]

*Name Registration*

Identity Federationin aikana henkilöllisyyden tarjoaja luo läpinäkymättömän sangan (opaque handle), joka toimii käyttäjän alkuperäisenä nimitunnisteena, jota sekä palvelun että henkilötunnisteen tarjoaja käyttävät viitatessaan käyttäjään (Principal) keskinäisessä kommunikoinnissaan. Palvelun ja henkilötunnisteen tarjoaja kommunikoivat keskenään, silloin kun ne tarjoavat käyttäjälle jotain Network Identityyn perustuvaa palvelua kuten SSO:ta. Tätä käyttäjän nimitunnistetta kutsutaan <lib.IdpProvidedNameIdentifier> (henkilötunnisteen tarjoajan antama nimitunniste). [39, s. 15 – 16.]

*Authentication Context*

Authentication Context eli tunnistusyhteys määritellään lisätiedoksi tunnistusvakuutuksen lisäksi. Tunnistusyhteys saattaa sisältää mm. tunnistusmekanismeja, tiedontallennusmekanismeja ja valtuuksien suojausta. [39, s. 17.]

*Single Sign-Out*

Single Sign-Out eli kertauloskirjautuminen tarkoittaa sitä, että kirjaudutaan kerralla ulos kaikista palveluista. Jotta tämä toimisi, palveluntarjoajan pitää lähettää henkilötunnisteen tarjoajalle katkaisupyyntö <lib:LogoutRequest> silloin, kun käyttäjä kirjautuu ulos heidän palvelustaan. Tämän jälkeen henkilötunnisteen tarjoaja lähettää saman katkaisupyynnön kaikille muille palveluntarjoajille, joille henkilötunnisteen tarjoaja tarjoaa tunnistevakuutuksen kyseisestä käyttäjästä. Muut palveluntarjoajat vastaavat <lib:LogoutResponce> viestillä, jolloin yhteys käyttäjän kanssa katkaistaan. Uloskirjautuminen voidaan aloittaa joko palveluntarjoajan tai henkilötunnisteen tarjoajan toimesta. [39, s.17 – 18.]

*Federation Termination Notification*

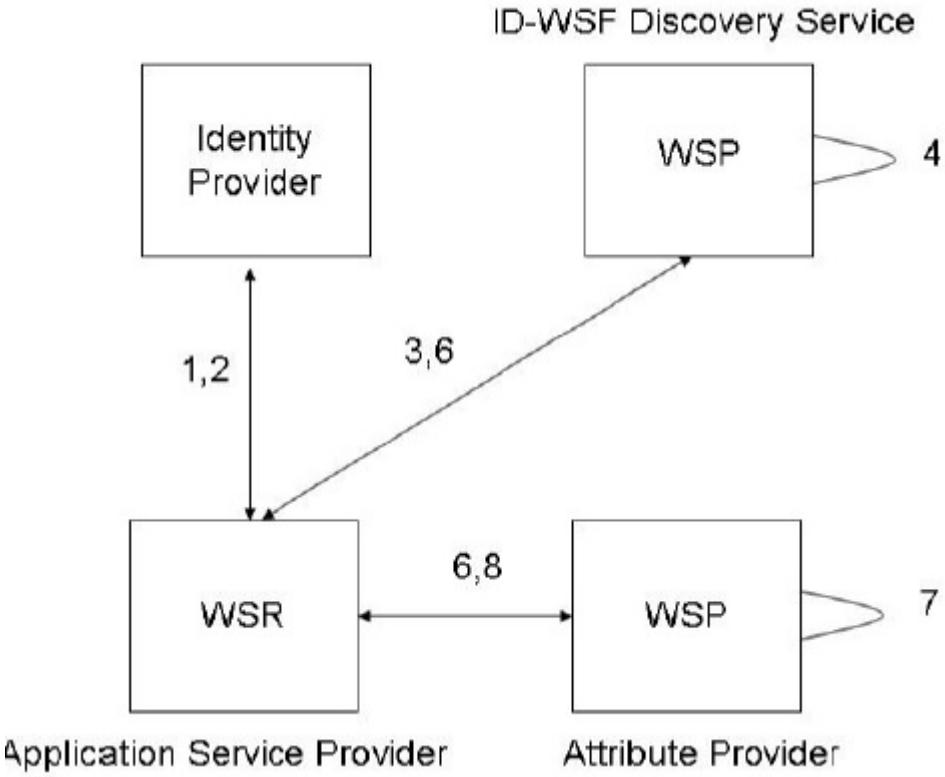
Federation Termination Notification -protokollaa käytetään, kun tilaaja katkaisee Identity Federationin palveluntarjoajan ja henkilötunnisteen tarjoajan välillä. Protokollalla on neljä eri vuorovaikutustilaa. Protokollan vuorovaikutuksen voi aloittaa sekä palvelun että henkilötunnisteen tarjoaja, ja sen toiminta perustuu joko HTTP:n uudelleenohjaustoimintoon tai SOAP/HTTP:n viestien vaihtoon. Sekä palveluntarjoajan että henkilötunnisteen tarjoajan on tuettava SOAP/HTTP-pohjaista palvelua. Lisäksi henkilötunnisteen tarjoajan olisi hyvä tukea HTTP-pohjaista palvelua, jotta se pystyy keskustelemaan HTTP-pohjaista käyttävien palveluntarjoajien kanssa. Protokollan toiminta on identtistä huolimatta siitä, aloittaako protokollan palveluntarjoaja vai henkilötunnisteen tarjoaja. [39, s. 19.]

## 7.3 OWSER NI Web Service Framework

OWSER NI Web Service Framework tarjoaa määritteet osista, joita tarvitaan toteuttamaan NI-RD (MWS Identity Management Requirement) -pohjaisen käyttäjän yksityisiä ominaisuuksia (esim. käyttäjän henkilökohtaiset tiedot) käyttävän, ja käyttäjän yksityisyyttä suojaavan verkkopalveluympäristön. OWSER-verkkopalvelu kehyksessä määritellään kolme eri tapaa käyttää OWSER-verkkopalveluita.

OWSER:n perusverkkopalveluita voi käyttää ilman OWSER NI:tä (OWSER Network Identity), jolloin verkkopalvelun tilaajan ja tarjoajan välinen keskustelu käydään jollain OWSER-julkaisussa määritellyllä perusprotokollalla esim. SOAP:lla. OWSER NI:tä käyttäessä pohjana on Liberty Allianssin Federation Framework, jossa määritellään mm. palvelun tarjoaja, tunnistuksen tarjoaja sekä niihin liittyvät protokollat (mm. nimen rekisteröinti ja Identity Federation), jotka yhdessä mahdollistavat yhdistyneet käyttäjätunnisteet (Federated user identities) useammalle palveluntarjoajalle yhden luottamuskentän (Circle of Trust) sisällä. Luottamuskentällä tarkoitetaan yhtä tai useampaa palveluntarjoajaa sekä tunnistuksen tarjoajaa, joilla on liiketoimintasuhde ja toiminnallinen yksimielisyys keskenään ja joiden välillä käyttäjä voi toimia turvallisesti ja ilmeisen saumattomasti kaupankäynnin aikana. Verkkohenkilöllisyys (Network Identity) tarjoaa myös mahdollisuuden kertakirjautumisen (SSO) käyttämiseen luottamuskentän alueella. Kolmas OWSER-verkkopalvelujen käyttömahdollisuus on Single Sign-on -palvelun (SSOS, *Single Sign- On Service*) käyttö Libertyn sallivassa verkkopalveluympäristössä. Tällöin Liberty-tunnistuksen tarjoaja (Liberty Identity Provider) voi tarjota kertakirjautumispalvelua verkkopalvelun pyytäjälle (WSR, Web Service Requester), mikäli tunnistuksen tarjoajan laitteet tukevat ID-WSF (Identity Web Service Framework) tunnistuspalvelua sekä ID-WSF-kertakirjautumista. Verkkopalvelun pyytäjä ei voi itse suoraan käyttää Liberty Alliassin määrittelemää kertakirjautumista, koska ne on suunniteltu selainpohjaiseen ympäristöön, jossa kertakirjautuminen palveluntarjoajalle toteutettiin uudelleen ohjaamalla selain palveluntarjoajan sivuille tunnistuspyynnön avulla. Koska uudelleen ohjauspalvelu puuttuu verkkopalvelun pyytäjä autentikoi suoraan tunnistuksen tarjoajan kanssa ja käyttää tunnistuksesta saatuja tietoja silloin, kun tunnistuksen tarjoajan luottamuskenttään kuuluvien palveluntarjoajien tarvitsee tunnistaa käyttäjä. [40, s. 5, 11 – 14.]

Kuvassa 17 näkyy esimerkki tiedon kulusta, jota tarvitaan kun halutaan päästä käsiksi käyttäjän tunnistus ominaisuuksiin. Esimerkkiä on yksinkertaistettu ja mm. salausta ja kirjautumista ei ole tässä esimerkissä käsitelty.



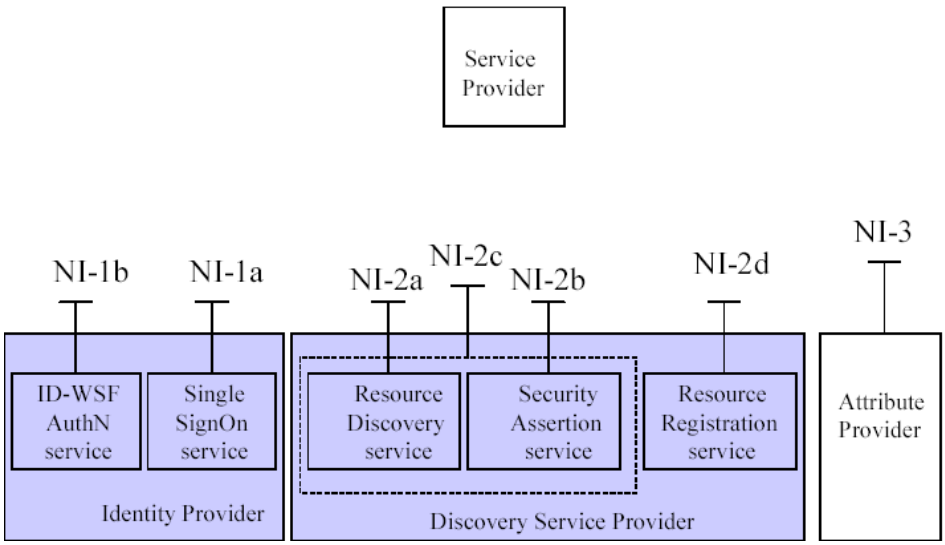

*Kuva 17: Tarvittavat vuorovaikutus suhteet, jotta päästään käsiksi käyttäjän tunnistusominaisuuksiin [40, s.13].*

1. Verkkopalvelun pyytäjä (WSR) eli ohjelmistopalvelun tarjoaja tunnistaa käyttäjän lähettämällä ID-FF (Identity Federation Framework) -tunnistuspyynnön henkilöllisyyden tarjoajalle (Identity Provider).
2. Henkilöllisyyden tarjoaja palauttaa ID-FF SAML (Security AssertionMarkup Language) -vakuutuksen, jossa tunnistetaan käyttäjä WSR:lle ja tarjotaan ID-WSF resursseja käyttäjän ID-WSF tunnistuspalvelulle (Discovery Service).
3. Verkkopalvelun pyytäjä käyttää tarjottuja resursseja sekä SAML -tunnistusvakuutusta käyttäjän tunnistukseen ja yhdistämiseen ID-WSF -löytämispalvelun (Discovery Service) ja pyytää ID-WSF resurssin tarjoamisen ja SAML -vakuutuksen (SAML assertion) ominaisuuden tarjoajaa varten (Attribute Provider), jolla on mahdollisuus luoda käyttäjän tarvitsemat tunnistusominaisuudet.
4. ID-WSF tunnistuspalvelu tunnistaa verkkopalvelun pyytäjän ja luo vastausviestin, joka sisältää pyydetyt ID-WSF resurssien tarjoamisen ja SAML-vakuutuksen ominaisuuksien tarjoajaa varten.

5. Tunnistuspalvelu palauttaa luomansa viestin palvelun pyytäjälle, tämä on kuvassa 17 numeroitu 6:ksi (WSF:n ja WSP:n (Web Service Provider) välinen kuutosen pitäisi siis olla numero viisi).
6. Verkkopalvelun pyytäjä käyttää saamiaan resurssin tarjoamista ja SAML' vakuutusta tunnistukseen ominaisuuden tarjoajan kanssa ja käyttäjän tarvitsemien ominaisuuksien pyytämiseen.
7. Ominaisuuden tarjoaja tunnistaa verkkopalvelun pyytäjän ja luo viestin, jossa on pyydetyt ominaisuudet.
8. Ominaisuuksien tarjoaja palauttaa viestin verkkopalvelun pyytäjälle. [40, s. 13.]

## 7.4 OWSER-arkkitehtuuri

OWSER-arkkitehtuuristandardi kertoo loogiset toimijat sekä rajapinnat, joita tarvitaan verkkopalvelun löytämiseen ja käyttöön, niin että käyttäjän ominaisuudet (Attributes) olisi suojattu. Ominaisuuksilla tarkoitetaan mm. käyttäjän henkilökohtaisia tietoja tai mieltymyksiä. [41, s. 10.]




*Kuva 18: Korkean tason ID-WSF arkkitehtuuri [41, s. 12].*

OWSER käyttää Liberty Allianssin tekemää ID-WSF (Identity Web Service Framework) -arkkitehtuuria, joka näkyy kuvassa 18. Liberty Allianssin arkkitehtuuri tarjoaa mahdollisuuden suojata käyttäjän yksityisyyttä käyttäjän keskustellessa eri palveluntarjoajin kanssa.

Tämä tehdään antamalla käyttäjälle käyttäjänimi (pseudonym) erikseen jokaista palveluntarjoajaa kohden. Palveluntarjoajat ankkuroituvat ensisijaiseen palveluntarjoajaan (henkilötunnisteen tarjoajaan), joka on taho, joka pystyy kartoittamaan jokaisen käyttäjänimen tiettyyn käyttäjään. Palveluntarjoajan ei näin tarvitse hoitaa muuta kuin paikallisen alueen tunnistustietoja heidän omassa järjestelmässään, eikä heidän tarvitse ylläpitää linkkejä muiden palveluntarjoajien välillä, joihin käyttäjällä voi olla yhteys. ID-WSF-arkkitehtuuriin kuuluu seuraavat palvelut:

NI-1-rajapinta tukee palvelutarjoajan ja henkilötunnisteen tarjoajan välistä keskustelua.

NI-1a-rajapinta tukee Single Sign-On -palveluja. Tämän rajapinnan tarjoamia funktioita ovat mm. Single Sign-Out, tunnistusyhteyden (Authentication Context) asettaminen hyväksytylle tilaajan tunnistusvahvistukselle sekä Palveluntarjoajan tunnistusvahvistuksen palauttaminen jo olemassa olevasta käyttäjän tunnistuksesta.

N1-1b-rajapinta tarjoaa välineet verkkopalvelu pyynnölle luoda tunnistus istunto henkilötunnisteen tarjoajan kanssa. Istunnossa vaihdettuja tietoja käytetään SSO-palvelun aloittamiseen.

NI-2-rajapinta tukee palveluntarjoajan sekä löytämispalvelun välistä keskustelua.

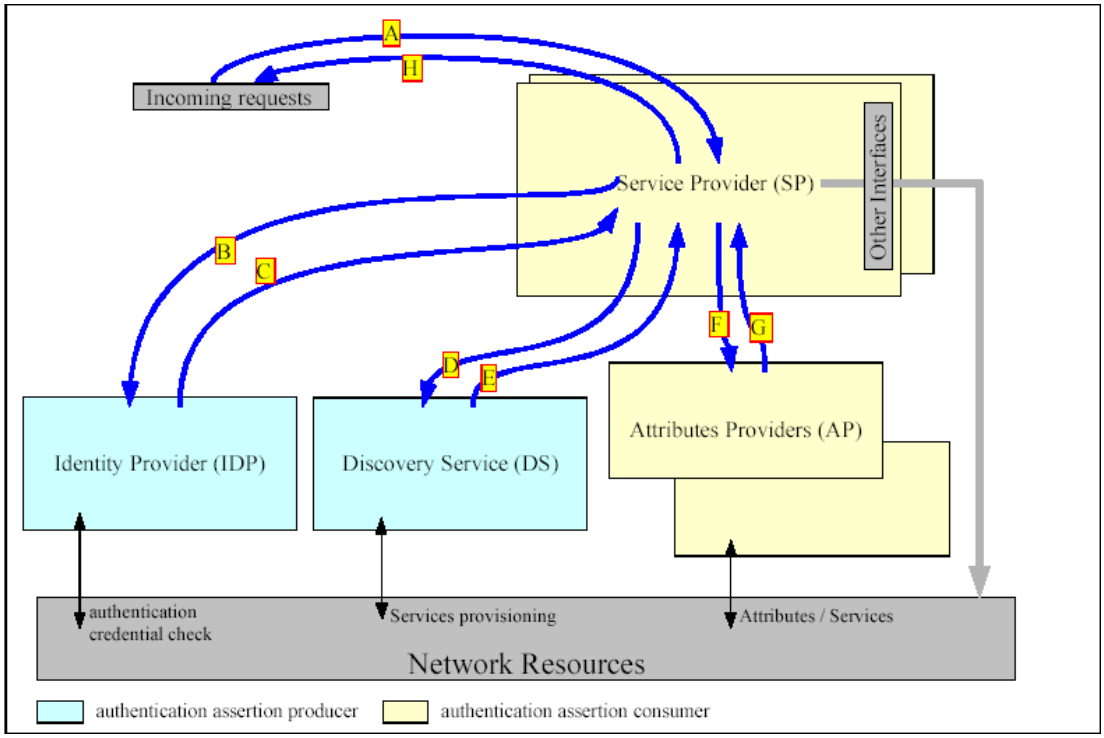
NI-2a-rajapinta tarjoaa käyttäjän ominaisuuden tarjoajan (Attribute Provider) löytämispalvelun. Ominaisuuden tarjoaja voi olla vaikkapa pankkipalvelu.

NI-2b-rajapinta vastaanottaa autentikointitunnisteen (authentication token) käyttäjän ominaisuuden tarjoajalta.

NI-2c tarjoaa sekä NI-2a että NI-2b:n tarjoamat palvelut, jolloin tietojen pyytäjä pystyy löytämään ominaisuuden tarjoajan ja saamaan autentikointi tunnisteen ominaisuuden tarjoajalta yhden pyyntö operaation aikana (palvelun löytäminen ja tunnisteen saaminen tapahtuu yhdellä viestillä, kun normaalisti palveluntarjoaja lähetettäisi ensin etsintäpyynnön ja vasta siihen vastauksen saatuaan autentikointipyynnön).

NI-2d tarjoaa rekisteröinnin käyttäjän ominaisuuspalveluille.

NI-3-rajapinta tukee palvelun tarjoajan ja kenen tahansa ominaisuuden tarjoajan välistä keskustelua. NI-3 rajapinnassa voi lukea, kirjoittaa, muokata ja tuhota ominaisuuksia (Attributes). Lisäksi siinä on käyttäjän vuorovaikutus palvelu, jossa käyttäjän suostumusta pyydetään, ennen ominaisuuksien jakamista muille. [41, s. 12 – 14.]




*Kuva 19: Tiedon kulku käyttäjän pyytäessä palvelua palveluntarjoajalta [41, s. 15].*

Yleinen tiedonkulku ID-WSF-arkkitehtuurissa silloin, kun käyttäjä pyytää palvelua palveluntarjoajalta on seuraavanlainen (kuva 19).

A: Käyttäjä lähettää palvelupyynnön palveluntarjoajalle.

B: Palveluntarjoaja tutkii tilaajan tunnistustiedot henkilötunnisteen tarjoajalta.

C: Henkilötunnisteen tarjoaja vastaa palveluntarjoajalle tunnistusvahvistuksella, jossa ilmoitetaan tilaajan tunnistustiedot, sekä tarvittaessa tarpeelliset esilataus tiedot (bootstrap information) tilaajan löytämispalveluun pääsemiseksi.

D: Palveluntarjoaja käyttää edellisessä kohdassa saatuja esilataustietoja kysyäkseen tilaajan käyttämästä löytämispalvelusta haluttua tilaajan ominaisuuden tarjoajaa. (esim. soittoääntä ostaessa kysytään tilaajan pankkipalvelua)

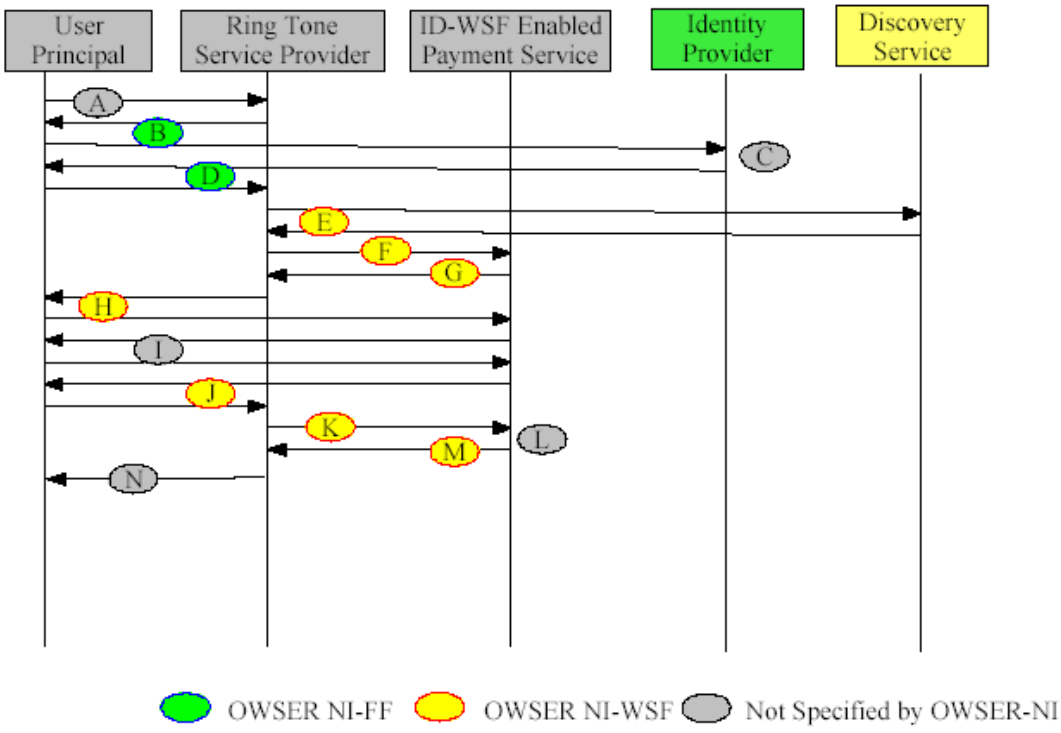
E: Löytämispalvelu palauttaa tunnistamisenvahvistuksen, jota palveluntarjoaja käyttää ominaisuuden tarjoajaan kirjautumisessa.

F: Palveluntarjoaja pyytää ominaisuuksia (attributes) tilaajan ominaisuuden tarjoajalta.

G: Ominaisuuden tarjoaja palauttaa pyydetyt tiedot.

H: Palveluntarjoaja vastaa tilaajan tilauspyyntöön. [41, s. 15 – 16.]

Kuvassa 20 näkyy käytännön esimerkki tiedon kulusta yhden luottamuskentän tapauksessa sekä missä OMA:n standardissa kyseisen viestin sisältö on määritelty. Yhden luottamuskentän tapauksessa palveluntarjoaja kuuluu operaattorin (tunnistuksen tarjoajan) luottamuskenttään, jolloin palveluntarjoaja hyväksyy tunnistuksen tarjoajan tunnistusviranomaiseksi.




*Kuva 20: Esimerkki viestinnän kulusta soittoäänen ostamisen yhteydessä. [41, s.18].*

A: Käyttäjä menee palvelun tarjoajan (tässä tapauksessa soittoäänien tarjoajan) sivuille ja tutkii tarjottuja soittoääniä. Tässä kohdassa käyttäjä on vielä tuntematon eli hän ei ole kirjautunut sisään.

B: Kun käyttäjä valitsee haluamansa soittoäänen ja painaa ”osta”-nappia hankkiakseen soittoäänen itselleen, palveluntarjoaja uudelleenohjaa käyttäjän tunnisteentarjoajan luokse.

C: Tunnistuspalvelun tarjoaja tunnistaa käyttäjän (Tarkistamalla käyttäjän IP-osoitteen RADIUS-palvelimelta).

D: Tunnistuksen tarjoaja palauttaa palvelun tarjoajalle tunnistus vakuutuksen (käyttäjän selaimen uudelleenohjauksella), jossa vahvistetaan autentikoinnin tila ja esilataus tiedot käyttäjä löytämispalvelun käynnistämiseksi.

E: Soittoäänien tarjoaja kysyy löytämispalvelulta käyttäjän ID-WSF standardissa määritettyä maksupalvelua (ID-WSF enabled Payment Service), ja saa vastauksena tiedot, joita tarvitaan yhteyden muodostamiseen maksupalvelun kanssa (Maksupalvelu on tässä tapauksessa ominaisuuden tarjoaja).

F: Soittoäänien tarjoaja lähettää seuraavaksi verkkopalvelupyynnön käyttäjän maksupalvelulle saadakseen maksun käyttäjän tilaamasta soittoäänestä.

G: Tässä tapauksessa maksaminen vaatii käyttäjän hyväksynnän, jolloin maksupalvelu palauttaa palveluntarjoajalle ”MUST-INTERACT” SOAP-virheen (SOAP fault), joka sisältää URL-osoitteen, jonne käyttäjän selain pitäisi uudelleen ohjata.

H: Palveluntarjoaja lähettää HTTP uudelleen vastauksen käyttäjälle, joka ohjatun ohjaa käyttäjän hänen maksupalveluunsa (käyttäjä ohjataan kohdassa G saatua URL-osoitteeseen).

I: Maksupalvelu ja käyttäjä neuvottelevat käyttäjän hyväksynnän maksamiselle HTTP pyyntö/vastaus protokollan avulla.

J: Maksupalvelu palauttaa HTTP uudelleenohjaus vastauksen, joka uudelleenohjaa käyttäjän soittoäänien tarjoajan sivuille.

K: Palveluntarjoaja lähettää uudelleen maksupyynnön maksupalvelulle.

L: Maksupalvelu hyväksyy maksupyynnön ja veloittaa käyttäjän tililtä vaaditun summan.

M: Maksupalvelu palauttaa tiedon maksun tilasta (eli että soittoääni on maksettu) soittoäänen tarjoajalle.

N: Soittoäänien tarjoaja lähettää soittoäänen ja maksun tilan käyttäjälle. [41, s. 16 – 17.]

## 7.5 Digital Rights Managment

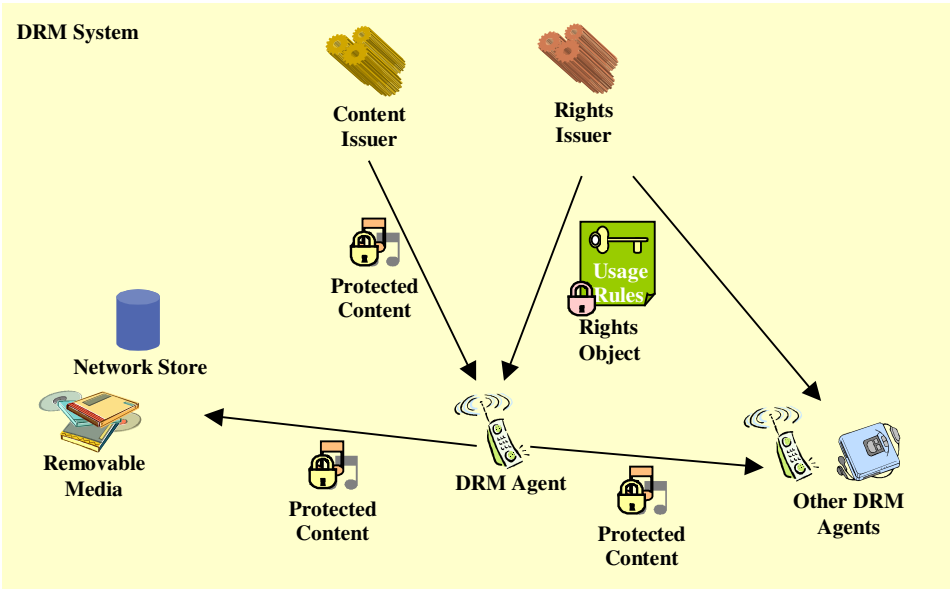
OMA:n DRM (*Digital Rights Managment*) on standardi, jonka tarkoituksena on myöntää käyttöoikeuksia digitaalisesti ladattaville palveluille sekä estää niiden luvaton käyttö. DRM-tuotteet suojataan ennen kuin ne laitetaan verkkoon ja niille luodaan käyttöoikeudet (Rights Object). DRM:n suojaamiin tuotteisiin sisältyy mm. kännykän soittoäänet, pelit, musiikki, videot ja kuvat. DRM-tuotteet voidaan ladata laitteeseen monilla eri tavoilla mm. LAN (*Local Area Network*)/ WLAN (*Wireless Local Area Network*)-verkon kautta tai siirrettävällä medialla (esim. USB-muistitikku), mutta käyttöoikeudet ovat tarkasti valvottuja Rights Objektien avulla. Rights Object XML-dokumetti, joka ilmaisee oikeudet ja rajoitukset liittyen jonkin DRM-sisällön osan käyttöön. XML-dokumentti on määritelty REL-kielellä (Rights Expression Language), joka onmääritelty OMA DRM:ssä. Ilman käyttöoikeuksia DRM-materiaalia ei voi käyttää eikä tuotteita voi käyttää muuhun kun käyttöoikeuksien myöntämiin käyttötarkoituksiin. [42, s. 17.]

## 7.6 Digital Rights Managment -arkkitehtuuri

DRM-arkkitehtuuriin kuuluu:

- DRM-agentti, joka on jonkin luotetun tahon DRM-agenttisofta käyttäjän laitteessa. DRM agentin vastuulla on valvoa DRM sisältöön liittyviä oikeuksien ja rajoituksia, DRM-materiaaliin pääsemistä jne. DRM-agenttiohjelmia ovat esim. Certicom Beep Science ja Philips Trust 2.1. Lisätietoja edellä mainituista agentti ohjelmista löytyy osoitteista http://www.certicom.com/index.php?action=product,beep ja http://www.safenet-inc.com/solutions/dev/wireless_drmagent.asp
- Sisällön toimittaja (Content Issuer) tuottaa DRM-sisällön.
- Oikeuksien jakaja (Rights Issuer) on taho joka jakaa oikeudet ja rajoitukset DRM-materiaaliin sekä luo käyttöoikeudet (Rights Object).
- Käyttäjä käyttää DRM-sisältöä. Käyttäjä pääsee käsiksi DRM- tiedostoihin ainoastaan DRM-agentin kautta.

- off-device storage. DRM-tiedostot on luonnostaan suojattu, joten ne voidaan tallentaa mm. PC:lle, verkkokauppaan tai USB-tikulle. Näin voidaan tehdä mm. jotta tiedostoista olisi varmuuskopio tai jotta kännykän muistia vapautuisi muihin käyttötarkoituksiin.




*Kuva 21: DRM:n toiminnallinen arkkitehtuuri. [43, s.10]*

OMA DRM erottaa DRM-materiaalin ja sen käyttöön liittyvät oikeudet (Rights Object) toisistaan. DRM-materiaalin ja oikeudet voi pyytää joko yhdessä tai erikseen. Esim. käyttäjä ostaa jonkin DRM-osan vaikkapa jonkin pelin. Pelin hinnan maksettuaan käyttäjä lataa verkosta sekä pelin että sen käyttöön oikeuttavat oikeudet samalla kerralla. Myöhemmin jos oikeudet vanhenevat, hän voi saada uudet oikeudet ilman, että hänen täytyy ladata peliä uudestaan verkosta. DRM-agentin tehtävä on valvoa DRM-sisältöön liittyviä oikeuksien ja rajoituksia. Käyttöoikeudet on salatusti sidottu tiettyyn DRM-agenttiin, joten vain kyseinen agentti pääsee käsiksi käyttöoikeuksiin (eli vain laitteella, jossa agentti on päästään käsiksi materiaaliin.). Se että DRM-materiaaliin pääsee käsiksi vain materiaalille myönnetyillä käyttöoikeuksilla mahdollistaa sen, että käyttäjät voivat vaihtaa vapaasti keskenään DRM-materiaalia. Aina uudelle laitteelle asennettaessa täytyy kyseisen laitteen saada materiaalin käyttöoikeudet laitteen DRM-agentille. Käyttöoikeuksien myöntäjä voi myös sitoa materiaalin käyttöoikeudet useammalle DRM- agentille, jotka muodostavat silloin domainin.

Käyttäjät voivat tässä tapauksessa jakaa DRM-materiaalia kaikkien DRM-agenttien kesken, jotka ovat samassa domainissa (eli kaikki agentit, joille on myönnetty käyttöoikeudet materiaaliin). Domainit mahdollistavat mm. käyttäjän pääsyn DRM-materiaaliin useamman koneen kautta ilman, että oikeuksia täytyy hakea jokaiselle koneelle erikseen. [43, s. 9 – 12.]

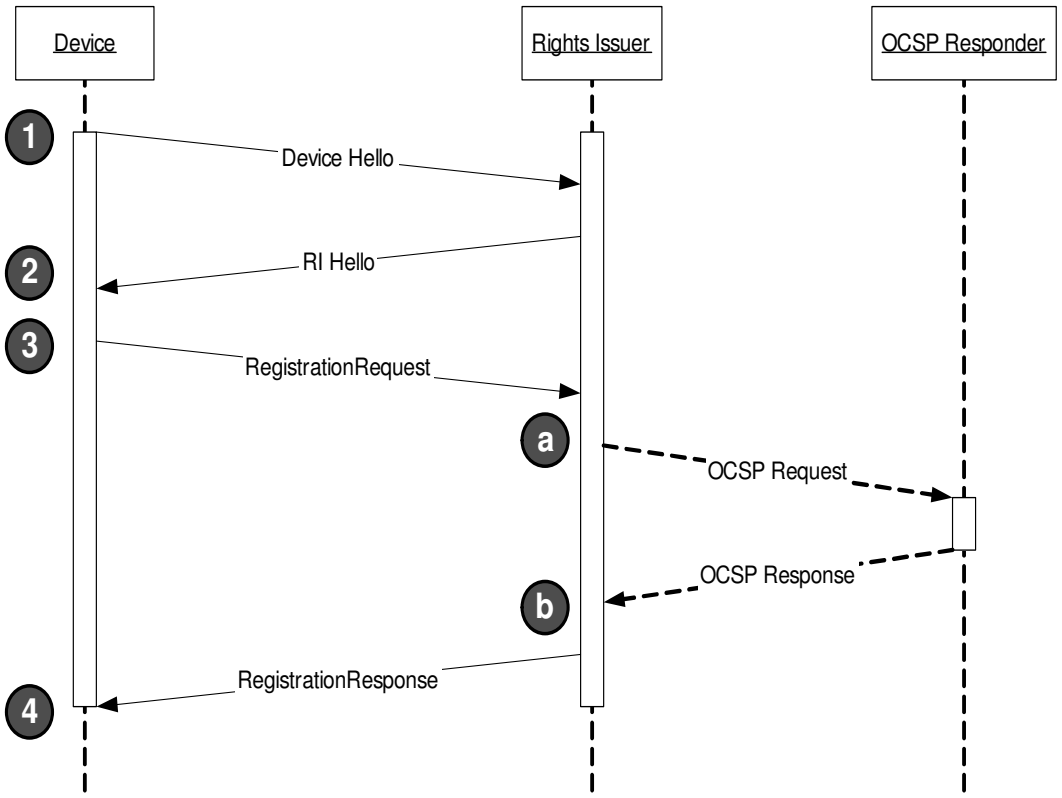
## 7.7 Digital Rights Managment -turvallisuus

Rights Objekt Acquision Protocol (ROAP) on yleisnimitys DRM-turvallisuusprotokollille, jotka toimivat Oikeuksien myöntäjän (Rights Issuer) ja DRM-agentin välillä. Käyttäjällä täytyy olla ostettu ja ladattu materiaali, ennen kuin hän voi hakea ja saada käyttöoikeudet ostamaansa tuotteeseen [40, s. 25]. ROAP-protokollat sisältävät 4-pass-protokollan, kaksi protokollaa, joiden avulla laite pyytää ja hankkii käyttöoikeudet (Rights Objects), nämä ovat 2-pass ja 1-pass -oikeuksien hakuprotokollat. Lisäksi ROAP-protokolliin kuuluuvat myös 2-pass-liittyminen ja 2-pass-yhteyden lopetusprotokollat. [42, s. 19.]

### *7.7.1 4-pass-rekisteröintiprotokolla*

4-pass-rekisteröintiprotokollaan (4-pass registeration protocol) kuuluu turvallisuustietojen vaihto sekä kättely (yhteyden muodostaminen kahden laitteen välille), joka tapahtuu oikeuksien antajan ja DRM-agentin välillä. Tätä protokollaa käytetään yleensä vain ensimmäisellä kerralla, mutta sitä voidaan tarvittaessa käyttää myös myöhemmin, mikäli laitteiden vaihtamia turvallisuus tietoja pitää päivittää tai kun DRM-aika (oikeuksienantajan suojatusta lähteestä saama aika, jota käyttäjä ei pääse muuttamaan) on oikeuksien myöntäjän mielestä epätarkka. Tähän protokollaan kuuluu mm. neuvottelut protokollan parametreista sekä protokollan versiosta, salausalgoritmit, valinnallinen DRM-laitteiden ajan synkronisointi (käytetään jos käyttäjän DRM-aika on oikeuksien myöntäjän mielestä epätarkka), varmenne etuoikeudet sekä varmenteiden vaihto (joka ei ole pakollinen).

Mikäli rekisteröiminen onnistuu, muodostuu oikeuksien myöntäjän konteksti (Rights Issuer Context) laitteeseen, jossa on oikeuksien myöntäjän turvallisuuteen liittyvät parametrit, kuten protokollan versio, sovitut protokolla parametrit sekä varmenne etuoikeudet (certificate preference). Tätä oikeuksien myöntäjän kontekstia tarvitaan, jotta muut ROAP-protokollat toimisivat.




*Kuva 22: 4-pass-rekisteröintiprotokolla [42, s. 20].*

Kuvassa 22 näkyy, mitä 4-pass-protokollan aikana tapahtuu. Aluksi käyttäjän laite lähettää Hello-viestin, johon oikeuksien jakaja vastaa omalla hello viestillään. Tämän jälkeen käyttäjä lähettää rekisteröintipyynnön. Käyttäjän laitteen DRM-aika pitää olla luottamusmallin (trust model) mukainen. Mikäli DRM -ikä on käyttäjän laitteessa väärin lähettää oikeuksien myöntäjä OCSP (*Online Certificate Status Protocol*) pyynnön, jonka vastauksen perusteella käyttäjän kone voi laittaa DRM-aikansa oikeaan aikaan. Protokollan suoritus loppuu kun oikeuksien myöntäjä lähettää käyttäjälle rekisteröinti vastauksen. [42, s. 19 – 20.]

### *7.7.2 2-pass- ja 1-pass-oikeuksien saantiprotokollat*

2-pass-oikeuksien saantiprotokolla (2-pass Rights Object acquisition protocol) on nimensä mukaisesti protokolla, jonka avulla käyttäjä saa hankittua oikeudet DRM-materiaalin käyttöön. Tämä protokolla sisältää molemmanpuoleisen laitteen ja oikeuksien myöntäjän välisen tunnistuksen, eheyssuojatun pyynnön ja kuljetuksen käyttöoikeuksille sekä suojatun siirron salaus avain materiaalille (REK, Rights Encryption Key, katso kpl 7.7.5), joita tarvitaan käyttöoikeuksien käsittelyssä. Protokollan aluksi käyttäjä lähettää käyttöoikeuspyynnön, johon käyttöoikeuksien myöntäjä lähettää vastauksen. Vastauksen saannin jälkeen protokollan suoritus loppuu. Käyttöoikeuksien myöntäjä voi lähettää OCSP-pyynnön, mikäli käyttäjän laitteen DRM-aika on väärä. OCSP-pyyntö toimii samalla tavoin kuin 4-pass-protokollassa. 1-pass-oikeuksien saantiprotokolla (1-pass Rights Object acquisition protocol) eroaa 2-pass-protokollasta siten, että siinä käyttäjä ei lähetä käyttöoikeus pyyntöä oikeuksien myöntäjälle. Yksi käyttötarkoitus tälle on oikeuksien jako tietyin aikavälein esim. sisällön tilauksen tukeminen. 1-pass protokolla on käytännössä 2-pass-protokollan viimeinen viesti. [42, s. 20 – 21.]

### *7.7.3 2-pass-liittyminen ja 2-pass-yhteyden lopetusprotokollat*

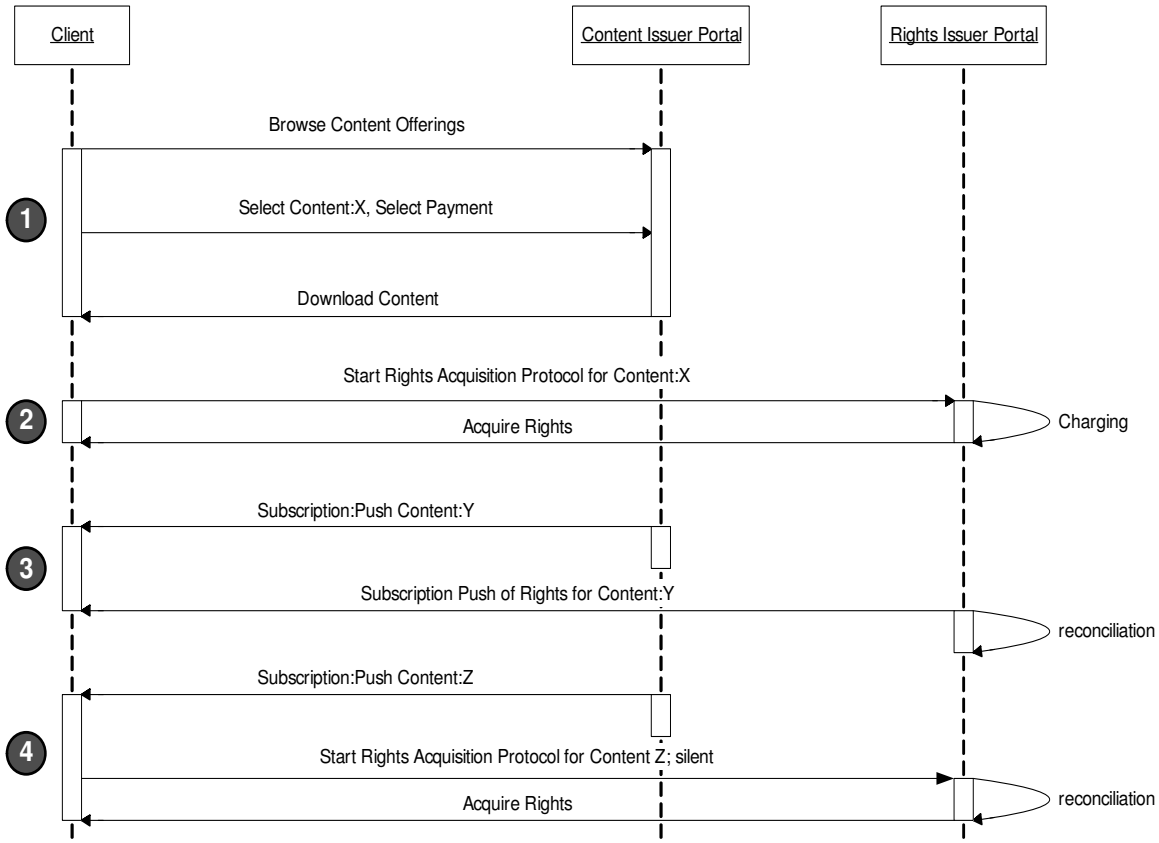
2-pass liittymisprotokolla (2-pass Join Domain Protocol) huolehtii nimensä mukaisesti koneen liittymisestä oikeuksien myöntäjän ylläpitämään domainiin. Protokolla toimii samaan tapaan kuin 2-pass oikeuksien saanti protokolla eli ensin käyttäjä lähettää Domainiin liittymispyynnön ja oikeuksien myöntäjä vastaa siihen. 2-pass-liittymisprotokollassa on myös OCPS-pyynnön mahdollisuus. Tämä toimii samalla tavoin kuin yllä kuvatuissa protokollissa. Kun liittymisprotokolla on käyty onnistuneesti läpi, luodaan domain konteksti, johon sisältyy kyseisen domainin turvallisuuteen kuuluvat tiedot kuten Domain avain. Domain konteksti on välttämätön, jotta kone voisi asentaa ja käyttää domainin käyttöoikeuksien. 2-pass-yhteyden lopetusprotokolla (2-pass Leave Domain Protocol) on protokolla, jonka avulla asiakas poistuu domainista. Käyttöjä lähettää Leave domain -viestin, johon oikeuksien jakaja vastaa. [42, s. 22.]

### *7.7.4 DRM-materiaalin lataaminen*

DRM:n luottamusmalli (Trust model) perustuu PKI:n käyttöön. Luottamus mallin tärkeimmät yksilöt ovat varmenteen antaja (CA), laite ja oikeuksien myöntäjä. Tunnistus- ja avaimensiirtoprotokollat vaativat, että oikeuksien myöntäjä ja asiakkaan laite pystyvät kumpikin tunnistamaan toisensa.

Molemmanpuoleinen tunnistus toteutetaan edellä kuvatun ROAP:n (Rights Objekt Acquision Protocol) avulla. Kuvassa 23 näkyy normaalissa DRM-materiaalin lataustapaukset.

1. Aluksi käyttäjä menee sisällön tuottajan sivuille ja valitsee sieltä haluamansa tiedoston. Käyttäjä voi tässä yhteydessä valita myös maksutavan, jonka jälkeen käyttäjä lataa tiedoston.
2. Sen jälkeen käyttäjä ottaa yhteyden oikeuksien myöntäjään (oikeuksien myöntäjän URL-osoite DRM materiaalin otsikosta) ja käynnistää ROAP-protokollan. ROAP-protokollan suorittamisen jälkeen käyttäjä saa käyttöoikeudet ostamaansa materiaaliin. Tätä mallia kutsutaan vetomalliksi.
3. Vaihtoehtoinen tapa kohdissa 1 ja 2 esitetylle toimintatavalle on tapa, jossa käyttäjä on tehnyt tilaus- ja laskutussopimuksen oikeuksien myöntäjän kanssa. Tämä mahdollistaa sen, että oikeuksien myöntäjä voi suoraan tietyin väliajoin ”työntää” sekä materiaalin että siihen liittyvät oikeudet käyttäjälle. Tätä mallia kutsutaan työntömalliksi
4. Kolmas mahdollisuus on se, että sisällön tuottaja lähettää käyttäjälle materiaalia, mutta ei ennestään tunne käyttäjää. Tällainen tilanne voi olla, jos esim. toinen käyttäjä on ostanut jonkin tuotteen lahjaksi kohteena olevalle käyttäjälle. Tällöin materiaalin saamisen jälkeen käyttäjä hakee käyttöoikeudet oikeuksien myöntäjältä samalla tavalla kuin kohdassa 2. Tämä malli on nimeltään osittainen ”vedon aloittava työntö” (push-initiated pull) malli. [43, s. 17 – 19.]




*Kuva 23: Normaali DRM-materiaalin lataus ”veto” (pull) ja ”työntö” (push) mallit [43, s. 18].*

### *7.7.5 DRM-materiaalin lataamiseen liittyvät turvallisuus toimenpiteet*

DRM-materiaalin lataamiseen liittyvät normaalit vaiheet voidaan tiivistää seuraaviin vaiheisiin:

1. Sisällön pakkaus: Sisältö pakataan suojatun sisällön pakkauksen (DCF, *DRM Content Format*) avulla ja suojataan symmetrisen sisällön salausavaimella (*Content Encryption Key*, CEK) ennen kuin se laitetaan verkkoon.

2. DRM-agentin tunnistus: DRM-agentin tunnistus tapahtuu PKI:n avulla eli kaikilla DRM-agenteilla on oma yksityinen ja julkinen avain sekä varmenne, johon sisältyvät perustietojen lisäksi myös mm. laitteen malli, laitteen tekijä, ohjelmistoversio sekä sarjanumero.

3. Oikeuksien (Rights Object) luominen: luodaan XML (eXtensible Markup Language) -dokumentti, jossa määritellään DRM-materiaalin käyttöoikeudet. Käyttöoikeudet sisältävät myös CEK:n, jota tarvitaan sisällön purkamiseen. CEK-avain estää DRM-materiaalin käytön ilman materiaaliin liittyviä oikeuksia.

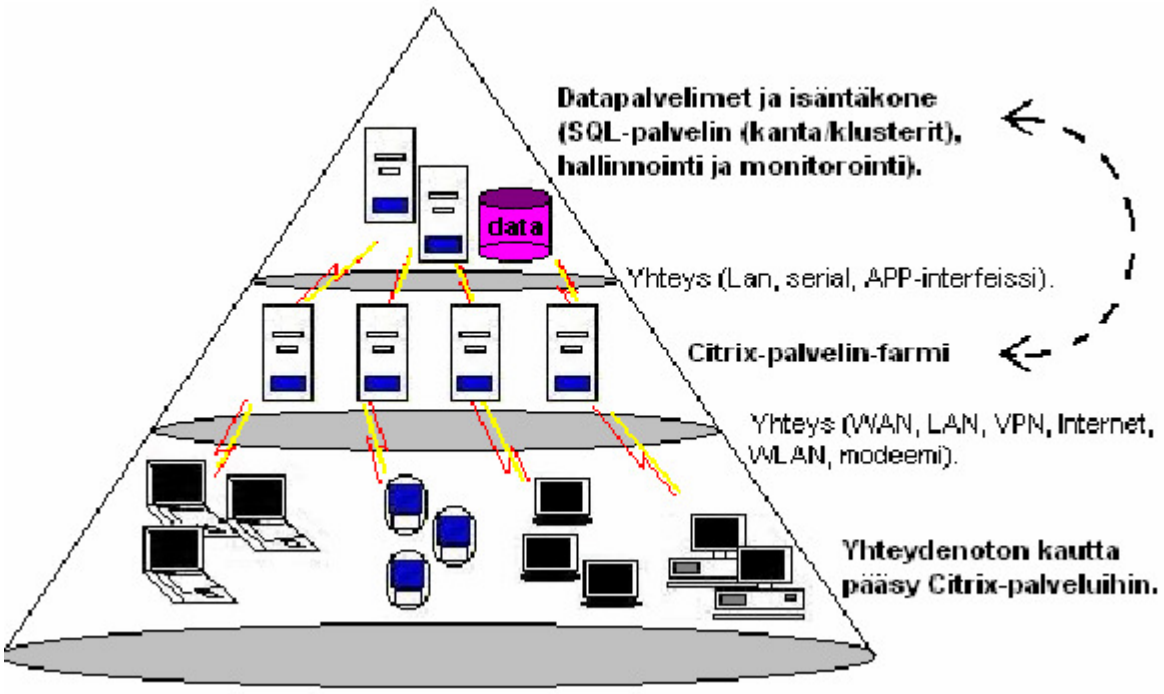
4. Oikeuksien suojaaminen: Ennen kuin oikeudet lähetetään käyttäjälle, alttiit osat (esim. CEK) suojataan REK:llä (Rights Encryption Key). Oikeudet sidotaan salatusti käyttäjän koneen DRM-agenttiin (Myös REK sidotaan käyttäjän DRM-agenttiin siirron ajaksi, näin varmistetaan että vain kyseinen agentti pääsee käsiksi oikeuksiin ja pystyy purkamaan CEK-salauksen.). Lisäksi oikeuksien antaja (Rights owners) allekirjoittaa oikeudet digitaalisella allekirjoituksella.

5. Oikeuksien lähetys käyttäjälle: Oikeuksien suojaamisen jälkeen oikeudet ja DCF-paketti lähetetään käyttäjän DRM agentille, mitä tahansa kuljetustapaa käyttäen (esim. HTTP tai MMS (*Multimedia Messaging Service*) käyttäen) joko yhdessä tai erikseen. DRM-agentin täytyy myös tunnistaa oikeuksien antaja oikeuksien lähetyksen yhteydessä. [42, s. 14 – 16.]

# 8 MOBIILI CITRIX

Citrix on yhdysvaltalainen, lähinnä ohjelmointiin keskittynyt yritys, joka tarjoaa etätyöhön tarkoitettuja tuotteita, jotka tarjoavat suojatun yhteyden käyttäjän käyttämille tai ulkopuolisille sovelluksille. Citrix tarjoaa ratkaisun sovellusten ja käyttäjien väliselle tietoliikenteelle tarjoten pääsynhallintaratkaisun etätyötä varten. Citrix-ympäristö asennetaan olemassa olevan järjestelmän (esim. Windows Server-2003-ympäristön) päälle, jolloin se tuo Citrixin tarjoamat edut ja palvelut järjestelmään. Citrix tarjoaa monille eri käyttäjille samanaikaisen mahdollisuuden ottaa etäyhteys terminaalipalvelimien kautta Windows-terminaali palveluihin (Windows Terminal Service).

Citrix käyttää etäyhteyden muodostamiseen itse kehittämäänsä protokollaa, josta käytetään nimitystä ICA (Independent Computing Architecture), jonka päätehtävänä on välittää työpöytänäkymä etäyhteyskoneelta käyttäjän koneelle ja käyttäjäkoneelta tehdyt operaatiot (operaatioita on esim. hiiren liikuttaminen, näppäimen painallus tai näkymän vaihto) palvelimelle. Citrix-järjestelmää voidaan käyttää lähes kaikissa yleisimmissä käyttöjärjestelmissä (mm. Java, Symbian, Windows ja Unix) ja selaimissa. ICA-käyttöliittymä toimii joko paikallisesti asennettuna (eli koneeseen asennettuna) tai selainpohjaisena. Selainpohjaisessa ympäristössä avataan yrityksen terminaaliyhteyteen varattu verkkosivu, josta saadaan ladattua oikea käyttöliittymä yhteydenottoa varten. Sivut ovat salattuja ja käyttäjä tarvitsee käyttäjätunnuksen ja salasanan lisäksi yrityksessä käytetyn avainlukukortin päästäkseen Citrix-palveluihin. Citrix voidaan yleisesti ottaen käsittää kolmen tason järjestelmänä (kuva 24).




*Kuva 24: Citrix järjestelmän kolme tasoa [44, s.9].*

Alin Citrix-järjestelmän kolmesta tasosta on käyttäjätaso, jossa sijaitsevat päätelaitteet (esim. puhelin tai kannettava tietokone) ja niille tehdyt sovellukset. Toisella tasolla sijaitsee Citrix-palvelin-farmi eli palvelinperhe. Tässä kerroksessa sijaitsevat Citrix-palvelimet palveluineen.

Toisen ja ensimmäisen kerroksen välissä on solmukohta, johon käyttäjät nivoutuvat ottaessaan yhteyden Citrix-palvelimeen. Ylimmällä tasolla sijaitsee isäntäkone, joka hallinnoi ja tarkkailee Citrix-palvelimia sekä tiedonkerääjät (Data Collectors), joiden tehtävänä on pitää yllä lisenssi- ja ohjelmatietoja sekä muita palvelufarmille (toiselle kerrokselle) tärkeitä tietoja. Kolmannessa kerroksessa kerätään myös kaikki tiedot Citrix-ympäristön sisällä tapahtuvasta liikenteestä. Käytännössä tämä kolmen kerroksen malli on kuitenkin häilyvä, sillä isäntäkone ja tiedonkerääjät sijaitsevat yleensä Citrix-farmissa eli toisessa kerroksessa. [44, s. 5, 7 - 10.]

Kuten aikaisemmin mainittiin, Citrixiä voi käyttää myös nykyisten puhelinten kanssa, sillä uudet puhelinlaitteet ovat riittävän tehokkaita pyörittämään ICA-käyttöliittymää ja GPRS-verkko ja sitä nopeammat verkkotekniikat mahdollistavat käyttäjäystävällisen yhteydenoton päätelaitteelta pääteohjelmistopalvelimelle. Citrix on kuitenkin kehittänyt mobiilisovelluksia melko nihkeästi, lähinnä sen takia, ettei Amerikassa mobiili-innostus ole vielä samalla tasolla kuin Euroopassa. Citrix teki vuonna 2001 Symbian 60-alustalle sovitetun laitteiston, muttei sen jälkeen vähään aikaan kehittänyt mobiili sovelluksia. Nyttemmin Citrix on kehittänyt ainakin Nokian 9500 ja 9300 -laitteille oman ICA-käyttöliittymän (Citrix Presentation Server Client for Series 80, Nokia 9500/Nokia 9300). Lisäksi ainakin Nokian E61, E71 ja 9120 sekä Sony Ericssonin P800 ja P900 malleille on olemassa ICA-käyttöliittymä (tietoja kyseisten laitteiden ICA-käyttöliittymistä löytyy Citrixen verkkosivuilta http://support.citrix.com/product/ica/epoc/). Kännyköiden synkronisointi Citrix-ympäristöön on melko helppoa. Päätelaitteesta, johon Citrix-palvelut halutaan, asennetaan ICA-käyttöliittymä, jonka voi ladata Citrixen kotisivuilta. Yhteys otetaan puhelimen selaimen kautta käytössä olevan verkkoyhteyden esim UMTS tai GPRS avulla web-pohjaiseen sovellusten jakamisjärjestelmään, joka tunnistaa käyttäjän. Sen jälkeen järjestelmä ottaa yhteyttä XML-pohjaiseen palveluun, joka tuo listan sovelluksista, joita käyttäjä voi käyttää. Kun käyttäjä valitsee selainsivulta jonkin sovelluksen, järjestelmä hoitaa tunnisteiden (token) vaihto-operaation ja generoi pyydetyn ICA-tiedoston, jota selain käyttää käynnistääkseen ICA-käyttöliittymän. Kun Citrix Secure Gateway saa käytettävän Citrix-palvelimen IP-osoitteen, se muodostaa yhteyden Citrix-palvelimelle.

Yhteyden muodostettua Secure Gateway salaa ja purkaa asiakkaan ja palvelimen välistä tietovirtaa. Istunnon aikana käyttöliittymän ja palveluiden välillä liikkuvan tiedon turvaamiseksi Citrix käyttää TLS/SSL-salausta. [44, s. 52 – 55.]

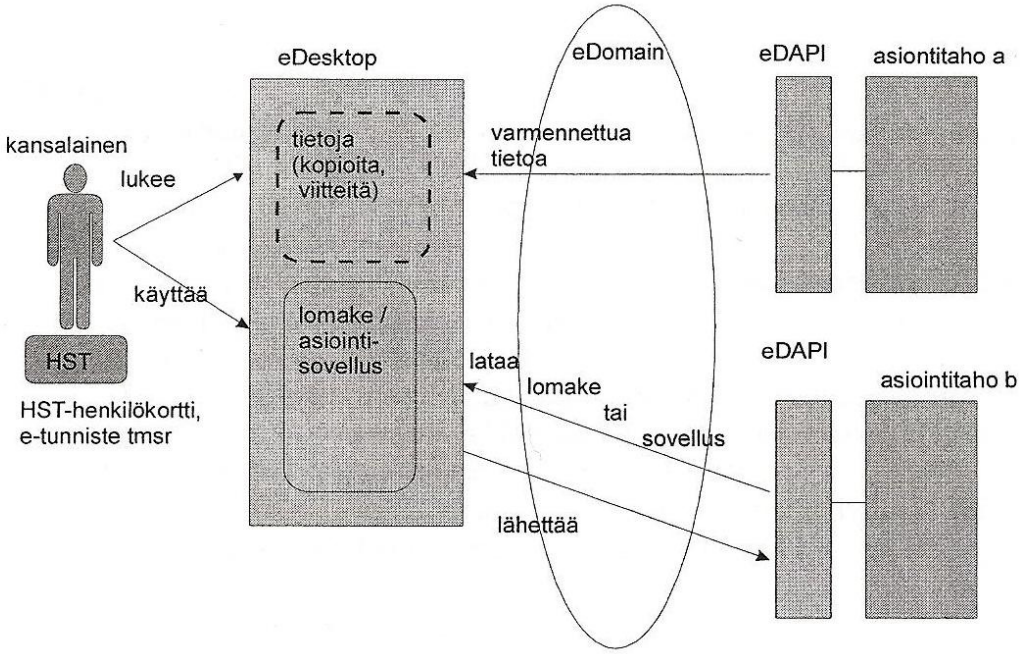
# 9 EDWARDENIN TOIMINTA JA ARKKITEHTUURI

eDWarden on sähköisen asioinnin menetelmä, jossa vahvan tunnistuksen avulla kirjaudutaan tietoturvalliselle alueelle internetissä. Keksinnön tarkoituksena on kehittää tiedon käsittely järjestelmä ja laitteiston toteutus, jolla pystyttäisiin korjaamaan nykyisen sähköisten palvelujen aiheuttamat ongelmat (kts. kappale 2). [45, s. 2.]

## 9.1 eDWardenin arkkitehtuuri

eDWarden arkkitehtuurin osat ovat eDesktop, eDomain ja eDAPI.

- eDesktop-ohjelmisto ladataan verkosta asiakkaan käyttöön kirjautumisen jälkeen.
- eDAPI (*eDesktop Application programming interface*) eli eDesktop-rajapinnan avulla palveluntarjoaja voi julkaista palveluja eDesktop ympäristöön ja hyödyntää eDesktop-ympäristön tarjoamaa toiminnallisuutta.
- eDomain tarkoittaa eDesktop-palvelinten ja palveluntarjoajien verkkoa ja eDWardenin asiakkaiden muodostamaa eDesktop-osoiteavaruutta. [1.]




*Kuva 25: eDWarden-arkkitehtuuri [1].*

eDWarden-arkkitehtuuri tuo työpöydän kiinteäksi osaksi sähköistä palvelua. Työpöytä on tässä tapauksessa laajempi käsite kuin esim. Windows-työpöytä. Sähköisen työpöytä (eDesktop) koostuu ohjelmallisista komponenteista, jotka ladataan verkosta käyttäjälle sähköisen tunnistamisen jälkeen, eli palveluja voi käyttää millä tahansa koneella ei pelkästään koti tai työkoneella. Työpöydän ominaisuuksia on mm. seuraavat asiat:

- Kirjautuminen työpöydälle tapahtuu vahvan sähköisen tai biometrisen tunnistuksen avulla.
- eDesktop työpöytä on saatavilla missä tahansa verkossa eri päätelaitteilla.
- Salaus- ja suojaustekniikat ovat yhdenmukaisia ja kattavia.
- Kansalainen pystyy valvomaan viranomaisten ja yritysten hänestä keskenään vaihtamaa materiaalia.
- Työpöydän avulla voi luoda, muokata, allekirjoittaa ja tallentaa omia asiakirjoja. (Asiakirjoja voidaan tallentaa verkkoon. Tämä säästää tilaa käyttäjän laitteesta ja tiedot ovat paremmassa turvassa verkossa, kuin esim. koti tai virastonkoneella.)

- Asiointipalveluita voidaan käyttää eDesktopin välityksellä.
- eDesktopin avulla on mahdollista tehdä kansalaisten keskinäinen todennettava asiointi.
- Toiselta oikeushenkilöltä saatavien tietoja ja asiakirjoja voidaan vastaanottaa ja tallentaa luotettavasti.
- eDesktopin avulla voidaan viestiä kahden todennetun osapuolen välillä. (esim. sähköposti ja IP-puhelut).
- Käyttäjä myös tallentaa tietonsa verkkoon eikä omalle koneelleen.

Mikäli tulevaisuudessa halutaan että sähköinen asiointi korvaisi kokonaan perinteisen asioinnin, eDesktop tai jonkin muun samoja ongelmia ratkaisevan menetelmän on pakko tulla käyttöön.

eDesktopin ohjelmisto osiin kuuluu ohjelmistot jotka toteuttavat yhden tai useamman seuraavista ominaisuuksista:

- sähköinen tunnistus
- digitaalisten varmenteiden hallinta
- omien tiedostojen ja asiakirjojen luonti, hallinta, muokkaus ja tallennus
- tiedostojen allekirjoitus ja salaus
- suojattujen tiedostojen lähetys ja vastaanotto
- asiointisovellusten suorittaminen
- IP-puhe
- komponenttipohjaisuus eli sovellusten ja palvelujen ajon aikainen käyttö ja koostaminen. [1.]

eDAPI eli e-desktop API (internet työpöydän rajapinta) toimii rajapintana, joka tarjoaa yhteyden e-desktopin ja viranomaisten palvelimien tai e-desktopin ja muiden käyttäjien tai palveluntarjoajien välillä.

API-rajapinnan tarkoituksena on se, että palveluntarjoaja toteuttaa API:n ja että eDWardenin asiakas voi käyttää palveluntarjoajan palveluita kertakirjautumisen jälkeen ilman eri sopimusta tai erillistä tunnistusta. E-desktop API voidaan kuvata mm. seuraavien määritelmien avulla:

- Network service objektille (ORB, *Object Request Broker*) täytyy tarjota rajapinta.
- Tarjottujen dokumenttien täytyy noudattaa joko DTD tai XML-järjestelmän määrityksiä.
- SOAP-viestejä käytetään viestien lähettämiseen.
- ORB-sovitettu palvelupyyntö noudattaa sovittua IDL (*Interface Definition Language*) kuvausta.
- Yhteydet on salattu (esim. SSL:n avulla).
- XML-dokumentit on salattu ja sertifioitu.
- Rajapinnan täytyy kyetä keskustelemaan muiden palveluntarjoajien rajapintojen kanssa. Keskustelu voidaan toteuttaa salatuilla SOAP-viesteillä. [45, s.15 - 16]

E-desktop-rajapinta tarjoaa myös kaavakkeiden ja lomakkeiden käyttömahdollisuuden, joka voidaan toteuttaa esim. XML-teknologian ja Java-komponenttien avulla. E-desktop rajapinnan ja e-desktop palvelun välinen keskustelu voidaan toteuttaa, joko SOAP-viesteillä tai SSL-salauksella. [45, s. 8.]

## 9.2 eDWardenin toiminta

Käyttäjän kirjautuessa eDWardenin tiedonkäsittely palvelimeen (eDesktop server) hänet tunnistetaan vähintään yhdellä seuraavista tavoista: salasana ja käyttäjätunnus, eID (*electronic personal Identification Card*) kortilla, biometrisellä tunnistuksella, salatulla kirjautumisavaimella, kännykän SIM-kortilla kun kirjaudutaan palveluun GSM -verkon kautta tai USIM-kortilla kun kirjaudutaan sisään UMTS-verkon kautta.

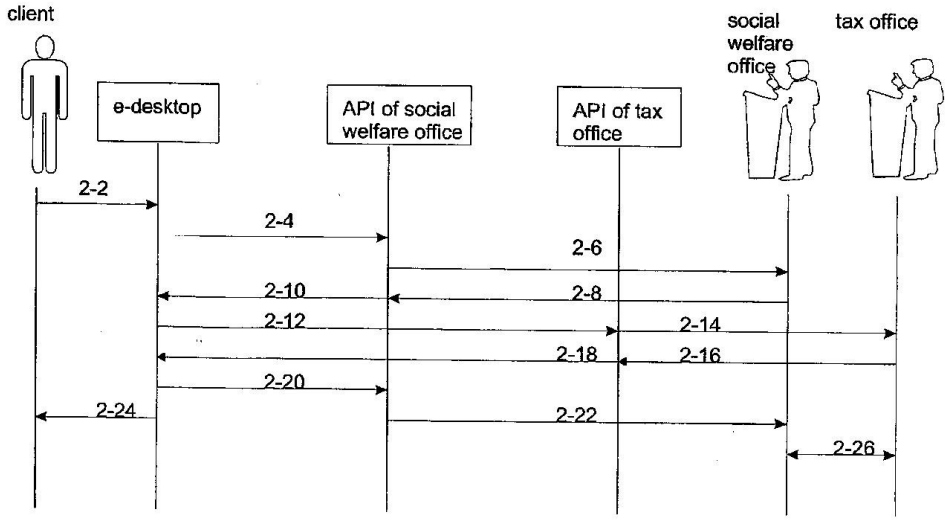
Kännykän kautta kirjautuessa käytetään lähinnä joko SIM tai USIM kirjautumista. eDWarden voidaan toteuttaa langattoman esim. GSM- tai UMTS-verkon tai langallisen esim. Internet-verkon serverille, luodaan käyttäjälle sähköinen asiointiympäristö(eDesktop) vahvan tunnistuksen jälkeen. Tähän asiointiympäristöön toteutetaan yksityisyyden suojaus yhtä tai useampaa ohjelmistotyökalua käyttäen. eDWardenin ohjelmat voidaan jakaa seuraavasti käyttötarkoituksen mukaan.

- Ohjelmat, jotka tarjoavat käyttäjärajapinnan käyttäjälle.
- Ohjelmat, jotka suorittavat elektronisen tai biometrisen tunnistuksen.
- Ohjelmat, jotka tarjoavat toimintalogiikan sähköisiin tapahtumiin liittyvien tietojen lähettämiseen, vastaanottamiseen, käsittelyyn ja tallentamiseen.
- Ohjelmat, jotka suorittavat sähköisten tapahtumien ja allekirjoitusten salaaminen ja purkaminen, sekä muiden varmenteiden käyttö ja laitteiden välisen keskustelun salaus.
- Ohjelmat, jotka toteuttavat keskusteluyhteyden tiedonkäsittely järjestelmiin, välittävät sähköisiä tapahtumia tiedonkäsittelyjärjestelmiin, tarjoavat tiedon tallennusmahdollisuuden, muiden henkilöiden sähköisiin toimintaympäristöihin sekä tiedonkäsittelyjärjestelmiin, tarjoavat mm. sähköisten toimintaympäristöjen hallintaa sekä laskutusta.

Sähköisen käyttöympäristön avulla käyttäjä voi mm. koota ja tallentaa verkkoon itseään koskevaa tietoa, osallistua itseään koskevan tiedon levitykseen ja siirtelyyn verkossa, muodosta yhteyden toisen laillisen henkilön kanssa ja välittää sähköistä tietoa heidän välillään, muodostaa rajapinnan sähköisen maksupalvelun palvelimen ja käyttäjän laitteen välille tai laskutuspalvelun alustan palvelimeen, käyttää maksullisia tietovirta (data stream) palveluja esim. IP-puheluita tai viestintäpalveluja kuten tekstiviesti (SMS, *Short Message Service*) ja MMS- viestipalvelut sekä päästää läpi tai blokata tiettyjä sovelluksia, tai tietyistä IP-osoitteista tulevaa tietoa, käyttäen hyväksi palomuuri teknologiaa.

Dokumentteihin voi myös tarpeen vaatiessa liittää sähköisiä varmenteita, mikäli niitä tarvitaan todentamaan informaation lähde, esim. veroviranomainen voi varmentaa käyttäjää koskevan dokumentin sähköisellä allekirjoituksella ja sähköisen palveluntarjoaja voi veroviranomaisen allekirjoituksesta todentaa dokumentin olevan aito ja muuttumaton.

Kuvassa 26 näkyy esimerkki tiedonkulusta eDWarden ympäristössä, kun käyttäjä täyttää asuntotukihakemusta sosiaalitoimistolle ja jossa asiakas tarvitsee verotietonsa hakemukseensa.




*Kuva 26:Tiedonkulkuesimerkki asuntotuen hakemisessa eDWarden ympäristössä [45, s.41].*

Aluksi (2-2) asiakas ottaa yhteyden e-desktop-palveluun, jossa asiakas tunnistetaan yhden tai useamman tunnistusmetodin avulla (esim. biometrisellä tunnistuksella, eID-kortilla tai SIM-kortilla).

e-desktop-palvelu rakennetaan käyttäjälle ja käyttäjä etsii haluamansa palvelun palveluvalikosta tai e-desktopin hakukoneen avulla. Kohdassa 2-4 e-desktop etsii käyttäjän haluaman osoitteen tai palveluobjektin verkosta (tässä tapauksessa sosiaalitoimiston osoitteen) ja lähettää osoitteeseen palvelupyyntö viestin. kohdassa 2-6 sosiaalitoimiston ohjelmistorajapinta (API) kommunikoi sosiaalitoimiston tiedonkäsittelyjärjestelmän kanssa ja kohdassa 2-8 tiedonkäsittelyjärjestelmä tuottaa pohjatiedot käyttäjärajapintaa varten.

Pohjatiedoilla tarkoitetaan tässä tapauksessa esim. aiemmin varmennettuja tietoja käyttäjästä, kuten aikaisemmat päätökset käyttäjään liittyen. Sosiaalitoimiston API lähettää kohdassa 2-10 kaavakkeen esitäytettyine tietoineen e-desktop ympäristölle, joka tarjoaa käyttäjärajapinnan kaavakkeen täyttöön. Käyttäjä tarvitsee viimevuoden verotustietonsa kaavakkeen täyttämiseen, joten hän ottaa kohdassa 2-12 yhteyden veroviraston tietokanta palveluun (Mikäli hänellä ei jo ole pyydettyjä tietoja). e-desktop etsii verkosta oikean palvelun (eli tässä tapauksessa veroviranomaisen). Veroviraston ohjelmistorajapinnan tarjoama palvelu voi suoraan hakea pyydetyn tiedon veroviraston tietokannoista (kohta 2-14), tai vaihtoehtoisesti käyttää e-desktop ympäristöä tiedon löytämiseen. Kohdassa 2-16 veroviraston tiedonkäsittelyjärjestelmä palauttaa pyydetyn tiedon veroviraston API:lle joka puolestaan siirtää halutun tiedon varmennetussa muodossa e-desktop ympäristöön (kohta 2-18), josta käyttäjä voi tarkistaa sen. Kohdassa 2-20 käyttäjä lähettää e-desktopin avulla täytetyn kaavakkeen, johon on lisätty veroviraston varmentama veroilmoitus tai viittaus kyseiseen veroiloitukseen. Käyttäjä voi myös tarvittaessa varmentaa itse viestin omalla elektronisella allekirjoituksellaan. Kohdassa 2-22 sosiaaliviraston API lähettää saamansa kaavakkeen sosiaalitoimiston tiedonkäsittely järjestelmään. Käyttäjä voi kohdassa 2-24 halutessaan tallentaa täyttämänsä kaavakkeen e-desktopiin, josta hän voi tarvittaessa hakea sen. Kopion voi myös tallentaa virtuaalisesti, eli pelkkänä viitteenä kaavakkeeseen. Kohdassa 2-26 sosiaalitoimisto ja verovirasto keskustelevat tarvittaessa suoraan keskenään, mikäli tietoa vaihdetaan pelkkinä viittauksina. Yhteys veroviranomaisen ja sosiaalitoimiston välille on mahdollista toteuttaa myös e-desktopin kautta. [45, s. 2 – 12.]

### 9.3 Kuinka eDWardeniin päästään mobiiliverkon kautta

eDWardeniin pääsy mobiiliverkon kautta voidaan toteuttaa integroimalla eDWarden Open Mobile Allianssin (OMA) standardeihin, joita käsiteltiin kappaleessa seitsämän. OMA:an integroidessa OWSER-standardi toteuttaa kertakirjautumisen, joka on yksi eDWardenin perusajatus. OWSER-standardissa palvelun- ja tunnisteen tarjoajan täytyy tukea SOAP-viestintää (HTTP:n sidottua SOAP:a), jota myös eDWarden tukee. Tämän takia palveluntarjoajan ja e-desktopin välinen keskustelu toteutettaisiin SOAP-viesteillä.

SOAP:n lisäksi OWSER:n käyttäminen vaatisi SAML:n, XML:n, SSL:n (tai TLS:n) ja PKI:n tukemista, OMA DRM vaatii niin ikään XML ja PKI-tuen toimiakseen. eDWardenin patenttihakemuksessa esitetyissä malleissa on tuki kaikille edellä mainituille protokollille, eli eDWarden voidaan integroida OWSER:n käyttämällä patenttihakemuksessa esitettyä eDWarden arkkitehtuuria. OMA DRM:n käyttäminen vaatisi edellä mainittujen standardien lisäksi myös OMA DRM-agentin. eDWardenista ladattu eDWarden agentti voi toimia myös DRM-agenttina.

# 10 PALVELUNTARJOAJIEN EDWARDENIIN TARJOAMAT MOBIILIPALVELUT

Tässä kappaleessa käydään muutamien esimerkkien avulla läpi palveluita, joita eDWarden mobiilikäyttäjille voi tarjota, sekä sitä miten ne käytännössä toimivat. Yksi tällainen mobiilipalvelu on Nokian Ovi-palvelu.

Nokian Ovi-palvelukokonaisuuteen kuuluu Nokia Music Store, josta käyttäjät voivat hankkia musiikkilevyjä sekä Nokia N-Gage, joka puolestaan tarjoaa käyttäjille erilaisia pelejä. Näiden lisäksi Ovi-palvelukokonaisuuteen kuuluu myös Nokia Maps, joka tarjoaa erilaisia karttoja ja kaupunkioppaita mobiilikäyttäjille. Ovi-palveluja tullaan myös tulevaisuudessa lisäämään. Ovi-palvelu toimii avoimena porttina verkkoyhteisöihin ja tarjoaa näin mahdollisuuden keskittää sisällön, yhteisöt ja kontaktit samaan paikkaan. Ovi-palveluun pääsee suoraan yhteensopivalta Nokia-laitteelta sekä tietokoneelta. Näistä palveluista voi hankkia pelejä, musiikkia ja karttoja esim. OMA:n kappaleessa kahdeksan kuvattujen OWSER:n ja DRM:n avulla. [46.]

Oletetaan, että käyttäjä haluaa ostaa Nokian Music Storesta jonkin levyn. Ensin käyttäjä kirjautuu eDWardeniin esim. SIM-tunnistuksen avulla. Sitten hän etsii Nokian Music Storen joko eDWardenin valikoista tai hakukoneen avulla. eDWarden ottaa Music Storeen (palvelun tarjoajaan) yhteyden SOAP-viestien avulla. Saatuaan yhteyden käyttäjä etsii haluamansa levyn ja ostaa sen. Tämän jälkeen edetään kuvan 18 mukaisessa järjestyksessä, eli palveluntarjoaja (Music Store) pyytää henkilöllisyyden tarjoajalta (tässä tapauksessa e-desktop) käyttäjäntunnistusta ja henkilöllisyyden tarjoaja lähettää SAML-vakuutuksen, jolla palveluntarjoaja tunnistaa käyttäjän.

Tämän jälkeen palveluntarjoaja etsii käyttäjän pankkipalvelun löytämispalvelun avulla. Mikäli käyttäjän pankkipalvelu tarvitsee käyttäjän hyväksynnän maksuun se lähettää SOAP-virheen palveluntarjoajalle, joka uudelleen ohjaa käyttäjän pankkipalvelun sivuille. Käyttäjä hyväksyttää maksun pankkipalvelulle, jolloin palvelu suorittaa maksun ja lähettää palveluntarjoajalle ilmoituksen laskun maksamisesta. Käyttäjä ohjataan takaisin musiikkikauppaan ja hän voi ladata ostamansa tiedoston.

Toinen esimerkki tarjottavasta palveluista ovat mobiilisähköpostipalvelut. Monet sähköpostipalvelut kuten Yahoo webmail ja MSN tarjoavat nykyään sähköpostipalveluita myös mobiililaitteille. Mobiilisähköposti on yleensä yksinkertaistetumpi versio normaaleista sähköpostipalveluista, jotta pienemmällä kännykät pystyisivät suoriutumaan sähköpostin käytöstä. Käytännössä mobiilisähköpostin avulla pystytään tekemään samat asiat kuin normaalillakin sähköpostilla. Otetaan esimerkiksi kännykällä otetun kuvan/viestin lähettäminen sähköpostissa. Käyttäjä ottaa ensin yhteyden e-desktop palveluun, jonka kautta hän ottaa yhteyttä mobiilisähköpostiin käyttäen kertakirjautumista (eli käyttäjän ei tarvitse kirjautua sähköpostiin käyttäjänimeä ja salasanaa käyttäen, koska e-desktop hoitaa käyttäjän tunnistuksen). Kirjautumisen jälkeen käyttäjä valitsee osoitteen, jonne hän haluaa kuvan/viestin lähettää sekä valitsee haluamansa kuvan puhelimestaan tai kirjoittaa haluamansa viestin ja lähettää kuvan/viestin kohe osoitteeseen.

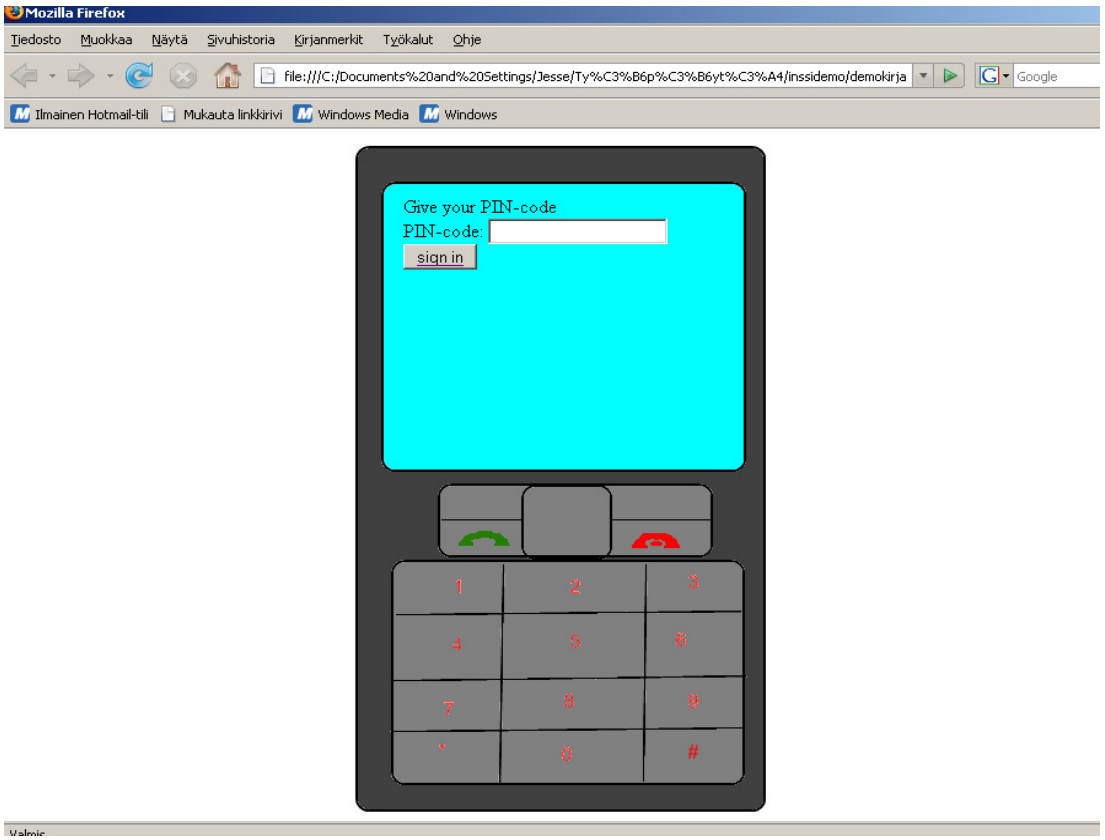
# 11 DEMO

Demon tarkoituksena oli kuvata jokin eDWardenin mobiilipalveluista ja demon aiheeksi valittiin mobiilisähköposti. Demo tehtiin HTML-kielellä. Tarkoituksena oli näyttää, kuinka viestin salauksen ja allekirjoituksen lisääminen ja niiden purkaminen saadusta viestistä toimisi käyttäjän näkökulmasta.

Demo on vain esimerkki siitä, miltä eDWarden voisi näyttää ja miten se voisi toimia. Demossa on oletettu että käyttäjällä on mobiilikansaliasvarmenne (kappale 4.5.4). Tämä oletus on tarpeellinen salatun viestin purkamisessa, jossa käytetään PKI:tä (jota ei normaalilla SIM-kortilla ole). Kuvassa 27 näkyy demon ensimmäinen sivu, jossa kirjaudutaan eDWardeniin.

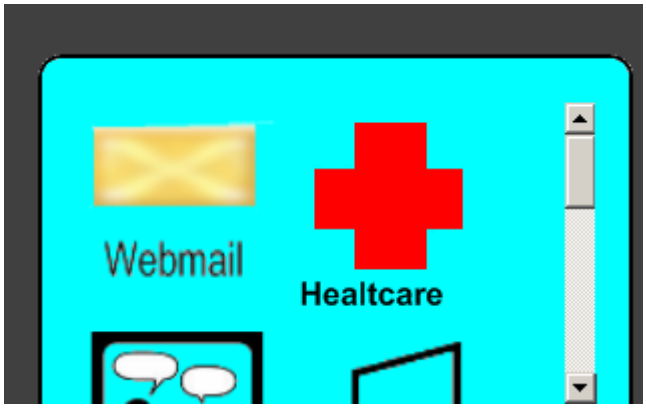
Kännykän kuva on taustakuva ja kaikki tapahtumat on laitettu kännykänkuvan näytön sisään.

Näin demossa saadaan hieman parempi käsitys siitä mitä kännykän käyttäjä oikeasti näkisi kyseistä palvelua käytettäessä. Kuvan kirjautumisessa käyttäjä käyttää puhelimen SIM-korttia kirjautumisessa, jolloin kirjautuminen tapahtuu kappaleessa 4.5.3 esitetyn tavan mukaisesti. SIM-kortilla tapahtuvan tunnistuksen lisäksi demossa kysytään vielä PIN-koodia.



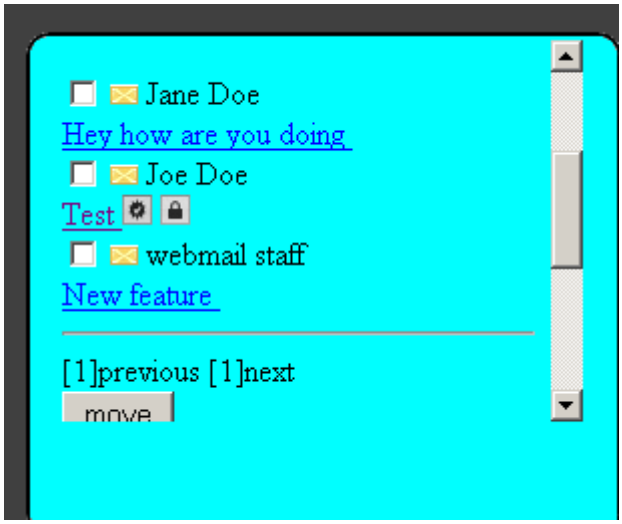

*Kuva 27: Kirjautuminen eDWadeniin*

Kuvassa 28 näkyy esimerkki siitä, miltä eDWarden työpöytä voisi näyttää mobiilikäyttäjän silmissä. Jokainen eDWardenin kautta saatavalla palvelulla on oma kuvakkeensa. Tämän demontapauksessa käyttäjä haluaa tutkia oman sähköpostinsa, joten hän painaa Webmail-kuvaketta, joka vie käyttäjän hänen omaan sähköpostiinsa. Mitään erillistä kirjautumista ei tapahdu, koska käyttäjä on jo kirjautunut eDWardeniin.



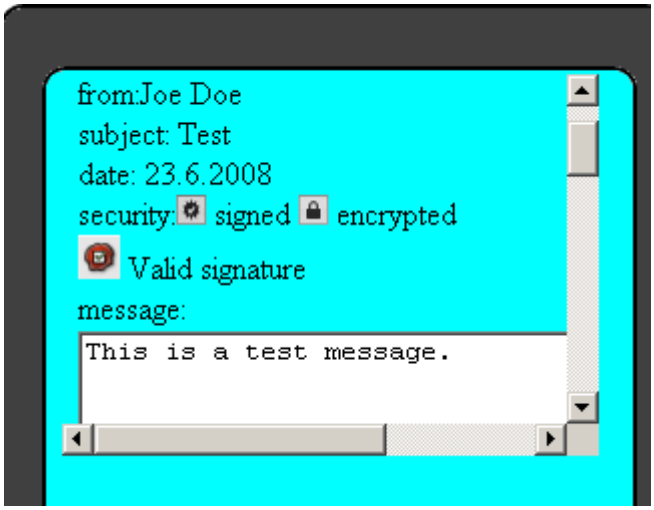

*Kuva 28: eDWarden työpöytä*

Kuvassa 29 näkyy demoon tehty mobiilisähköpostin inbox-sivu. Keskimmäinen kolmesta viestistä on salattu ja allekirjoitettu, minkä huomaa viestin vieressä olevista merkeistä. Kun käyttäjä haluaa avata salatun viestin se tapahtuu viestiä klikkaamalla. Salauksen purkamiseen tarvittava PKI-avain haetaan automaattisesti käyttäjän SIM-kortilta (koska käytetään kertakirjautumista), joten viesti avautuu käyttäjän kannalta katsoen samalla tavoin kuin normaali sähköposti viesti.



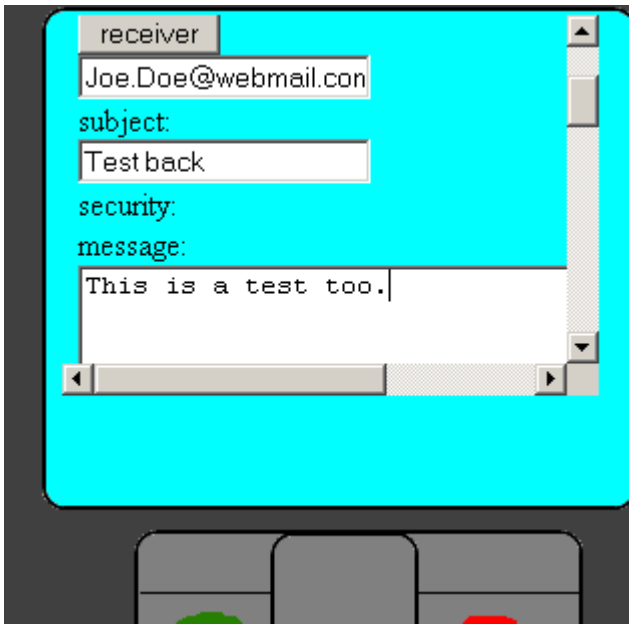

*Kuva 29: Webmail*

Kuvassa 30 näkyy purettu viesti, joka on allekirjoitettu. Allekirjoituksen oikeellisuus näkyy avatussa viestissä.

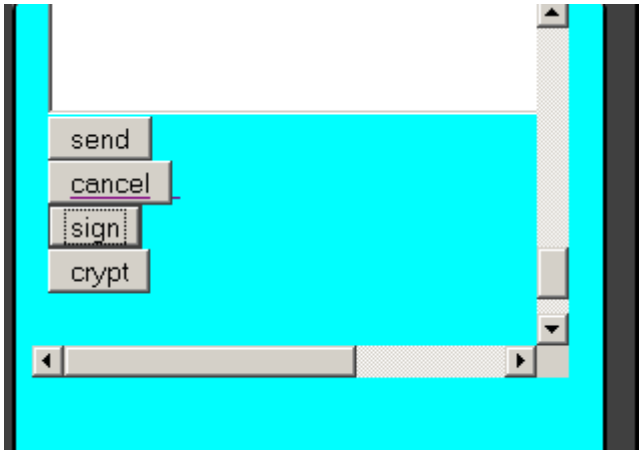



*Kuva 30: Avattu allekirjoitettu viesti*

Seuraavaksi käyttäjä haluaa lähettää viestin takaisin lähettäjälle, joten hän luo uuden viestin, jonka hän sitten salaa ja allekirjoittaa (kuva 31). Tämän jälkeen käyttäjä sulkee webmailin ja voi joko käyttää muita eDWardenin tarjoamia palveluita tai kirjautua ulos eDWardenista.




*Kuva 31: Viesti ilman salausta ja allekirjoitusta*




*Kuva 32: Oman viestin salaus ja allekirjoitus*


*Kuva 33: Salattu ja allekirjoitettu viesti.*

## 12 JOHTOPÄÄTÖKSET

Työn tuloksena voidaan todeta, että eDWardenin patenttihakemuksessa esitetyt toiminteet voidaan toteuttaa yllä esitettyjen protokollien ja standardien avulla. eDWardeniin kirjautuminen vahvan tunnistusmenetelmän avulla, olisi mobiililaitteilla luontevinta hoitaa, käyttäen s. 21 - 23 esiteltyä haaste-vastausmekanismiin perustuvaa EAP-SIM tunnistusta. Lisäksi olisi hyvä kysyä vielä käyttäjän PIN-koodia tai salasanaa tunnistautumisen yhteydessä. Tällä estettäisiin tai ainakin vaikeutettaisiin muitten kuin omistajan pääsyä sähköisiin palveluihin, mikä olisi tarpeellista mm. silloin, kun kännykkä varastetaan (olettaen että varas pystyy käyttämään puhelinta ilman, että hänen tarvitsee kirjoittaa PIN-koodia). Tällainen tilanne voisi olla esim. jos kännykkä on päällä silloin kuin se varastetaan. SIM-tunnistusta käytettäessä täytyisi eDWardenin käyttää teleoperaattorin tarjoamaa tunnistusta.

Kirjautumisen jälkeisen sähköisen käyttöjärjestelmän luomiseen tarvittavat ladattavat objektit voivat sijaita RMIRegisteryssä (s. 14 - 15), jossa voi olla viittauksia etäisiin objekteihin ohjelmistoympäristön sisällä.

eDWardenin patenttihakemuksessa esitellyn API-rajapinnan tarkoituksena on se että palveluntarjoaja toteuttaa API:n (s. 88 e-desktop API:n määritteet) ja että eDWardenin asiakas voi käyttää palveluntarjoajan palveluita kertakirjautumisen jälkeen ilman eri sopimusta tai erillistä tunnistusta. Kaikki eDWarden API:n toteuttavien yritysten palvelut olisivat automaattisesti löydettävissä eDWardenin hakemistopalvelun kautta (katso patenttihakemus s.14 rivit 8 - 16 ja API:n vaatimukset sivulta 15 riviltä 23 - sivulle16 riville 9). Tämän toteuttamiseen HTTP-ympäristössä tarvitaan viittauksessa s.13 esitellyt URL- ja URI-nimiavaruudet, sillä nimet ja hakemistot ovat yksi tärkein tiedon- ja henkilöllisyydenhallinnan osa. Ilman URL- ja URI-nimiavaruuksia monet verkon käytössä itsestään selvänä pidettävät asiat kuten viittaus dokumentista toiseen dokumenttiin ilman, että dokumenttien tekijöiden tarvitsee sopia keskenään käytettävistä ohjelmista tai palvelimista ei toimisi. API:n tukemien identifointi protokollien tulisi pitää huolta siitä, että jokaisella tiedostolla on eDWardenin nimiavaruuessa yksilöllinen nimi. eDWardenilla tulisi olla myös oma hakemistopalvelu (patenttihakemus s.26 rivit 29 -35), jonka käyttö vaatisi tunnistautumista (hakemiston täytyy tunnistautua hakiessaan tietoa muista hakemistoista).

Kuten kappaleessa 9.3 todettiin, eDWardenin mobiili kertakirjautuminen voidaan toteuttaa OMA:n OWSER standardin avulla, joka perustuu Liberty Allianssin määrittelemiin standardeihin (s. 64). Jotta OWSER-standardia voidaan käyttää, täytyy eDWardenin API:n määrittää samat asiat (mm. XML-mallit, SSO, tunnistusyhteys todistukset, tunnistusyhteysluokat ja protokollien ja viestien sitomisen sekä niiden profiilit) kuin s. 64 esitellyt Liberty Allianssin standardit, joihin OWSER perustuu. Osa edellä mainituista asioista on määritelty eDWardenin patenttihakemuksen s. 15 riviltä 23 alkaen, jossa on lueteltu esimerkkejä siitä, mitä API:n tulisi määrittää, hakemuksen s. 2 riveillä 21 - 33 on puolestaan lueteltu eri käyttäjän tunnistus tapoja (tässä työssä samat asiat löytyvät s.88).

eDWarden voi OWSER-standardia käyttäessään käyttää, teleoperaattorin tunnistusta, koska mobiiliverkon operaattorit toimivat OWSER-standardissa useimmiten henkilötunnisteen tarjoajana, joka luo, ylläpitää ja käsittelee henkilöllisyystietoja, sekä tarjoaa tunnistusvakuutuksen kyseiseen tahoon luottaville palveluntarjoajille. Tässä tapauksessa käytettäisiin kappaleessa 6.2 esitetyistä kertakirjautumisluokista True SSO:ta, eli tosikertakirjautumista. Tosikertakirjautumisessa käyttäjä tunnistautuu palveluihin tunnistuspalveluntarjoajan (ASP, OWSER-standardissa yleensä mobiiliverkon operaattori) avulla, jolla on yhteys kaikkiin palveluntarjoajiin, ja joka informoi palveluntarjoajia käyttäjän tunnistuksesta tunnistus julistusten avulla. Proxy-based true SSO tarjoaa myös mahdollisuuden päästä salasana tiedostoon mm. älykortin ja SIM-koodin avulla, kun taas pseudo SSO:n ainoa tunnistustapa on mastersalasana (lisää aiheesta s. 62 - 63).

Sivulla 65 alkava Identity Federation ja SSO-kappaleen alussa kuvataan pyyntö-vastaus-protokolla, jota tässä tapauksessa käytettäisiin (kun teleoperaattori toimii henkilötunnisteen tarjoajana) tunnistusvakuutuksen antamiseen. Mikäli eDwarden toteutetaan niin että teleoperaattori myy tunnistuspalveluja eDWardenille, silloin voidaan liikennöintiprotokollana käyttää kappaleessa 4.2 esiteltyä EAP:ä (EAP ei ole osa IP-tasolla toimivaa eDWardenia ilman sopimusta teleoperaattorin kanssa, koska ne liikennöivät linkkitasolla, eivätkä näin ollen tarvitse IP-osoitteita kuljettaakseen viestejä laitteelta toiselle). Mikäli eDWardenia ei integroida puhelinoperaattorin verkkoon, tapahtuisi sisäänpääsyn kontrollointi sovelluskerroksessa eikä verkkokerroksessa. Tämä siksi että sovelluspalomuurit ovat turvallisempia kuin pakettisuodattimet, koska ne pystyvät valvomaan sekä ohjelmatason käskyjä että piilottamaan käyttäjän ulkopuoliselta verkolta. Tässä tapauksessa täytyy palvelua vastaava proxy-palvelu asentaa sovellussuodattimeen (kts. sovelluspalomuurit s. 46 - 47). OWSER NI Web Service Framework standardissa (kappale 7.3) puolestaan määritellään osat, joita tarvitaan käyttäjän yksityisiä ominaisuuksia käyttävään ja suojaavaan verkkopalveluympäristöön, jollainen eDWardenkin patenttihakemuksen mukaan on (s. 89, enemmän tietoa eDWardenin patenttihakemuksessa s.3 rivit 8 - 18). Esimerkki tämän toiminnasta saadaan kun sijoitetaan eDWarden WSR:n paikalle sivun 69, kuvasta 17 otettuun esimerkkiin.

Yksi eDWardenin patenttihakemuksessa mainittu asia oli se, että asiakkaalla on vain hänen omassa käytössään kaikki häntä itseään koskevat tiedot ja että asiakkaalla on mahdollisuus tallentaa sähköiseen ympäristöönsä häntä koskevia tietoja (patenttihakemus s. 4 rivit 20 -25 ja s.5 rivit 17 -22). Tämä ominaisuus voidaan toteuttaa DRM:n (kappale 4.13) avulla. DRM-arkkitehtuurin pääasiallisista toimijoista (s. 50) lisenssipalvelin olisi eDWardenin frameworkiin kuuluva palvelin ja sisältöpalvelimella olisi eDWarden asiakkaan tiedot. Sivun 50 kuvasta 12 annettu DRM esimerkki voidaan kääntää suoraan eDWarden sovelluksen esimerkiksi kun sana DRM korvataan sanalla eDWarden. DRM-standardeista voitaisiin käyttää kappaleissa 7.5 - 7.7 esiteltyä OMA:n DRM:ä, jossa käyttöoikeudet annetaan REL-kielen määrittelemänä XML-tiedostona. OMA DRM vaatii toimiakseen XML ja PKI tuen, jotka kummatkin löytyvät eDWardenin patenttihakemuksessa esitetyistä esimerkeissä (patenttihakemuksen s. 17 rivit 9 - 11 (PKI) ja s. 15 API:n vaatimukset rivit 29 -32 (XML)). Lisäksi DRM tarvitsee toimiakseen DRM-agentin. DRM-agenttina voi eDWardenissa toimia eDWardenista ladattu eDWarden-agentti (patenttihakemus s.18 rivit 15 -24). eDWardenin patenttihakemuksessa mainittu kansalaisten keskinäinen todennettava asiointi (patenttihakemus s.5 rivit 3 -10) on mahdollista toteuttaa sähköisellä allekirjoituksella ja PKI:lla, joiden avulla pystytään varmistamaan viestin eheys, tehdyn sopimuksen pitävyys sekä tunnistamaan viestinnän toinen osapuoli (enemmän tietoa kappaleissa 4.6 ja 4.7).

eDWardenin patenttianomuksessa esitettyjen toiminteiden avulla voidaan myös kappaleessa 8 läpikäydyn Citrixin toiminta muuttaa kuvaamaan eDWardenin toimintaa, kun vain sanan Citrix tilalle laitetaan sana eDWarden. Poikkeuksena on ICA-protokolla, jonka Citrix on patentoinut ja jota ei näin ollen voida tietenkään käyttää eDWardenissa.

**VIITELUETTELO**

[1] Sähköisen asioinnin työpöytä - eDesktop.

[2] Penttinen, Jyrki, GPRS tekniikka. WSOY, Helsinki 2001.

[3] 4G. Wikipedia [Verkkoartikkeli, viitattu 10.4.2008]. Wikipedia > 4G
Saatavilla: http://fi.wikipedia.org/wiki/4G.

[4] Manner, Jukka, Provision of Quality of Service in IP-based Mobile Access Network. Helsingin yliopisto. Tiedonkäsittely tieteen toimiala. 2003. Saatavilla: https://oa.doria.fi/bitstream/handle/10024/2937/provisio.pdf?sequence=1.

[5] Isoc, Network Design with Mobile IP. [Verkkodokumentti, viitattu 26.4.2008]. Saatavilla: http://www.isoc.org/inet2001/CD_proceedings/T40/inet_T40.htm.

[6] Keski-Kasari, Sami, Verkkopalveluiden autentikointi yhteisen käyttäjätietokannan avulla. Diplomityö. Tampereen teknillinen korkeakoulu. Tietotekniikan osasto. Tampere 2002. Saatavilla: http://www.wirlab.net/pdf/di_tyo_samikk.pdf.

[7] Lehmonen, Harri, 802.1X-autentikoinnin käyttöönotto toimistoverkossa. Insinöörityö. Helsingin Ammattikorkeakoulu. Tietoliikenteen koulutusohjelma. Helsinki 2007. Saatavilla: https://oa.doria.fi/bitstream/handle/10024/5804/stadia-1177576999-9.pdf?sequence=1.

[8] Penttinen Jyrki, 3G ja erityisverkot, WSOY, Helsinki 2006.

[9] Asokan, N. ; Nyberg, Kaisa; Niemi, Valtteri, The Insecurity of Tunnelled Authentication Protocols. [Verkkodokumentti, viitattu 29.4.2008 ] Nokia Recearch Center. 2002. Saatavilla :http://www.saunalahti.fi/~asokan/research/MitM_2002-10-03-henry.ppt#262,17, Analysis of the problem.

[10] Philip J.Windley; Digital Identity. O`Reilly Media Inc. 2005.

[11] Two-factor authentication. Wikipedia [verkkoartikkeli, viitattu 16.2.2008] Wikipedia > Two-factor authentication. http://en.wikipedia.org/wiki/Two-factor_authentication.

[12] Valonen, Mika, Sähköisen kauppapaikan turvallisuus. Pro gradu –tutkielma. Joensuun yliopisto. Tietojenkäsittelytiede. Joensuu 27.11.2005 ftp://cs.joensuu.fi/ pub/Theses/2005_MSc_Valonen_Mika.pdf.

[13] Kauppinen, Kimmo, Älykortit. Mediatekniikan raportti. EVTEK-ammattikorkeakoulu. Mediatekniikka. 19.4.2005. streams.evtek.fi/mts05/raportti/KauppinenKimmo_korjattu.doc.

[14] Liberty Alliance, SIM Strong Authentication to Internet Services. [Verkko dokumentti, viitattu 19.2.2008] Whitepaper. 2006. Saatavilla: www.projectliberty.org/liberty/content/ download/397/2750/file/SIM_Strong_Authentcation_Whitepaper.pdf.

[15] Klemetti, Kristiina, Mobiilivarmenne. [Verkkodokumentti, viitattu 27.4.2008]. FiCom > Tietoa toimialasta > Tekniikkaa suomeksi > Tiedote Saatavilla: http://www.ficom.fi/tietoa/ tietoa_4_1.html?Id=1126527825.html.

[16] Penttonen, Katja, Luottamuksen hallinta avoimissa verkoissa käyttäen julkisen avaimen järjestelmää. Kandidaatin tutkielma. Tietojärjestelmätiede. Jyväskylän yliopisto. Saatavilla: http://www.cs.jyu.fi/~airi/opinnaytteet/Penttonen.pdf.

[17] Kettunen, Eero, Tiedonsalaus [Verkkodokumentti viitattu 22.1.2008]. Lahden ammattikoulu. Liiketieteenlaitos. Lahti 2004. Saatavilla: http://www.lpt.fi/it/opetus/atk-matematiikka/ tiedonsalaus.pdf.

[18] Lehto, Gitta, Sähköiset allekirjoitukset. Sähköisen kaupankäynti seminaari. Helsingin yliopisto. Tietojenkäsittelytieteen laitos. Helsinki. [Viitattu 28.3.2008] .Saatavilla: http://www.helsinki.fi/~gmlehto/ seminaari/sahkoalle.doc.

[19] Jim Brayton, Andrea Finneman, Nathan Turajski, Scott Willey, PKI [verkkodokumentti, viitattu 16.2.2008]. Searchsecurity.techtarget http://searchsecurity.techtarget.com/ sDefinition/ 0,,sid14_gci214299,00.html.

[20] SOAP (protocol). Wikipedia [Verkkoartikkeli, viitattu 21.3.2008] Wikipedia > SOAP. Saatavilla: http://en.wikipedia.org/wiki/SOAP_(protocol).

[21] W3C, SOAP version 1.2 Part 1: Messaging Framework. [Verkkodokumentti, viitattu 21.3.2008] http://www.w3.org/TR/soap12-part1/.

[22] Chan, Henry; Lee, Raymond; Dillon, Tharam ; Chang, Elizabeth, E-Commerce Fundamentals and Application. John Wiley & Sons, Chichester, 2001.

[23] Transport Layer Security. Wikipedia [Verkkoartikkeli, viitattu 10.4.2008] Wikipedia > Transport Layer Security. http://en.wikipedia.org/wiki/Transport_Layer_Security.

[24] XML. Wikipedia [Verkkoartikkeli, viitattu 9.2.2008] Wikipedia > XML. Saatavilla: http://fi.wikipedia.org/wiki/XML.

[25] Security Assertion Markup Language. Wikipedia [Verkkoartikkeli, viitattu 28.4.2008] Wikipedia > SAML. http://en.wikipedia.org/wiki/SAML.

[26] Palomuuri. Wikipedia [Verkkoartikkeli, viitattu 17.3.2008] Wikipedia > Palomuuri. http://fi.wikipedia.org/wiki/Palomuuri.

[27] Oksanen, Ville, Trusted Computer Platform Alliance - Hyvästit yleiskäyttöiselle PC:lle? [Verkkoartikkeli, viitattu 21.9.2008] http://www.iprinfo.com/page.php?page_id=36&action=articleDetails&a_id=162&id=14.

[28] Hassinen, Marko, Studies in Mobile Security. Doctoral dissertion. Kuopion yliopisto. Informaatioteknologia ja kauppatieteet. Kuopio 2007. Saatavilla: http://www.uku.fi/ vaitokset/2007/isbn978-951-781-989-3.pdf.

[29] Raisamo, Roope, Käyttöliittymien ohjelmistoarkkitehtuurit. Tampereen yliopisto. Tietojenkäsittelylaitos. Tampere 2006. Saatavilla: http://www.cs.uta.fi/reports/bsarja/B-2006-1.pdf.

[30] Lindholm Samu, Bluetooth-pohjaisen RFID-lukijan liittäminen valmiiseen kehitysympäristöön. Insinöörityö. EVTEK-ammattikorkeakoulu. Mediatekniikan koulutusohjelma. 5.5.2006. Saatavilla: http://streams.evtek.fi/thesis_seminar_2006/reports/3-InsTyo Lindholm20060505.pdf.

[31] VoIP. Wikipedia [Verkkoartikkeli, viitattu 19.4.2008] Wikipedia > VoIP. Saatavilla: http://fi.wikipedia.org/wiki/VoIP.

[32] Antikainen, Harri; Rusanen, Jarmo, OuKa mobiilivyöhyke. Esisuunnitelma. Oulun yliopisto. Maantieteellinen laitos. 10.5.2005. Saatavilla: http://www.pohjois-pohjanmaa.fi/file.php?1266.

[33] Opengroup, Introduction to Single Sign-On. [Verkkodokumentti, viitattu 18.1.2008] Opengroup > security > sso. Saatavilla: http://www.opengroup.org/security/sso/sso_intro.htm.

[34] Dunne, Chris, Build and implement Single Sign-On solution. IBM. [Verkkodokumentti, viitattu 22.1.2008] IBM > developerworks > web > library > wa-singlesign. Saatavilla: http://www.ibm.com/developerworks/web/library/wa-singlesign/.

[35] Byfuglien, Mats, A mobile single sign-on system. [Verkkodokumentti, viitattu 5.3.2008] Master Thesis. Department of Computer Science and Media Technology. Saatavilla. http://hig100.hig.no/imt/file.php?id=2484 viitattu 5.3.2008.

[36] Wikipedia [Verkkoartikkeli, viitattu 16.2.2008] Wikipedia >Open Mobile Alliance. Saatavilla: http://en.wikipedia.org/wiki/Open_Mobile_Alliance.

[37] Open Mobile Alliance, OMA-OWSER-Network_Identity-Specification-V1_0-20040316-C. [OWSER-standardi] Saatavilla: www.openmobilealliance.com.

[38] Liberty Alliance, Liberty ID-FF Authentication Context Spesification version: 1.3 [Liberty Allianssin standardi] Saatavilla: http://www.projectliberty.org > resource_center > specifications > liberty_alliance_specifications_support_documents_and_utility_schema_files.

[39] Open Mobile Alliance, OMA-TS-OWSER_NI_FF-V1_0-20051220-C. [OWSER- standardi] Saatavilla: www.openmobilealliance.com.

[39] Open Mobile Alliance, OMA-TS-OWSER_NI_WSF-V1_0-20051220-C. [OWSER-standardi] Saatavilla: www.openmobilealliance.com.

[40] Open Mobile Alliance, OMA-AD-OWSER_NI-V1_0-20060328-A. [OWSER-standardi] Saatavilla: www.openmobilealliance.com.

[41] Open Mobile Alliance, OMA-TS-DRM-DRM-V2_0_2-20080118-D [DRM-standardi]. Saatavilla: www.openmobilealliance.com.

[42] Open Mobile Alliance, OMA-AD-DRM-V2_0_1-20080118-D [DRM-standardi]. Saatavilla: www.openmobilealliance.com.

[43] Sinkko, Sami, Citrix-terminaalipalvelujen perusteet ja hyödyt yritykselle esimerkkinä Aurinkomatkat Oy. Insinöörityö. Helsingin ammattikorkeakoulu. Tietoliikennetekniikka. Helsinki 2006. Saatavilla: https://oa.doria.fi/bitstream/handle/10024/5551/stadia_1165238794_7.pdf?sequence=1.

[44] Mäkipää Innovationin Patenttihakemus hakemusnumero PCT/FI2004/000119.

[45] Nokia, Avaa Ovi uuteen aikakauteen. [Verkkoartikkeli, viitattu 28.4.2008]. Nokia > Nokia > lehdistö > tiedotteet > arkisto > Q3 2007. Saatavilla: http://www.nokia.fi/A4351118?newsid=11497