

hyväksymispäivä

arvosana

arvostelija

## Julkisen hallinnon sähköiset yhteispalvelut

### **Pro gradu - tutkielma**

Topi Laamanen

Helsinki 6.11.2004

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET

Tiedekunta/Osasto <b>Matemaattis-luonnontieteellinen</b>		Laitos – Institution <b>Tietojenkäsittelytieteen laitos</b>	
Tekijä – Författare <b>Topi Laamanen</b>			
Työn nimi – Arbets titel <b>Julkisen hallinnon sähköiset yhteispalvelut</b>			
Oppiaine – Läroämne <b>Tietojenkäsittelytiede</b>			
Työn laji – Arbets art <b>Pro gradu –tutkielma</b>		Aika – Datum <b>6.11.2004</b>	Sivumäärä – Sidoantal <b>73</b>
Tivistelmä – Referat  <p>Erilaisten palvelujen tarjoaminen verkossa on yleistynyt internetin leviämisen myötä. Myös julkiseen hallintoon kohdistuu paineita tarjota palveluita verkossa. Tähän mennessä julkisen hallinnon tarjoamat verkkopalvelut ovat olleet lähinnä tietoa tarjoavia, tai yhteen virastoon kytkeytyviä palveluita. Tavoitteena on, että kansalaiset voisivat asioida yhden portaalin kautta ja monimutkaistenkin transaktioiden teko (esimerkiksi veroilmoitus) onnistuu luontevasti.</p> <p>Tietojenkäsittelytieteen näkökulmasta tällaisten yhteispalveluiden (one-stop government) toteuttaminen vaatii kehittyneitä sovellusintegraatiotekniikoita. Transaktioiden käsittely vaatii myös työkulkujen hallintaa ja virtuaalisten organisaatioiden muodostamista. Tietoturva ja tietosuoja ovat myös oleellisia asioita sähköisessä asioinnissa.</p> <p>Tieteellisiä artikkeleita aihe-alueelta löytyy paljon, mutta konkreettisia teknisiä ratkaisuja on vain muutama. Tämä tutkielma on state-of-the-art tyyppinen katsaus näihin tekniikoihin ja pilottihankkeisiin. Laajempaan tarkasteluun on otettu neljä projektia: ADSS-OSCAR, PASSPORT, eGOV ja WebDG.</p> <p>Tutkimusongelmana on: onko yhteispalveluiden toteutuksessa muodostunut jo vakiintuneita standardiratkaisuja, ja mitä ongelmia on vielä ratkaisematta? Jo muodostuneiksi standardeiksi todeetaan XML, ja siihen perustuvat sovellukset, erityisesti web-palvelut (SOAP, WSDL, UDDI). Myös J2EE-standardi on vahvasti esillä. Alueita, jotka vaativat suurempaa huomiota tai joihin liittyy ratkaisemattomia ongelmia, ovat asiakaslähtöisyys, läpinäkyvyys, käytettävyys, työkulkujen hallinta, tietosuoja ja suorituskyky. Tekniset välineet palveluiden rakentamiseen ovat kuitenkin olemassa ja suurimmat ongelmat ovat ei-tekniisiä.</p> <p>ACM Computing Classification System (CCS):  H.5.3 [Group and Organization Interfaces]  J.1 [Administrative Data Processing]</p>			
Avainsanat – Nyckelord <b>julkinen hallinto, yhteispalvelut, sovellusintegraatio, portaalit, one-stop government, XML</b>			
Säilytyspaikka – Förvaringställe <b>Kumpulan tiedekirjasto, sarjanumero C-2004-</b>			
Muita tietoja – Övriga uppgifter			

## Sisältö

1	Johdanto.....	1
2	Yhteispalvelut – ”one-stop government” .....	3
2.1	Keskeiset käsitteet ja määritelmät .....	3
2.2	Tarve yhteispalveluille .....	5
2.3	Julkisen hallinnon erityispiirteitä .....	7
2.4	Vaatimuksia yhteispalveluille .....	9
2.5	Viitemalli .....	13
3	Sovellusintegraatio.....	16
3.1	Portaalit.....	17
3.2	Integraatioteknologiat.....	19
3.3	Integraatiostandardeja.....	23
3.3.1	XML .....	23
3.3.2	Web-palvelut .....	24
3.3.3	RosettaNet.....	26
3.4	Työnkulkujen hallinta.....	27
3.4.1	Transaktiot .....	27
3.4.2	Virtuaaliset organisaatiot .....	28
3.4.3	Työnkulkuihin liittyviä projekteja.....	30
4	Kohti sähköisiä yhteispalveluita.....	40
4.1	Hankkeissa käytettyjä teknisiä ratkaisuja .....	40
4.1.1	ADSS-OSCAR .....	40
4.1.2	PASSPORT/CB-BUSINESS .....	42
4.1.3	eGOV .....	45
4.1.4	WebDG .....	48
4.2	Sähköisen hallinnon hankkeista saatuja kokemuksia.....	52
4.3	Tietoturva ja tietosuoja .....	55
5	Johtopäätökset.....	60

# 1 Johdanto

Internetin yleistymisen myötä erilaiset palveluntarjoajat ovat siirtäneet palveluitaan verkkoon. Esimerkiksi Suomessa ihmiset ovat alkaneet tottua siihen, että suurin osa palveluista on mahdollista hoitaa myös internetin kautta. Sähköisten palveluiden edelläkävijöitä ovat olleet kaupalliset yritykset, kuten kirjakaupat. Myös julkiseen hallintoon on alkanut kohdistua paineita sähköisten palveluiden käyttöönoton suhteen, ja muutenkin viimeisen viidentoista vuoden aikana vaatimukset julkisen hallinnon palveluiden parantamiseen ovat kasvaneet.

Teollistuneissa länsimaissa, kuten esimerkiksi pohjoismaissa ja Saksassa, julkisen hallinnon vastuulle on kertynyt lukuisia palveluita. Pohjoismaisessa hyvinvointiyhteiskunnan mallissa julkinen hallinto pyrkii huolehtimaan monista kansalaisten perustarpeista. Julkisella sektorilla toimintatavat ovat olleet erilaiset kuin yksityisellä sektorilla. Yksityisellä sektorilla rahaa on tultava enemmän kuin sitä menee, ja täten on pyrittävä tehokkaaseen toimintaan. Ja kun maksaja on asiakas, pyritään yleensä mahdollisimman asiakaslähtöiseen toimintaan. Julkisella sektorilla asiakas, eli kansalainen, ei ole maksaja. Maksajana toimii valtio. Toki viime kädessä maksajana on kansalainen verojen muodossa, mutta suoraa vaihtosuhdetta (rahat palvelua tai tuotetta vastaan) ei yksityisestä sektorista poiketen ole. Muun muassa tämä on johtanut julkisella sektorilla tehottomaan toimintaan, joka monesti on kaikkea muuta kuin asiakaslähtöistä. Parantamista ei ole ollut pelkästään asiakasrajapinnassa, vaan myös prosesseissa ja erityisesti eri viranomaisten välisessä yhteistyössä. Useimmille on tuttua luukulta toiselle juoksuttaminen.

Sähköiset palvelut tarjoavat julkiselle hallinnolle mahdollisuuden parantaa palveluita. Palveluiden siirtäminen verkkoon on jo osittain alkanut, mutta oikein toteutettuna voidaan myös korjata byrokraattisuuden ongelmia. Uudet teknologiat mahdollistavat oikein käytettynä asiakaslähtöisemmän toiminnan.

Viimeisen viiden vuoden aikana onkin ryhdytty keskustelemaan hankkeista joilla julkisen hallinnon palveluita voitaisiin parantaa uuden teknologian avulla. Keskustelua on käyty julkisen hallinnon piirissä ja myös akateemisissa julkaisuissa. Käytän tässä tutkielmassa termiä ”yhteispalvelut”, joka tarkoittaa kansalaisen mahdollisuutta saada kaikki viranomais- ja muut palvelut keskitetysti ja asiakaslähtöisesti yhdestä paikasta yhdellä kontak-

tilla. Englanninkielisessä kirjallisuudessa käytetään muun muassa termejä ”integrated service delivery” ja ”one-stop government”. Ensimmäinen viittaa asian tekniseen puoleen ja jälkimmäinen taas asiakasrajapintaan. Tässä tutkielmassa painopiste tulee olemaan ensin mainitulla, teknisellä puolella. Suomessa näitä asioita pohditaan aktiivisesti valtionvarainministeriön ja sisäasiainministeriön piirissä. Yhdysvalloissa aihe on aktiivinen kehityksen kohde ja EU:n sisällä on ollut yhteisprojektejakin, kuten eGOV.

Tutkielmassa selvitetään sähköisten yhteispalveluiden teknisiä toteutusmahdollisuuksia. Myös julkishallinnon erityispiirteitä käydään läpi, jotta toteutuksen vaatimukset selviävät. Tutkielman tyyppi on vertaileva kirjallisuusanalyysi, ja sitä voisi kuvata myös state-of-the-art tyyppiseksi katsaukseksi aihealueelta. Aihealueen tietotekninen ydin on sovellusintegraatio, ja sovellusintegraatiotyypeistä portaalisuuntautunut. Tässä tapauksessa sovellusintegraatiossa on erityisesti kiinnitettävä huomiota siihen, että järjestelmän monimutkaisuus piilotetaan kansalaiselta ja asioiden hoito onnistuu vaivattomasti.

Tutkielmassa tarkastellaan ensin yhteispalvelujen käsitettä, tarvetta ja vaatimuksia, joihin pohjautuen esitellään viitemalli. Tämän jälkeen käsitellään toteutusten teknistä pohjaa, sovellusintegraatiota, työnkulkujen hallintaa ja virtuaalisia organisaatioita, joka on abstraktimpi käsite. Tämän jälkeen esitellään tehtyjä pilottihankkeita ja luokitellaan niitä viitemallin perusteella. Johtopäätöksissä arvioidaan, kuinka hyvin esitetyt pilottihankkeet toteuttavat yhteispalveluille asetetut vaatimukset ja sijoitetaan ne viitemalliin. Hankkeiden tekniset puutteet tuodaan esille ja arvioidaan, miksi tiettyjä osa-alueita on painotettu. Tutkielman konkreettinen tavoite ja tutkimusongelma on siis selvittää, onko yhteispalveluiden toteutuksessa muodostunut jo vakiintuneita standardiratkaisuja, ja mitä ongelmia on vielä ratkaisematta.

## 2 Yhteispalvelut – ”one-stop government”

Tässä luvussa käydään läpi yhteispalveluiden käsitettä ja pohditaan, mihin yhteispalveluita tarvitaan. Yhteispalvelut käsitteenä ei ole yksiselitteinen, koska sitä käytetään monissa eri yhteyksissä. Tarkempi termi tässä yhteydessä onkin ”one-stop government”, joka rajaa käsitteen tarkemmin. ”Yksi pysähdys” ja ”julkinen hallinto” ovat siis olennaisia termejä. ”One-stop government” ei kuitenkaan käänny luontevasti suomeksi, ja pyrin käyttämään suomenkielisiä termejä, jos mahdollista. Niinpä tämän käsitteen kohdalla olen päättänyt käyttämään termiä ”yhteispalvelut”.

Lisäksi luvussa käsitellään julkisen hallinnon erityispiirteitä, jotka myös vaikuttavat palveluiden vaatimukseen. Näillä erityispiirteillä erotellaan B2B, ja muut yksityisen sektorin vastaavat palvelut ja sovellusintegraatoratkaisut julkisen hallinnon vastaavista. Vaatimusten perusteella esitellään viitemalli, joka on esitetty kuvissa 1 ja 2.

### 2.1 Keskeiset käsitteet ja määritelmät

Yhteispalvelut, eli ”one-stop government”, on melko uusi termi, ja selkeää vakiintunutta suomenkielistä termiä ei vielä ole. Englanninkielisen termin määritelmä on kuitenkin melko yksiselitteinen. Termi viittaa julkisen hallinnon palveluiden integrointiin. Yksi pysähdys, johon ”one-stop” viittaa, tarkoittaa että kansalainen voi hoitaa kaikki asiansa yhdellä kontaktilla yhdestä rajapinnasta. Itse rajapintaa käsitteeseen ei ole kiinnitetty, vaikka useimmiten puhutaan internetin kautta toimivasta palvelusta. Yhteispalvelukanava voi kuitenkin olla myös henkilökohtaista palvelua, tai jokin muu kanava. Tällöinkin palvelun toteutus kuitenkin perustuu tietotekniikan käyttöön. Käsitteet ”single-window”, ”integrated service delivery” ja ”seamless government” viittaavat samaan asiaan [Hak00, s 7-8]. Kuitenkin esimerkiksi käsitteellä ”integrated service delivery” tarkoitetaan aihealueen teknisempää puolta. Artikkeleissa kyseisiä käsitteitä käytetään sekaisin, eli vakiintuneita nimeämiskäytäntöjä ei ole vielä muodostunut.

Tutkielman pääpaino on G2C-yhteispalveluissa (Government to Citizen), eli julkisen hallinnon ja kansalaisen välisissä palveluissa. Muita palvelutyyppisiä ovat esimerkiksi G2B (Business), G2CB, G2O (Organization) ja G2G [MaM03]. G2G-palveluiden toteuttaminen tosin usein vaatii myös edellä mainittuja muita palvelumuotoja.

Julkisen hallinnon tarjoamat hyödykkeet voidaan jakaa kahteen osaan, tietoon ja palveluihin [PaM04]. Tästä jaottelusta seuraa myös yhteispalveluiden luokittelu. Yhteispalvelut voidaan luokitella kolmeen luokkaan, (1) *ensimmäiseen pysähdykseen* (first-stop), (2) *palvelumyymälään* (convenience store) ja (3) *todelliseen yhteispalveluun* (true one-stop) [Hak00, s. 8-9]. *Ensimmäisen pysähdyksen* pisteestä saa lähinnä tietoa, jolloin tarvitaan vähintään toinen pysähdys itse asian hoitamiseen. *Palvelumyymälässä* on saatavilla useita eri palveluita. Palvelut saattavat kuitenkin olla paikallisia, ja monimutkaisempia palveluita ei ole saatavilla. *Todellisessa yhteispalvelussa* asiakas voi hoitaa kaikki asiansa julkiseen hallintoon yhdellä kontaktilla. Palvelut on integroitu niin, että esimerkiksi eri virastojen toimivaltojen rajat eivät vaikuta palveluun. Tavoite on todellisessa yhteispalvelussa, ja sitä käsitteellä pohjimmiltaan tarkoitetaan. Esimerkiksi Kanadan Government On-Line hanke noudattelee samantyyppistä jakoa [MaM03]. Myös West [Wes04] käyttää samantyyppistä luokittelua, joka kattaa hieman laajemman skaalan kuin edellä mainittu. Hänen luokkansa ovat (1) *ilmoitustaulu* (billboard), (2) *osittainen palvelu* (partial-service-delivery), (3) *portaali* (portal) ja (4) *vuorovaikutteinen demokratia* (interactive democracy). Vuorovaikutteisella demokratialla West tarkoittaa mahdollisuutta palveluiden räätälöintiin mieltymysten mukaan ja kansalaisten helpompaa osallistumista julkiseen keskusteluun ja päätöksentekoon. Luokkia voidaan kutsua myös kypsyystasoiksi [PHA01], joka onkin kuvaavampi termi. Myös muita luokitteluja on olemassa [Hak00, s 9-10], mutta tämän työn kannalta ne eivät ole mielestäni käyttökelpoisia.

Portaalilla tarkoitetaan web-sivustoa, johon on koottu erilaisia palveluita. Termin määritelmä ei ole kuitenkaan kovin selvä. Ensimmäiset portaalit 90-luvun puolivälissä olivat lähinnä hakukoneita [GaG02]. Portaaliksi saatetaan kutsua myös pelkkää linkkikokoelmaa, joka on siis yhteispalveluiden alimmalla kypsyystasolla. Tällainen on esimerkiksi Suomen julkishallinnon verkkopalveluiden yhteinen osoite [www.suomi.fi](http://www.suomi.fi). Parhaimmillaan portaalit kuitenkin toimii porttina ja käyttöliittymänä integroituun palveluun. Ihannetapauksessa kansalainen voi hoitaa kaikki asiansa portaalin kautta, ja oman näkemykseni mukaan portaalit tulevat olemaan tulevaisuudessa tärkein yhteispalvelukanava. Portaalien teknistä puolta käsitellään luvussa 3.1.

## 2.2 Tarve yhteispalveluille

Koska julkiselle hallinnolle on kertynyt paljon erilaisia tehtäviä, hallinto on usein jakautunut toimialoittain ja alueellisesti hajanaiseksi kokonaisuudeksi. Tämän seurauksena julkisen hallinnon palvelut eivät aina ole kovin asiakaslähtöisiä [HaK00, s. 1]. Byrokraattinen, Weberiläinen [Ho02] toimintatapa on johtanut siihen että kukin yksittäinen työntekijä tai yksikkö hoitaa vain ”oman tonttinsa” [HaK00, s. 1], joka johtaa helposti asiakkaiden juoksettamiseen luukulta toiselle. Toisaalta hajanaisuudelle saattaa olla hyvätkin perustelut, jotka voivat liittyä esimerkiksi kansalaisoikeuksiin tai puolueettomuuteen [TrW01]. Usein kuitenkin hallinto itsekin kärsii hajanaisuudesta, ja se näkyy tehottomana toimintana ja kuiluna kansalaisten ja hallinnon välissä [Kra02].

Ensimmäiset askeleet kohti yhteispalveluita otettiin jo vuonna 1961, kun yhdysvaltalaiset tilastotieteilijät kehittivät idean kolmesta isosta tietokannasta, joihin oli tarkoitus kerätä suuri määrä tietoa. Paikalliset julkisen hallinnon yksiköt olisivat sitten saaneet etäyhteyden näihin tietokantoihin. Hanke kaatui kuitenkin vastustukseen lähinnä tietosuojan takia [Len02]. Varsinaisista yhteispalvelupisteistä puhuttiin ensimmäisen kerran 70-luvulla Sveitsissä. Tuolloin hankkeen teknistä puolta ajoi IBM [Len02]. 80-luvulla yhteispalvelun pilottihankkeita oli jo käynnissä Ruotsissa ja Saksassa [Len02], ja muutenkin julkisen hallinnon toimintaa ryhdyttiin tehostamaan tietotekniikan avulla [GaG02, Ho02]. Useimmat yhteispalveluhankkeet kuitenkin kaatuivat vastustukseen ja resurssien puutteeseen [Ho02]. Kehittyvä tietotekniikka ei myöskään muuttanut hallinnon byrokraattisia perusrakenteita, vaikka monet asiantuntijat toivatkin asian aika ajoin esille [LeT02]. Merkittävä muutos tapahtui 1993, kun Yhdysvaltain silloinen varapresidentti Al Gore otti näyttävästi esille julkisen sektorin kehittämisen [LeT02].

Internetin, ja erityisesti world wide webin leviäminen on tuonut mukanaan mullistuksen ja tekniset mahdollisuudet paljon monipuolisemmille sovelluksille. Monet tehdyt projektit ovat kuitenkin olleet tekniskeskeisiä, eikä käyttäjien tarpeita ole juurikaan mietitty. Esimerkiksi Saksassa joidenkin projektien tarkoituksena on saksalaisten tutkijoiden mukaan [LeT02] vain näyttämisen halu teknisessä osaamisessa, kansalaisten tarpeiden sijasta. Muutenkin palveluiden kenttä on vielä hyvin pirstoutunut. Esimerkiksi Yhdysvalloissa on tuhansia erillisiä julkisen hallinnon verkkosivuja [MSB03]. Iso-Britanniassa taas on lähdetty kunnianhimoiseen hankkeeseen, jonka tavoitteena on, että kaikki julkiset palvelut ovat saatavilla sähköisessä muodossa vuonna 2005 [DoM02]. Iso-Britannian hanke tuskin



tulee toteutumaan kyseisessä ajassa, mutta hanke osoittaa rohkeutta kehittää julkisia palveluita. Kansallisia julkisen hallinnon portaaleja on lisäksi muun muassa Itävallassa, Kreikassa, Suomessa, Ranskassa ja Portugalissa. Nämä portaalit ovat kuitenkin lähinnä tietoa antavia ja toiminnallisuutta on vähän [Tam01a].

Monet julkisen hallinnon asiantuntijat [HaK00, s. 1; Ho02] ovat tuoneet esille tarpeen uudentilaiselle mallille, jossa palveluprosessit perustuvat asiakaslähtöisyyteen, eivätkä organisaation rakenteeseen. Asialla eivät ole ainoastaan asiantuntijat vaan myös kansalaiset, jotka ovat jo alkaneet tottua organisaatioiden väliseen yhteistyöhön yksityisellä sektorilla [MaM03]. Yksityisellä sektorilla kokonaisvaltaisempi palvelutapa on usein normaalia toimintaa. Esimerkiksi vanhan ajan kivijalkakaupat, jossa jokainen kauppa myy yhtä tuotetta (liha, maito, leipä), ovat ainakin Suomesta kadonneet. Kaupat tarjoavat koko tuotelaajitelman. Samaan tapaan kansalaisen tulisi saada kaikki julkisen hallinnon palvelut yhdestä pisteestä. Julkisessa hallinnossa ei aina ole kyse vain palveluiden hajanaisuudesta, vaan jopa siitä että ei ole selvää, kenelle tietty asia kuuluu [HaK00, s. 1]. Eräs tutkimus [Ho02] jopa väittää rohkeasti, että sähköiseen hallintoon siirtyminen sinänsä muuttaa jäykkiä toimintatapoja dynaamisemmiksi.

Pyrkimys kokonaisvaltaisuuteen tarkoittaa käytännössä integrointia. Koska perimmäisenä tarkoituksena on asiakaslähtöinen toiminta, pelkkä integrointi ei saa olla ohjenuorana, vaan palveluiden kehittämisessä on tarkasteltava asiakaslähtöisyyttä laajemmin [Hak00, s. 2-3]. Integrointi on vain väline, jolla tavoitteeseen päästään. Saksassa 90-luvun alussa tehdyssä tutkimuksessa [HaK00, s. 2] selvitettiin, mitä kansalaiset pitivät tärkeimpänä asiana julkisissa palveluissa. Tärkeimmäksi asiaksi koettiin ystävälliset asiakaspalvelijat. Kaikkien asioiden hoituminen yhdessä paikassa oli vasta sijalla 11. Toki on muistettava, ettei tuohon aikaan ollut internet-palveluita, eikä niitä myöskään osattu vaatia. Useassa Euroopan maassa 1998 tehdyssä tutkimuksessa [Ric99] taas havaittiin, että kaikkein mieluiten kansalaiset asioisivat internetin kautta juuri julkisen hallinnon kanssa. Ja kolmannen, myös Euroopassa tehdyn tutkimuksen mukaan [Len02], 70% ihmisistä haluaisi hoitaa asiansa niin, ettei tarvitsisi itse tulla paikan päälle esimerkiksi virastoon. Myös Yhdysvalloissa kansalaiset odottavat kasvavassa määrin saavansa julkista palvelua internetin kautta [CCD03]. Paineita internetin laajempaan käyttöön siis on.

Perinteisesti julkisten palveluiden saatavuus on ollut paikasta kiinni. Esimerkiksi Suomen kaltaisessa harvaan asutussa maassa ihmiset ovat eriarvoisessa asemassa. Palvelu, joka olisi saatavilla 24 tuntia vuorokaudessa oman katon alla, parantaisi monien palveluiden saatavuutta. Tähän liittyy kuitenkin muita ongelmia, koska kansalaiselta vaaditaan tietyt perusvalmiudet ja laitteet tekniikan hyväksikäyttämiseksi. Tuo keskustelu rajataan kuitenkin tämän tutkielman ulkopuolelle.

Useissa maissa on jo olemassa jonkinlainen sähköisen hallinnon strategia, joka tähtää entistä tehokkaampaan tietotekniikan hyväksikäyttöön [Pra03]. Pelkästä tietotekniikan hyväksikäytöstä, eli pelkistetystä sähköisestä hallinnosta, on kuitenkin vielä matkaa todellisiin yhteispalveluihin. Esimerkiksi Yhdysvalloissa tehdyssä tutkimuksessa [GaG02] todettiin, että suurin osa julkisen hallinnon portaaleista on kypsyystasoltaan ilmoitustauluvaiheessa.

Yhteispalveluista eivät hyötyisi pelkästään kansalaiset, vaan myös julkisen hallinto itse. Huomion kiinnittäminen asiakkaisiin parantaa julkisen hallinnon imagoa, ja tehokkaamat toimintatavat parantavat julkisen hallinnon toimintaa sisältä päin ja tuovat säästöjä [HaK00, s. 4; Lam03, CCD03]. Julkisen sektorin kasvavat tehokkuusvaatimukset ovatkin kasvattaneet julkisen hallinnon kiinnostusta yhteispalveluita kohtaan [MaM03]. Lisäksi tietotekniikan avulla voidaan parantaa päätöksenteon ja asioiden läpinäkyvyyttä [Pra03], jolloin kansalainen saa helpommin selville tietyn päätöksen perustelut, tai asian käsittelynsä hetkisen vaiheen. Pelkkä tietotekniikan olemassaolo ei tätä tee, vaan järjestelmien on tuettava läpinäkyvyyttä.

Kun yhteispalveluita ryhdytään tekemään, ongelmia ja kustannuksia on odotettavissa paljon. Lisäksi edellä kuvatut hyödyt perustuvat siihen, että kaikki tavoitteet saavutetaan ja projekti onnistuu. Kiinnostus yhteispalveluihin on räjähtänyt muutaman viime vuoden aikana, mutta internetin potentiaali tällä saralla on vielä pääosin hyödyntämättä [GaG02].

### **2.3 Julkisen hallinnon erityispiirteitä**

Julkinen hallinto poikkeaa monelta osin yksityisistä yrityksistä, ja tämän takia julkisen hallinnon yhteispalveluiden rakentaminen on vaativampi tehtävä kuin vastaavien palveluiden rakentaminen yksityisellä sektorilla.

Julkisissa palveluilla eivät päde samanlaiset markkinalait kuin yksityisissä palveluilla. Palveluntarjoajalla on yleensä lain turvaama monopoli, joten vaihtoehtoisia palveluntarjoajia ei ole. Valinnanvapaus siis puuttuu, mutta kansalaisella on oikeus tiettyihin palveluihin, ja tämä asettaa julkisille palveluille tiettyjä velvoitteita [GLL02]. Kansalaiset osaatvatkin nykyisin vaatia hyvää palvelua, varsinkin Suomen kaltaisissa maissa joissa verot ovat korkeat.

Julkisella sektorilla ei välttämättä olla totuttu tekemään yhteistyötä eri yksikköjen välillä. Jos prosesseja integroidaan, työssä vaaditaan eri yksikköjen yhteistyötä. Yhteistyö byrokraattiseen toimintatapaan tottuneiden työntekijöiden välillä saattaa kuitenkin olla takkuista [Wes04, MaM03]. Kyse ei ole vain yhteistyöstä, vaan myös siitä että julkisen hallinnon yksiköt saattavat olla jopa mustasukkaisia tiedoista, jotka ovat aikaisemmin olleet vain heidän saatavillaan, ja jotka nyt pitäisi jakaa [PHA01]. Tiedon eristämisen syynä saattaa kuitenkin olla myös hyvät perusteet, tai jopa lainsäädäntöä, eikä tiedonjakoa pidä levittää kevyin perustein [TrW01, Ho02]. Lainsäädäntö saattaa koskea myös prosesseja [TrW01], mikä hankaloittaa prosessien muokkausta. Muutenkin prosessien uudelleenmuokkaus ja työnkulut saattavat olla julkiselle hallinnolle outoja [Tam01a]. Lisää haastetta tuo se, että julkisen hallinnon prosessit ovat välillä huonosti määriteltyjä [Kre02]. Eräät tutkijat [MaM03] ennustavat yhteispalveluportaaleista jopa taistelukenttää julkisen hallinnon yksikköjen välille. Muutosvastarintaa on aina, mutta on vaikea uskoa, että tällaista ”taistelukenttää” syntyisi laajemmassa mittakaavassa.

Yhteistyötä ei tarvita vain julkisen hallinnon sisällä, vaan myös sen ulkopuolella. Sekä tekninen toteutus, että palveluiden toteuttaminen tuskin onnistuu ilman yksityisen sektorin apua. Julkisella hallinnolla ei ole aina välttämättä kokemusta yhteistyöstä yritysten kanssa [PHA01]. Tämä tuskin pitää täysin paikkaansa, koska ainakin Suomessa esimerkiksi Tietoenatorilla on pitkät perinteet tietoteknisten ratkaisujen toimittamisesta julkiselle hallinnolle.

Rahaa julkisella hallinnolla ei ole käytettävissä yhtä paljon kuin yksityisellä sektorilla. Budjetit ovat tiukkoja [Wes04], joten yhteispalveluiden kaltaisille kehityshankkeille voi olla vaikeaa saada rahoitusta, tai sitten rahoitusta saadaan liian vähän, eikä hanketta pystytä viemään kunnialla loppuun asti. Myös IT-osaaminen on joidenkin tutkimusten [EAT02]

mukaan julkisella sektorilla monesti puutteellista. Toisaalta julkisella hallinnolla on täysin eri intressit kehittää palveluita kuin yksityisellä puolella, koska toiminnan ei välttämättä tarvitse tuottaa voittoa. Täten voidaan päätyä innovatiivisiin teknisiin ratkaisuihin, jotka hylättäisiin yksityisellä puolella liian kalliina [ScE03].

Julkinen päätöksenteko voi olla poukkoilevaa. Jos vaalien jälkeen esimerkiksi Suomen hallituksen kokoonpano muuttuu merkittävästi, politiikkaan saattaa tulla täyskäännös. Päätöksenteon luonne on siis kausittaista. Nykyisin ei tosin ole harvinaista, että suuryritykset tekevät täyskäännöksiä johtajanvaihdoksen seurauksena, joten tämä ei välttämättä ole pelkästään julkisen hallinnon piirre. Päätöksenteko julkisessa hallinnossa on myös usein hidas. Tämä saattaa olla kohtalokasta, jos esimerkiksi lyödään lukkoon teknisiä ratkaisuja, jotka havaitaankin myöhemmin vääriksi ja vanhentuneiksi.

Julkisen hallinnon potentiaalinen asiakasjoukko kattaa jokaisen kansalaisen. Kaikkia on siis pystyttävä palvelemaan. Koska yhteispalvelut tulevat perustumaan internetin käyttöön, täytyy internetin käyttöön tarjota myös mahdollisuus niille ihmisille, joilla ei tähän muuten mahdollisuutta ole. Tämä tavoite saavutetaan esimerkiksi erityisillä yhteispalvelupisteillä [ARS02], joita Suomessakin on jo 180 kappaletta [Yht04].

Demokraattisen valtion julkisessa hallinnossa ei pitäisi olla ”yrityssalaisuuksia”. Täten prosessien kulussa ja tiedon saannissa on noudatettava avoimuutta, toisin kuin yksityisissä yrityksissä. Läpinäkyvyydestä puhutaan enemmän seuraavassa luvussa. Avoimuus koskee vain hallintoa ja sen prosesseja, nimittäin kansalaisten tietosuojaan on kiinnitettävä erityistä huomiota [CCD03].

## **2.4 Vaatimuksia yhteispalveluille**

Perimmäisenä lähtökohtana yhteispalveluissa on, tai ainakin pitäisi olla, asiakaslähtöisyys. Tämä tarkoittaa sitä, että toiminnot eivät perustu organisaation sisäiseen rakenteeseen, vaan asiakkaan tarpeisiin. Yksityisellä puolella asiakaslähtöisyydestä on tullut fraasi, josta lähes jokainen yritys puhuu. Kuitenkin palvelut saattavat myös yksityisellä puolella olla varsin yksikkökeskeisiä, joka johtaa asiakkaan ”pompotteluun” [MaM03]. Asiakkaan ongelmat pitäisi siis ratkaista kokonaisvaltaisemmin, eikä asiakkaan tarvitsisi tietää mitään palveluiden sisäisestä rakenteesta. Käytännössä palvelut tulisi koostaa aitojen elämäntilan-

teiden (life episode) tai liiketoimintatilanteiden (business situation) mukaan [PaM04]. Pardon ja Scholl [PaS02] kuvaavat epäonnistumisen oikopolkuja (shortcuts to failure) julkisen hallinnon tietojärjestelmäkehityksessä. Heidän tutkimustensa mukaan varmin tapa epäonnistua projektissa on asianosaisten (stakeholder) jättäminen kehityksen ulkopuolelle prosessin kaikissa vaiheissa. Tässä skenaariossa asiakaslähtöisyys on unohdettu kokonaan. Marche ja McNiven [MaM03] luokittelevat julkisen hallinnon tarjoamat sähköiset palvelut hallintokeskeisiin (government-centric) ja asiakaskeskeisiin (customer-centric).

Kokonaisvaltaisuus johtaa siihen, että myös organisaation on muututtava prosessien mukana. Julkinen hallinto voi siis varautua isoihinkin organisaation muutoksiin [MaM03, PaM04], joiden ongelmiin edellisessä luvussa viitattiin. Yhteistyön lisäksi vaaditaan myös tiedon jakamista [PaM04], joka saattaa olla joillekin julkisen hallinnon yksiköille outoa [PHA01]. Muutosjohtaminen saattaakin olla yksi yhteispalveluiden toteutumisen avaintekijöistä [PHA01], ja varsinkin tiedon jakamisen kysymyksiin on paneuduttava [WiT00, PaM04]. Täydellinen prosessien uudelleenjärjestely voi olla liian raskas operaatio, mutta yhden pysähdyksen portaalien rakentaminen tulee vaatimaan prosessien uudistamista ainakin jollakin tasolla [PaM04].

Kuten edellisessä luvussa mainittiin, julkisen hallinnon on pidettävä huolta siitä, että jokaisella kansalaisella on mahdollista saada palvelua. Julkinen hallinto ei siis voi lähteä pankkien linjalle, jossa pakotetaan ihmiset internetin käyttöön. Vaihtoehtoisia kanavia on oltava olemassa. Lisäksi internetin käytössä on otettava huomioon erityisryhmät, kuten ikääntyvä väestö. Portaalien käytettävyyden suunnittelussa on otettava huomioon kaikki erityisryhmät [DoM02]. Fyysinen palvelutiski tuskin tulee näillä näkymin katoamaan, vaikka sen merkitys väheneekin [Len02].

Vaihtoehtoisten kanavien olemassaolo ei kuitenkaan tarkoita sitä, että henkilökohtaista tai räätälöityä palvelua saa vain fyysisellä palvelutiskillä. Monet julkisen hallinnon kanssa hoidettavat asiat ovat monimutkaisia, ja jos niitä tarjotaan verkossa, sovelluksesta on myös löydettävä älykkyyttä. Käyttöliittymän käytettävyydelle on asetettava korkea vaatimustaso, jotta käyttäjät tarvitsisivat mahdollisimman vähän erillistä ohjetta. Portaalien tulisi toisin sanoen toimia muuntimena byrokraattisen ammattikielen ja jokapäiväisen elämän välillä [TrW01, Kre02].

Prosessien läpinäkyvyyttä (transparency) voidaan parantaa tekniikan avulla. Tässä yhteydessä läpinäkyvyydellä tarkoitetaan sitä, että kansalainen näkee hallinnon prosessin ”läpi” ja ymmärtää sen. Sanalla ”transparency” on myös toinen merkitys, jonka voi suomentaa tuntumattomuudeksi. Sillä tarkoitetaan, että kansalaiselta peitetään turhat yksityiskohdat ja byrokraattinen monimutkaisuus. Kansalaisen ei tarvitsisi asian hoitaakseen tietää, mikä virasto on vastuussa kyseisestä asiasta. Kansalaiselta pitäisi siis toisaalta peittää asioita ja toisaalta paljastaa tiettyjä asioita. Esimerkiksi jossain hallinnon prosessissa olisi mahdollista nähdä, kuinka kauan kukin vaihe vie, kuka on vastuussa kustakin vaiheesta, ja keneen voi ottaa yhteyttä vauhdittaakseen prosessin etenemistä [MaM03]. Varsinkin vastuukysymykseen on kiinnitettävä huomiota [KIW02], koska vastuun mahdollinen katoaminen on yksi sähköisten palveluiden uhkatekijä.

Sähköisen kaupankäynnistä ja pankkipalveluista saatujen kokemusten perusteella voidaan todeta, että luottamus on huomattavan tärkeää, jotta palvelulle saadaan käyttäjiä [KIW02, MSB03]. Läpinäkyvyys edesauttaa luottamuksen saavuttamisessa, mutta myös tietoturvan ja tietosuojan on oltava kunnossa, koska julkisissa verkkopalveluissa saattaa liikkua arkaluonteista tietoa. Ainakin aiemmin mainitun tutkimuksen [ScE03] mukaan, luottamus julkiseen hallintoon on lähtökohtaisesti hyvällä tasolla. Koska käyttäjä joka tapauksessa tunnustetaan palvelussa, järjestelmään kannattaa myös rakentaa myös henkilökohtaista räätälöintiä [Kre02].

Yhteispalveluiden toteuttamisessa on otettava huomioon, että muitakin sidosryhmiä kuin asiakkaat, on olemassa. Muita sidosryhmiä ovat itse palvelun tuottajat (esimerkiksi verovirasto), välikäsi (joka on vastuussa järjestelmän käytöstä), kolmannet osapuolet (esimerkiksi pankit), täydentävät välikädet (joita saatetaan tarvita prosessissa) ja teknologian tarjoajat [Lam03].

Palveluiden kehittämisessä on pidettävä mielessä asteittainen kehittäminen. Julkinen hallinto on perinteisesti kehittynyt pienin askelin [Wes04], ja myös yhteispalveluita on kehitettävä näin. Luvussa 2.1 esitellyt kypsyystasot ovat hyvä lähtökohta asteittaiselle kehittämiselle niin, että yhteispalveluita kehitetään vaiheittain korkeammalle kypsyysasteelle. Kertarysäyksellä tapahtuva iso muutos ei ole realismia [PHA01, Wes04, Len02]. Strategiana on joko integrointi tai vähittäinen migraatio, jossa heterogeenisiä järjestelmiä ryhdy-

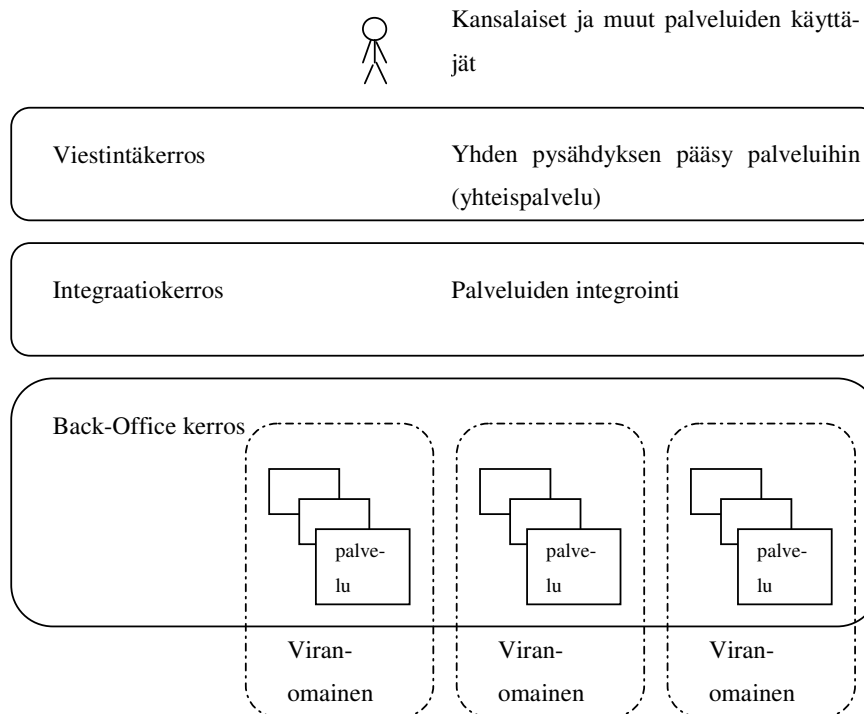
tään pikkuhiljaa muuntamaan [MeP01]. Tässä tutkielmassa tarkastelun kohteena on integrointi.

Kun tehdään yhteenvetolista edellä mainituista vaatimuksista, päästään hyvin samankaltaiseen listaan kuin mitä Lenk ja Traunmüller [LeT02] ovat esittäneet:

1. kohderyhmien tarpeisiin keskittyminen
2. monikanavaisuus
3. palvelun monimutkaisuuden selvittäminen
4. riittävä palveluiden integrointi
5. tietoturva
6. tiedon suojaus ja läpinäkyvyys
7. luotettavuus ja käytettävyys

Ensimmäisellä tarkoitetaan asiakaslähtöisyyttä ja erityisryhmien vaatimuksia. Monikanavaisuus viittaa mahdollisuuteen käyttää tietokoneen lisäksi esimerkiksi kannettavia laitteita, mutta myös perinteisiä palvelukanavia. Kolmannella kohdalla viitataan monimutkaisten työkulkujen aiheuttamiin ongelmiin. Integroinnilla tarkoitetaan mahdollisuutta todelliseen yhden pysähdyksen palveluun. Tietoturvalla viitataan tunnistukseen, todentamiseen, dokumenttien ja tietoliikenteen kiistattomuuteen. Tiedon suojauksella tarkoitetaan kansalaisten yksityisyyden turvaamista, ja läpinäkyvyydellä mahdollisuutta toimeksiantojen ja päätösten tarkasteluun. Seitsemäs kohta viittaa hyvään käyttöliittymäsuunnitteluun. Edellä esitettyä listaa käsitellään tästä eteenpäin yhteispalveluiden teoreettisina vaatimuksina.

Yhteispalvelun rakenne voidaan esittää kuvana (Kuva 1), kuten Pappa ja Makropoulos [PaM04] ovat esittäneet.



Kuva 1. Yhteispalveluportaali [PaM04].

*Viestintäkerros* käsittää pääsyn kaikkiin palveluihin yhdestä paikasta, saumattomat palvelut (palvelun sisäinen monimutkaisuus ei näy asiakkaalle), monikanavaisuuden, tietoturvan ja –suojan, sekä hyvän käytettävyyden. *Integraatiokerros* on nimensä mukaisesti vastuussa erillisten palveluiden suorituksesta yhtenä käyttäjälle näkyvänä palveluna (elämäntapahtuma). *Integraatiokerros*ssa on käytettävä jotain yhteistä toimintamallia, esimerkiksi web-palveluita. *Back-Office kerros* hoitaa jonkin tuotetun palvelun osan. Tietyille viranomaiselle tulee jokin palvelupyyntö (ja mahdollisesti parametreja) ja viranomaislainen toteuttaa palvelun ja sen tuotokset. Sisäistä suoritustapaa ei ole kiinnitetty, eikä yksittäisen viranomaisen tarvitse olla tietoinen koko prosessista, jonka osana kyseisen viranomaisen tuottama palvelu on [PaM04].

## 2.5 Viitemalli

Edellä esitetyn perusteella yhteispalveluille voidaan rakentaa viitemalli, jonka pohjalta voidaan edelleen rakentaa teknistä arkkitehtuuria. Viitemallin täytyisi ratkaista kolme ongelmaa: palveluiden luokittelu sisällön perusteella (*palveluontologiat*), miten palveluihin viitataan (*palvelun nimeämis-skeemat*) ja miten varastoidaan palveluiden metatiedot (*pal-*



*velutarjousvarastot*) [GLL02]. *Palveluontologiat* (service ontology), *palvelun nimeämiskeemat* (service naming scheme) ja *palvelutarjousvarastot* (service repository) muodostavat abstraktin palvelun viitemallin [GLL02].

Kun itse palvelut on määritelty, on määriteltävä myös *palveluiden kutsuun liittyvät mallit* (service invocation model). Näitä tarvitaan tiedon esitystapojen ja kommunikaatiokanavien heterogeenisuuden takia [GLL02]. Palvelun kutsumallien tulisi pystyä abstrahoimaan palvelukutsujen (service invocation calls), -kanavien (channel parametrization) ja -sanomien (information transliteration) yksityiskohdat [GLL02].

Yksi palvelupyynnö saattaa vaatia useampien tahojen ja prosessien yhteistyötä. Tämä vaatii *palvelun koordinoitimalleja* (service co-ordination model), joiden tulee selviytyä monimutkaisista prosesseista ja synkronisoinnista. Mallien tulee selviytyä atomisuusvaatimuksista, epädeterministisistä tilanteista, sekä prosessin kulusta ja tilasta on kyettävä raportoimaan [GLL02].

Edellisen perusteella voidaan muodostaa OSP (one-stop service provider) kerrosmalli (Kuva 2). Kuvassa näkyvät myös teknologiaehdotukset kuhunkin kerrokseen.

abstrakti palvelun koordinoitimalli	koordinointisuunnitelman laatiminen	koordinointisuunnitelman hallinta	WFMS, agenttitekniikat
	koordinointisuunnitelman toimeenpano		
abstrakti palvelun kutsu-malli	tiedon translitterointi	kanavien parametrisointi	XML, DHTML, SOAP
	palvelukutsut		
abstrakti palvelun viitemalli	palveluiden nimeäminen	palvelutarjousvarasto	WSDL, UDDI, DBMS
	palveluontologiat		
viestintäalusta	palvelun pyyntösanomien	tulosten paluukanava	Internet/HTTP, langaton tietoliikenne, paikkatietojärjestelmät

Kuva 2. OSP-kerrosmalli ja siihen liittyviä teknologioita [GLL02; Cro02b, s.10-13].

Viitemalli on tehty osana CB-BUSINESS projektia. Viitemalli pyrkii ratkaisemaan yhteispalveluiden heterogeenisuuden aiheuttamat ongelmat teoreettisella tasolla. Mallissa on rohkeasti pyritty viemään ongelmat abstraktimpaan viitekehukseen (vrt. tekninen arkkitehtuuri), ja tähän on hyvät perusteet, koska ratkaisevat teknologiat ovat vielä kehittyvässä

vaiheessa. Malli ei ole kuitenkaan kovin selkeä, ja sitä käsittelevät artikkelit [GLL02 ja Cro02b, s. 7-14] on luettava tarkkaan ja moneen otteeseen, jotta asia aukeaa. Tietojenkäsittelytieteessä runsas kerrosmallien käyttö ei myöskään ole aina perusteltua, koska kerrosten keskinäinen järjestys ei läheskään aina ole itsestään selvää. Esitetty kerrosmalli on kuitenkin tällä hetkellä ainoa tämän alueen teoreettinen viitemalli.

### 3 Sovellusintegraatio

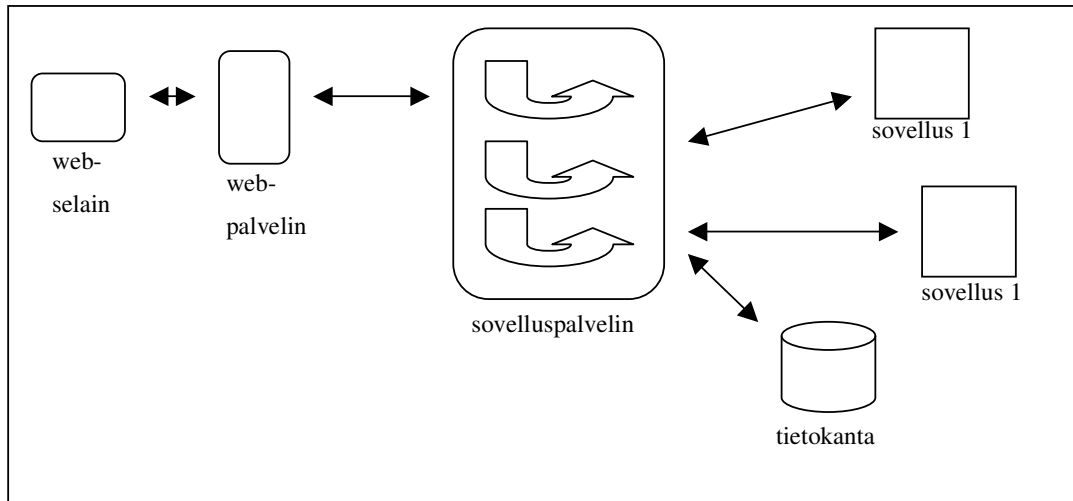
Julkishallinnon tietojärjestelmät ovat kehittyneet pieninä saarekkeina, eli kullakin virastolla on omat itsenäiset järjestelmät. Toiminnan tehostamiseksi heterogeeniset järjestelmät on saatava keskustelemaan keskenään ja toimimaan yhteistyössä. Yhden ison järjestelmän rakentaminen ei ole järkevää ja voi olla mahdotonta. Sovellusintegraation avulla erilliset heterogeeniset järjestelmät saadaan toimimaan kuin yksi iso järjestelmä. Perinteiset väliohjelmistot (traditional middleware) ovat olleet aikaisemmin ratkaisuna, mutta tilanteessa jossa integroinnissa on useita erillisiä tahoja, nämä perinteiset menetelmät eivät enää ole käyttökelpoisia [Lin01, s. 11-13]. Uudemmat väliohjelmistoratkaisut kuitenkin mahdollistavat monipuolisemman integroinnin.

Linthicum jakaa sovellusintegraation viiteen tyyppiin. Näitä ovat *tietosuuntautunut* (data-oriented), *sovellusrajapinta-suuntautunut* (application interface-oriented), *metodisuuntautunut* (method-oriented), *portaalisuuntautunut* (portal-oriented) ja *prosessi-integraatiosuuntautunut* (process integration-oriented) [Lin01, s. 26-30]. Yhteispalveluiden kannalta olennaisin on *portaalisuuntautunut integraatio*, joten ensimmäinen alaluku käsittelee portaaleita. Tämän jälkeen käsitellään integraatioteknologioita ja –standardeja. Edellisen jaottelun mukaan portaaleista puuttuu tietynlainen älykkyys, joka löytyy tämän jaottelun *prosessisuuntautuneesta integraatiosta*. Niinpä yhteispalveluiden toteutus vaatii myös prosessisuuntautunutta integrointia. Luvun lopussa käydään läpi työnkulkujen ja transaktioiden hallintaa, joka liittyy juuri edellä mainittuun prosessisuuntautuneeseen integrointiin.

Sovellusintegraatiot voidaan jakaa myös toisella tavalla. Nämä paradigmat ovat interconnectivity, interoperability ja intermediation [GMG01]. Ensimmäinen viittaa matalan tason tiukkaan integrointiin (vrt. tietosuuntautunut), toinen rajapintojen integrointiin (vrt. sovellusrajapinta) ja kolmas sanomanvälittäjätyyppiseen ratkaisuun. Näistä viimeisin on yhteispalveluiden kannalta käyttökelpoisin.

### 3.1 Portaalit

Sovellusintegraation yhteydessä portaalissa on kyse siitä, että käyttäjälle näkyy vain yksi käyttöliittymä, usein web-sivu, joka peittää alla toimivat heterogeeniset järjestelmät. Portaalien perusarkkitehtuurin osat ovat: web-asiakas, web-palvelin, tietokantapalvelin, back-end sovellus ja sovelluspalvelin (kuva 3) [Lin01, s. 97-98]. Tämä on teknisempi esitys kuvasta 1. Portaalien toteutuksen löytyy useita valmiita kaupallisia ja avoimen lähdekoodin alustoja [Cro02b, 24-34].



Kuva 3. Portaaliarkkitehtuuri.

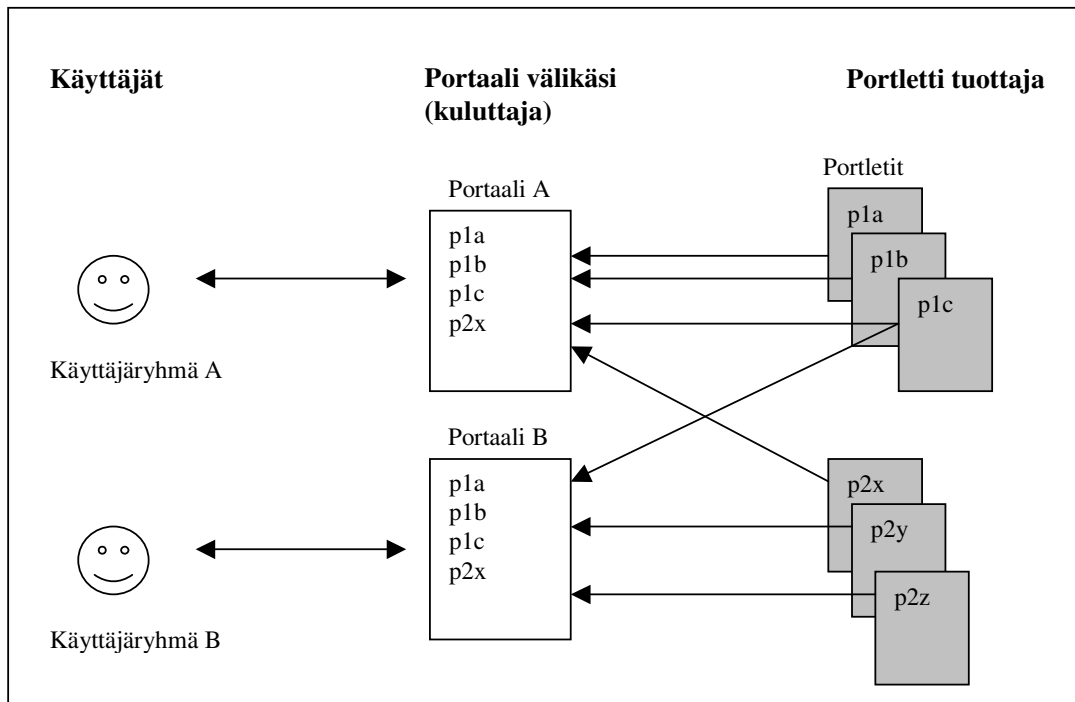
Web-palvelin vastaa yksinkertaisimmillaan selaimen pyyntöön tietystä sivusta ja toimittaa sivun selaimelle käyttäen HTTP-protokollaa. Tietokantapalvelin toimii samantyyppisellä asiakas-palvelija periaatteella, eli vastaa asiakkaan pyyntöihin, yleensä toimittaen tietoa. Back-end sovellukset tai loppusovellukset voivat olla mitä tahansa heterogeenisiä sovelluksia, joiden yhteistyön vuoksi integrointiin on ryhdytty. Sovelluspalvelin toimii välittäjänä arkkitehtuurin eri osien välillä, ja sovelluspalvelimen vastuulla on järjestelmän toimintalogiikka. Tällainen perusarkkitehtuuri ei ole riittävä monimutkaisen yhteispalvelujärjestelmän toteutukseen, mutta antaa yleiskuvan periaatteesta.

Portaali toimii palvelun käyttöliittymänä, ja kuten aiemmassa luvussa todettiin, käytettävyys on yksi toimivan palvelun avaintekijöitä. Käyttöliittymän toiminta ei saa perustua sisäisiin prosesseihin, vaan asiakkaan käyttötapauksiin (elämäntapahtumat), kuten hyvässä käyttöliittymäsuunnittelussa on tapana tehdä. Eräs visio [TrW01] on myös, että portaalihin on integroitu älykäs henkilökohtainen avustaja. Tällaiseen ajatukseen on syytä suhtautua

tua epäilevästi, koska tällaisten henkilökohtaisten avustajien (kuten Office-avustaja) toiminta on ainakin tähän mennessä ollut enemmän ärsyttävää kuin avustavaa. Julkisen hallinnon portaalien käytettävyydessä on myös otettava huomioon erityisryhmät ja mahdollistettava henkilökohtainen sisällön räätälöinti [GaG02].

Portaalin toimivuutta voidaan arvioida neljän ulottuvuuden perusteella. Nämä ovat: avoimuus, räätälöitävyys, käytettävyys ja läpinäkyvyys [GaG02]. Avoimuudella tarkoitetaan sitä, kuinka hyvin portaalin kautta päästään kaikkeen julkisen hallinnon tarjoamaan tietoon ja palveluihin. Räätälöitävyydellä tarkoitetaan mahdollisuutta muokata käyttöliittymää juuri omiin tarpeisiin sopiviksi. Käytettävyydellä tarkoitetaan käytön helppoutta, kuten navigointia ja intuitiivisuutta. Läpinäkyvyydellä tarkoitetaan portaalin tarjoamaa mahdollisuutta seurata oman prosessin tilaa, päätöksen syitä ja vastuukysymyksiä.

Portaali voidaan koostaa tuottaja-kuluttaja mallin mukaisesti portleteista. Portletti toteuttaa jonkin toiminnallisuuden ja portleteilla voi olla eri tuottajia. Portlettimallissa (Kuva 4) kuluttajana on välikäsi, joka kokoaa erillisistä portleteista kokonaisuuden. Loppukäyttäjät taas käyttävät välikäden tuottamaa palvelua, eli portaalia. Kyseistä tekniikkaa kutsutaan nimellä WSRP (Web Services for Remote Portlets) [Web03]. Kun portletti on kerran sisällytetty portaaliin, portletin tuottaja voi muokata sitä etänä ilman että välikäsi joutuu tekemään mitään muutoksia. Itse portletin toteutustekniikkaa ei ole kiinnitetty, vain se miten sisällytys portaaliin tehdään [GaG02]. WSRP mahdollistaa kytke ja käytä (plug-and-play) tyyppisen ratkaisun erillisten palveluiden sisällyttämisessä portaaliin [Web03].



Kuva 4. Yksinkertainen WSRP-palvelu [GaG02].

### 3.2 Integraatioteknologiat

Teknologioita, jotka mahdollistavat sovellusintegraation, kutsutaan väliohjelmistoiksi (middleware). Yksinkertaisimmillaan väliohjelmisto mahdollistaa kahden erillisen komponentin kommunikoinnin. Väliohjelmistojen avulla voidaan peittää alla olevien ohjelmistojen yksityiskohdat. Väliohjelmistot kuitenkin asettavat tietyt rajat kommunikoinnille. Väliohjelmiston nimi tulee siitä, että se on ylimääräinen ohjelmistokerros itse sovelluksen ja käyttöjärjestelmän verkkopalveluiden välissä [TaS02, s. 36].

Kiinnostavimpia väliohjelmistotyyppisiä yhteispalveluiden kannalta ovat sellaiset, jotka kykenevät transaktioihin. Transaktiolla tarkoitetaan jotain tapahtumaa, joka voi koostua useista erillisistä vaiheista eri järjestelmissä. Ensimmäisen sukupolven transaktioihin kykenevä teknologia oli TP-monitorit [Lin01, s. 143-145]. Uudempaa teknologiaa sen sijaan edustavat sovelluspalvelimet ja sanomanvälittäjät.

Sovelluspalvelimen tehtävänä on hoitaa sovelluksen toimintalogiikka, ja toimia välikätenä järjestelmän eri osien välillä. Sovelluspalvelin on itse asiassa vain kehittyneempi TP-

monitori, joka on suunniteltu web-sovelluksia silmälläpitäen [Lin01, s. 145-146]. Sovelluspalvelimien toteutuksessa Java-teknologiat tarjoavat varteenotettavan vaihtoehdon. JavaSoft ei kuitenkaan tee itse tuotteita, vaan tarjoaa ainoastaan spesifikaatiot ja teknologian [Lin01, s. 215]. Sovelluspalvelimista Javaa käyttävät muun muassa WebLogic, Net-Dynamics ja Netscape Application Server [Lin01, s. 223]. Myös vanhemmat TP-monitoriympäristöt, kuten IBM:n CICS, ovat siirtyneet Javaan [Lin01, s. 224]. Loppuvuodesta 1999 julkaistu Java 2.0 Enterprise Edition (J2EE) on pitkälle kehitetty väliohjelmistoteknologia. Erityisesti siihen sisältyvä EJB (Enterprise Java Beans) mahdollistaa hajautetut transaktiot.

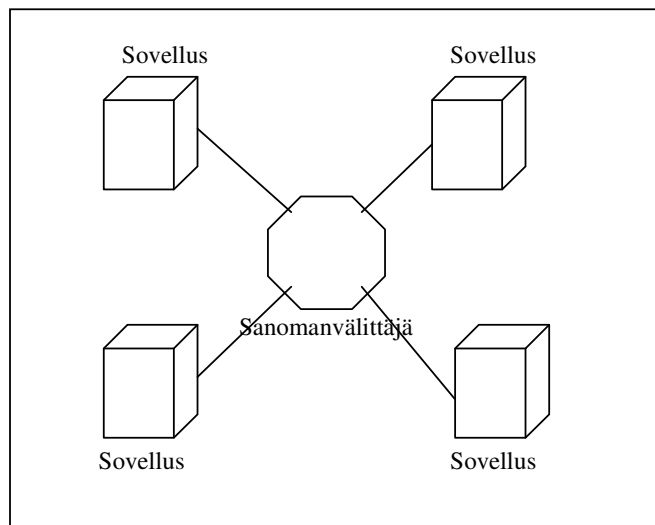
Sanomanvälittäjät ovat uusinta integraatioteknologiaa. Sanomanvälittäjien suurin lisäarvo on se, että ne mahdollistavat erilaiset sanomamuodot järjestelmän osien välillä. Sanomanvälittäjä siis muuntaa sovelluksesta A lähtevän viestin sovelluksen B ymmärtämään muotoon. Tämä piirre tuo joustavuutta verrattuna perinteisempiin väliohjelmistoihin. Sanomanvälittäjät tarjoavat myös useita palveluita, jotka on perinteisesti jouduttu hoitamaan itse sovelluksissa. Palvelut voidaan jaotella seuraavasti [Lin01, s. 232]:

- Sanoman muunnos (message transformation)
- Älykäs reititys (intelligent routing)
- Sääntöjen prosessointi (rules processing)
- Sanomien säilytys (message warehousing)
- Vuon valvonta (flow control)
- Säilytyspalvelut (repository services)
- Hakemistopalvelut (directory services)
- Hallinto (management)
- Sovellusliittymät ja sovittimet (APIs and adapters)

Älykkäällä reitityksellä tarkoitetaan sitä, että sanomanvälittäjä osaa tunnistaa sanomasta ja lähettäjistä, mille sovellukselle se on lähetettävä ja mitä sille on tehtävä [Lin01, s. 241]. Sääntöjen prosessoinnilla tarkoitetaan sanomanvälittäjän ohjelmoitavuutta, jolla voidaan ohjata mm. sanomien muunnosta ja reititystä [Lin01, s. 242-243]. Sanomanvälittäjä voi myös säilöä käsittelemiään sanomia [Lin01, s. 243-245]. Säilytyspalveluilla tarkoitetaan tietokantamaista tietoa sovelluksista, jotka ovat yhteydessä sanomanvälittäjään [Lin01, s. 245-246]. Hakemistopalveluilla viitataan lähinnä tietoliikenteen hakemistopalveluihin, kuten DNS-palveluun [Lin01, s. 248]. Hallintokerroksen tehtävänä on valvoa sanomanvä-

littäjän sanomaliikennettä, ja tarvittaessa puuttua esimerkiksi suorituskykyongelmiin [Lin01, s. 248-249]. Sovittimet ovat välikerroksia sanomanvälittäjän ja sovellusten välillä. Sovitinstandardeja ei ole kuitenkaan vielä muodostunut [Lin01, s. 249]. Sovittimien lisäksi sanomanvälittäjien ja sovellusten välissä on myös usein sovellusliittymä. Sanomanvälittäjät ovat tällä hetkellä vielä kehittyvässä vaiheessa ja uusia toimintoja tulee jatkuvasti [Lin01, s. 254].

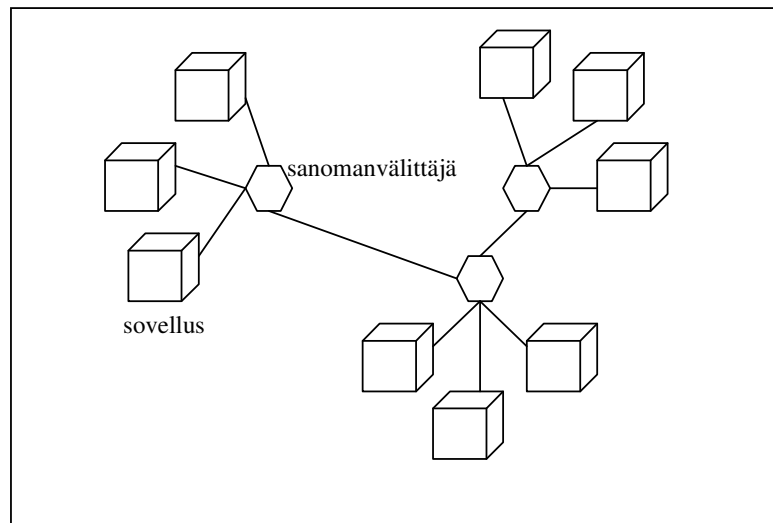
Sanomanvälittäjien topologiassa yksinkertaisin ratkaisu on tähdenmuotoinen (Kuva 5).



Kuva 5. Hub-and-spoke topologia [Lin01, s.255].

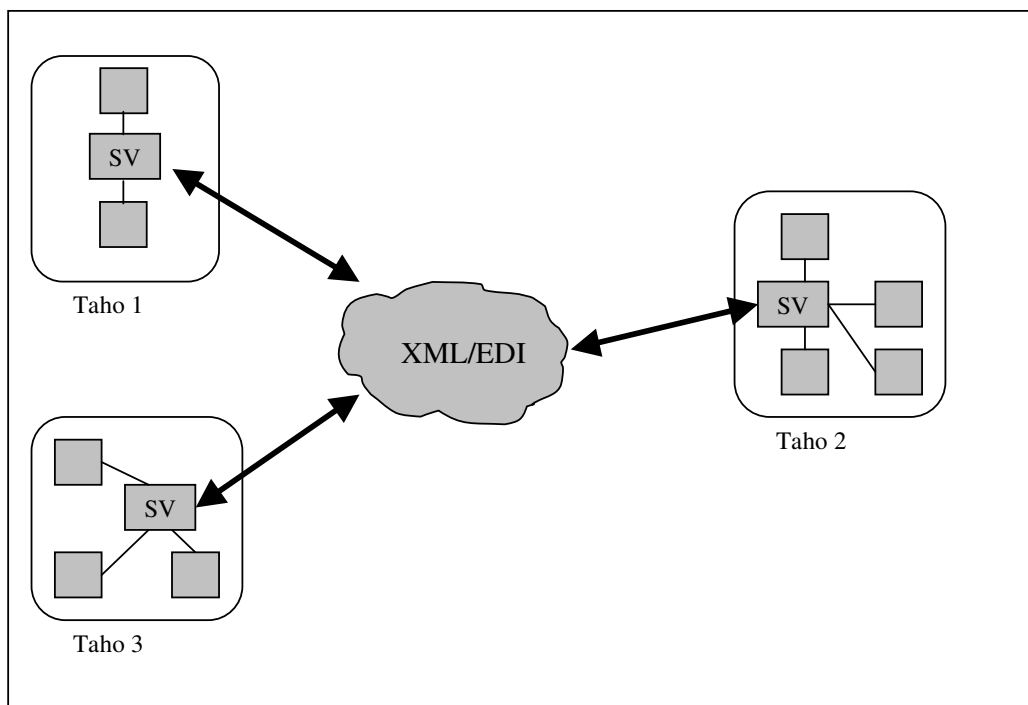
Tähdenmuotoinen topologia ei skaalaudu hyvin, joten suurempiin järjestelmiin soveltuu paremmin multihub-topologia (Kuva 6), jossa on useampia sanomanvälittäjiä [Lin01, s. 255-256]. Jos multihub rakennetaan tuntumattoman hajautetun järjestelmän tapaan, on mahdollista tasata sanomanvälittäjien kuormaa ja selviytyä tilanteista, joissa yksi solmu kaatuu.





Kuva 6. Multihub topologia.

Kolmas topologiavaihtoehto on liittotopologia (federated). Myös siinä on useita sanomavälittäjiä, jotka keskustelevat keskenään, mutta sovellukset on sidottu tiettyyn sanomavälittäjään, ja topologian sanomavälittäjät ovat erilaisia. Sanomavälittäjät keskustelevat keskenään esimerkiksi XML:n avulla (Kuva 7) [Lin01, s. 255-257].



Kuva 7. Liittotopologia.

Eräs ongelma, jota sanomanvälittäjilläänkin pyritään ratkaisemaan, on saman tiedon esittäminen eri tavalla eri sovelluksissa. Esimerkiksi ”palkka” voi tarkoittaa yhdessä järjestelmässä samaa kuin ”tulot” toisessa. Toisaalta ”tulot” voi tarkoittaa kahdessa järjestelmässä kahta eri asiaa. Tiedon esittämisen yhdenmukaistamiseen täytyy olla siis jokin keino. Eräs tekniikka on ontologioiden käyttö. Ontologioille ei ole tietojenkäsittelytieteessä yleisesti hyväksyttyä määritelmää. Ontologiat kuvataan kuitenkin yleensä joukkona termejä puumaisessa rakenteessa. Yleisemmät termit ovat ylempänä puussa ja erikoisemmat termit syvempänä [Hov03]. Esimerkiksi termillä ”palkka” voi olla alatermit ”virasto1-palkka” ja ”virasto2-palkka”, ja näillä omat määritelmät [Hov03]. Ontologiat pyrkivät siis antamaan metatietoa termeistä, eli merkkijonoista.

### **3.3 Integraatiostandardeja**

Integraatioteknologioiden kehittyessä myös standardeja on alkanut muodostua. Standardit ovat kuitenkin vielä uusia, joten on vaikea nähdä, mitkä niistä tulevat varmasti menestymään jatkossa. Yhteis palveluprojektit ovat isoja hankkeita, joten väärän standardin valinta voi osoittautua muutaman vuoden päästä virheeksi. Tässä luvussa ei esitellä läheskään kaikkia standardeja, vaan keskitytään muutamaamaan lupaavimpaan standardiin. Moni standardista perustuu XML-kieleen, joten se esitellään ensin.

#### **3.3.1 XML**

XML (eXtensible Markup Language) on laajalti tunnettu kieli ja standardi. Kyseessä on metakieli, joka antaa informaatiota datan sisällöstä ja merkityksestä, ja auttaa siten ratkaisemaan edellisessä luvussa käsiteltyä erilaisten tietomuotojen ongelmaa. XML:n avulla erilliset sovellukset, jotka eivät muuten ymmärtäisi toisiaan, voivat kommunikoida keskenään [Lin01, s. 263]. Osapuolten täytyy vain muuntaa tieto johonkin yhteisesti sovittuun XML-muotoon, ja kuvata yhteinen DTD (document type definition).

XML-dokumentissa olevat tiedot kuvataan alku- ja lopputageilla. Sisäkkäiset tagit ovat myös mahdollisia. XML-dokumentti on hyvinmuodostettu (well-formed), kun siinä on vain yksi juurielementti ja jokainen elementti alkaa ja loppuu samalla elementillä. Lisäksi dokumentin on noudatettava XML-spesifikaatiota ja jokainen jäsennetty entiteetti (parsed

entity) on myös hyvinmuodostettu [W3C04a]. XML-jäsennin (parser) on ohjelma, joka muuntaa sovelluksen tuottamaa tietoa XML-dokumentiksi tai toisin päin. Jäsentimet tulevatkin tulevaisuudessa kuulumaan väliohjelmistokerroksen [Lin01, s. 267]. Esimerkiksi jos sovelluksen tuottama datayksikkö on seuraavanlainen:

```
Kerttu Suomalainen
120344-567Q
Pääkatu 1
00001
Helsinki
```

Jäsennin voisi tuottaa edellisestä datasta seuraavanlaista XML-koodia:

```
<kansalainen>
  <nimi> Kerttu Suomalainen</nimi>
  <hetu>120344-567Q</hetu>
  <lähiosoite> Pääkatu 1</lähiosoite>
  <postinumero>00001</postinumero>
  <kaupunki> Helsinki</kaupunki>
</kansalainen>
```

Jokainen taho voi periaatteessa luoda oman metadata-mallin, mutta yhteistyön kannalta on järkevää luoda myös metadatastandardeja tietyille teollisuuden aloille. Laajalti hyväksytyjä metadatastandardeja ei ole vielä syntynyt [Lin01, s. 267]. XML itsessään ei ole ratkaisu mihinkään, vaan tarjoaa lähinnä perustan ja standardin. XML:n huonona puolena on sen vaatima tila, tekstimuodossa oleva data vie moninkertaisen tilan verrattuna binäärimuotoiseen [Lin01, s. 271]. Levytilan kannalta tämä ei ole ongelma, mutta tiedonsiirto voi aiheuttaa pullonkaulan. XML on kuitenkin todellinen standardi, lähes jokaisen tässä tutkielmassa esitetyn projektin ja pilotin teknologia nojaa XML-kieleen.

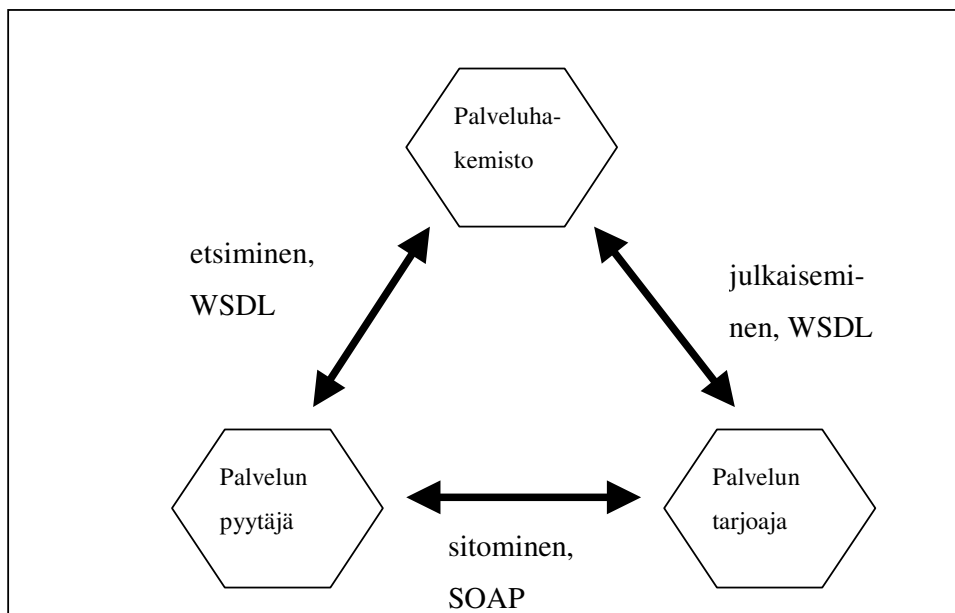
### 3.3.2 Web-palvelut

Web-palvelut (Web Services) on, kuten XML-kielikin, W3C:n kehittämä suositus sovellusten hajautukseen. Web-palveluiden kehitystyö on vielä kesken [W3C04b], mutta suosituksista on alkanut muodostua standardi. Web-palvelut arkkitehtuuriin katsotaan kuuluvan XML-pohjaiset suositukset SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language) ja UDDI (Universal Description, Discovery and Integration) [W3C04b]. SOAP määrittelee yhteydenpidossa käytettyjen viestien rakenteen, WSDL kuvaa web-palveluratkaisun rajapinnan ja UDDI määrittelee käytettävän palveluhakemiston.

SOAP-viesti on XML-kieltä, ja se voidaan lähettää muun muassa HTTP-protokollaa käyttäen. XML ja HTTP-tuki siis tarvitaan, mutta muuten SOAP on ohjelmistosta ja laitteistosta riippumaton. Valmis SOAP-toteutus löytyy esimerkiksi Apachelta [Apa04]. SOAPin avulla voidaan toteuttaa myös etäproseduurikutsuja [W3C03]. SOAP ei kuitenkaan vielä skaalautu kovin hyvin ja määrittelystä löytyvät vain yksinkertaisimmat tietotyypit [MRB03]. Lisäksi SOAP itsessään ei tue minkäänlaista transaktion hallintaa [Erk02, s. 66], vaan transaktioiden hallinta on hoidettava muualla.

WSDL:n [W3C01] avulla palvelu voidaan kuvata abstrakteina portteina, jotka käsittelevät tietyn tyyppisiä viestejä. Viestit on sidottu tiettyyn verkkoprotokollaan ja viestimuootoon. Kuvauksen avulla saadaan selville se, miten tietyn palvelun kanssa on keskusteltava.

UDDIn [Udd02] avulla palveluhakemistoon rekisteröidään palveluita. Asiakas voi etsiä tarvitsemaansa palvelua UDDI-hakemistosta, ja kun palvelu löytyy, sen kuvaus voidaan saada WSDL-muodossa. UDDIn käyttöä voisi kuvata myös palveluiden mainostamiseksi. Palveluiden haku voidaan tehdä staattisesti tai dynaamisesti, jolloin sopivaa palvelua etsitään ja siitä neuvotellaan [Cro02b, s.36-37]. Kuvassa 8 on kuvaus web-palvelumallista.



Kuva 8. Web-palvelut malli [ViM02].

Eräs web-palveluiden parhaimpia piirteitä on yksinkertaisuus ja riippumattomuus. Koska kaikki perustuu XML-kieleen, erillisiä alustoja tai kehityskieliä ei tarvita. Myös WSDL, UDDI ja SOAP ovat jo nyt laajasti hyväksytyjä standardeja [Cro02b, s.36].

### 3.3.3 RosettaNet

RosettaNet on yli viidensadan organisaation konsortio, joka kehittää avoimia sähköisen liiketoiminnan standardeja [Ros04]. Kehityksen kohteena ovat prosessit ja mekanismit, jotka mahdollistavat erillisten tahojen yhteiset pelisäännöt sähköisessä liiketoiminnassa. RosettaNet on keskittynyt prosessien kuvaukseen, ja sen tärkein lisäarvo onkin kuvata, mikä prosessi tekee mitäkin ja missä [Lin01, s. 278].

RosettaNetin tärkeimmät osat ovat PIP (Partner Interface Process) ja sanastot. PIPit määrittelevät organisaatioiden väliset yhteiset prosessit. Jokaisessa PIPissä on kuvattu jonkin yhteisen prosessin kulku, mukaan lukien logiikka, viestien sisältö ja viestinvälitys [Lin01, s. 280-282]. Kyseessä on siis kahden prosessin (organisaatioiden omat prosessit) liittäminen yhteen tietyn standardin mukaan. RosettaNet on määritellyt valmiita PIP-segmenttejä ja klustereita [Lin01, s. 282-283], joita organisaatiot voivat käyttää tarpeen mukaan. PIPin käyttäminen vaatii siis ensin omien prosessien määrittelyä ja tunnistamista, jos sitä ei ole vielä tehty. Sanastot jakautuvat kahteen osaan, tekniseen sanastoon ja liiketoimintasanastoon [Lin01, s. 280]. Tekninen sanasto sisältää kaikkien tuotteiden ominaisuudet. Liiketoimintasanasto taas sisältään ominaisuudet yhteistyölle ja transaktioille.

RosettaNet on kehittänyt kerrosmallin kommunikaatioon, jossa kerrokset sisältyvät OSI-mallin sovelluskerrokseen [Lin01, s. 283-284]. Kerrokset ovat ylhäältä alaspäin: toimintokerros, transaktiokerros, prosessikerros, palvelukerros, agenttikerros, sanomankäsittelykerros ja siirtokerros. Toimintokerros (action layer) välittää liiketoimintatoimintoja ja transaktiokerros valvoo transaktioiden toteutumista. Prosessikerros kapseloi transaktion PIPin suoritusta varten. Palvelukerros tarjoaa verkkoon liittyviä palveluita. Agenttikerros on rajapinta muille sovelluksille. Sanomankäsittelykerros hoitaa asynkronisen informaation kuljetuksen. Siirtokerroksessa on mekanismi tiedon muunnoksia varten. Lisäksi sovelluskerroksen ulkopuolella on turvallisuuskerros valtuutusta ja tunnistamista varten [Lin01, s. 283-284]. PIP-sanomat on kuvattu XML-dokumentteina [Lin01, s. 285].

RosettaNetin etu muihin standardeihin on se, että siinä kuvataan myös prosessit tarkalla tasolla. Esimerkiksi yhteispalveluiden monimutkaisissa transaktioissa tämä on selvä etu.

### 3.4 Työnkulkujen hallinta

Heterogeenisessä hajautetussa järjestelmässä ei riitä, että järjestelmän osat kykenevät kommunikoimaan keskenään. Yksi loppukäyttäjälle näkyvä toiminto voi olla monimutkainen prosessi järjestelmän eri osien välillä. Tätä varten järjestelmän prosessien ymmärtäminen ja kuvaaminen on tärkeää. Tätä prosessien hallintaa kutsutaan työnkulkujen hallinnaksi (workflow management). Julkisessa hallinnossa, jossa monet prosessit ovat byrokraattisia ja useiden virastojen välisiä, työnkulkuihin on kiinnitettävä erityistä huomiota. Työnkulkujen hallintajärjestelmien (WFMS) avulla määritellään ja automatisoidaan toimintaprosessit, sekä eristetään toimintalogiikka ja eri järjestelmien integrointi toisistaan [PDE02]. Kaupallisia työnkulkujen hallintajärjestelmiä ovat esimerkiksi InTempo, Staffware, Visual and Panagon Workflo ja FORO-WF [Cro02b, s.15-17]. Julkisessa hallinnossa ongelmana voi olla prosessien epämääräisyys, joka ei johdu prosessien määrittelemättömyydestä vaan yksinkertaisesti epädeterministisistä työnkuluista [Kre02]. Tämä asettaa työnkulkujen hallinnalle melkoisia joustavuus- ja epälineaarisuusvaatimuksia. Tutkimusten mukaan [PaS02] prosessien ja työnkulkujen tarkka määrittely ja analyysi, sekä niihin liittyvien tietorakenteiden suunnittelu, on isojen projektien onnistumisen kannalta kriittistä.

#### 3.4.1 Transaktiot

Transaktiolla tarkoitetaan jotain toimintoa, esimerkiksi matkan varausta, joka koostuu useammasta pienemmästä toiminnosta. Transaktio on sarja peräkkäisiä tapahtumia, jotka joko kaikki toteutuvat tai sitten yksikään ei toteudu. Transaktion on jätettävä järjestelmä eheään tilaan, eli osittaista toteutumista ei saa tapahtua. Transaktio on siis yksi jakamaton, eli atominen kokonaisuus.

Transaktion tulisi toteuttaa ACID-omaisuudet [Lin01, s. 154]. Atomisuus (atomicity) tarkoittaa, että kaikki toiminnot toteutuvat, tai yksikään ei toteudu. Oikeellisuus (consisten-

cy) tarkoittaa, että järjestelmä on aina eheässä tilassa. Eristyvyys (isolation) tarkoittaa, että useat käynnissä olevat transaktiot eivät vaikuta mitenkään yksittäiseen transaktioon. Pysyvyys (durability) tarkoittaa, että transaktion suoriuduttua mikään häiriö ei voi enää muuttaa sen tilaa. Väliohjelmistoissa transaktiot on perinteisesti hoidettu TP-monitoreilla [Lin01, s. 155-158]. Web-pohjaisissa sovelluksissa transaktiot hoidetaan usein sovelluspalvelimilla [Lin01, s. 158-160], mutta niiden suorituskyky ja luotettavuus ei vielä yllä TP-monitoreiden tasolle [Lin01, s. 161].

### 3.4.2 Virtuaaliset organisaatiot

Virtuaalinen organisaatio (virtual enterprise, virtual organization) on jokin useamman erillisen tahon yhteenliittymä, jonka avulla halutaan saavuttaa yhteisiä tavoitteita. Yhteispalveluissa virtuaalinen organisaatio voi muodostua esimerkiksi useammasta valtion virastosta ja pankeista. Kuten yhteispalveluidenkin yhteydessä, myös virtuaalisissa organisaatioissa on eri elinkaaren vaiheita. Nämä vaiheet ovat virtuaalisen organisaation perustaminen, toiminnan kehittymisvaihe ja toiminnallinen vaihe [Tag01]. Tästä voidaan päätellä, että yhteispalvelun kypsyytaso ja virtuaalisen organisaation elinkaaren vaiheet ovat sidoksissa toisiinsa. Vasta toiminnallisessa vaiheessa, jossa syntyy tehokasta yhteistyötä ja prosessit etenevät kuin yhden organisaation sisällä, voi syntyä todellista yhteispalvelua. Web-palvelutekniikat (UDDI, WSDL, SOAP) ovat lupaavia tekniikoita elinkaaren kypsymisen suhteen [Cam03], koska ne tukevat virtuaalisen organisaation dynaamisuutta.

Virtuaalisen organisaation rakenne vaikuttaa teknisiin valintoihin. Jos jokin organisaation osapuoli on selvästi vahvempi kuin muut, se voi määrätä tiedon jaon periaatteet ja tiedon siirron protokollat [Tag01]. Yhtä mieltä ollaan kuitenkin siitä, että XML tulee olemaan useimpien standardien pohjana [Tag01]. Laajasti hyväksyttyä viitemallia ei kuitenkaan ole [Cam03], ja tämä jatkuvasti kehittyvän teknologian kanssa aiheuttaa ongelmia järjestelmien toteuttajille.

Virtuaalisen organisaation infrastruktuurissa voidaan nähdä kolme lähestymistapaa [Cam03]. *Kerros pohjainen* tai transaktiosuuntautunut lähestymistapa lisää olemassa olevien järjestelmien päälle uuden kerroksen. *Agenttipohjainen* lähestymistapa pohjautuu älyk-

käiden agenttien käyttöön. *Palveluliitto/palvelumarkkina*-lähestymistapa perustuu systeemiin, jossa osapuolet julkaisevat palvelujaan ja toiset osapuolet voivat niitä käyttää.

Kerros pohjainen lähestymistapa on perinteisin ja perustuu usein tunnettuihin tekniikoihin, kuten etäproseduurikutsut, CORBA ja RMI [Cam03]. Nykyisin tämäkin tekniikka pohjautuu yhä useammin XML-kieleen. Tästä huolimatta yleisesti hyväksyttyä viitemallia ei ole, ja järjestelmät ovat monimutkaisia, mutta silti ominaisuuksiltaan köyhiä [Cam03].

Agenttipohjaisessa lähestymistavassa on ainakin tutkijoiden [Cam03] mielestä paljon hyviä puolia ja mahdollisuuksia. Agenttitekniikat ovat pääosin Java-pohjaisia ja niiden väitetään tuovan järjestelmään älykkyyttä ja dynaamisuutta. Laajoja agentti-infrastruktuureja ei ole kuitenkaan vielä olemassa, vaan toteutetut ympäristöt ovat lähinnä tutkijoiden tekemiä prototyyppisiä. Agenttitekniikka kehittyy kuitenkin jatkuvasti.

Palveluliitto-lähestymistapa on sama kuin aiemmin käsitelty web-palvelumalli, jossa tekniikoina ovat WSDL, UDDI ja SOAP. Tätä lähestymistapaa on käytetty esimerkiksi FETISH-ETF-projektissa, jossa on muodostettu virtuaalinen organisaatio erilaisia turismia tukevista tahoista [Cam03]. Palveluverkostossa kukin paikallinen palvelun tarjoaja muodostaa oman solmunsa, joka tarjoaa palveluluettelon ja työkaluja palvelun käyttämiseen. Infrastruktuuri on toteutettu Java/JINI- ja WSDL-tekniikoilla.

Eräs muoto virtuaalisista organisaatioista ovat ammattimaiset virtuaaliset yhteisöt, PVC (professional virtual communities) [Cam03]. Ne ovat fyysisesti hajanaisia yhteisöjä, esimerkiksi eri yritysten insinöörejä, jotka työskentelevät yhteisen tavoitteen (esimerkiksi uusi tuote) eteen tietotekniikan tarjoamin apuvälinein. PVC:t eivät ole suoranaisesti kiinnostavia yhteispalveluiden kannalta. TeleCARE-projekti, jossa on muodostettu virtuaalisia yhteisöjä vanhojen ihmisten auttamiseksi [Cam03], on kuitenkin kiinnostava. Yhteispalveluiden kasvavana kohderyhmänä ovat nimittäin juuri eläkeikäiset. Projektissa on käytetty Java-pohjaista AGLET-alustaa, jossa käytetään liikkuvia agenteja [Cam03]. Projektin voisi siis luokitella myös agenttipohjaisen lähestymistavan alle.

Virtuaalisten organisaatioiden yhteydessä puhutaan myös käsitteestä ”breeding environment”, jonka voisi suomentaa kasvu-ympäristöksi. Taustalla on ajatus siitä, että dynaamisten ja pitkäikäisten virtuaalisten organisaatioiden muodostuminen on hidasta muun muassa



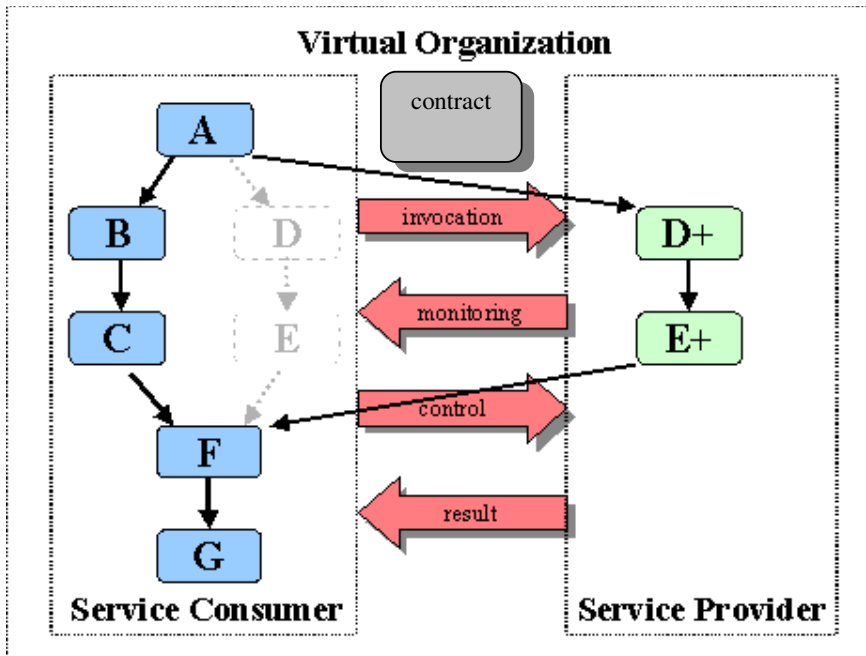
alussa olevan luottamuksen puutteen takia. Virtuaalisten organisaatioiden kasvuympäristöt (VBE, virtual breeding environment) [Cam03] olisivat palveluliittojen perusideaan pohjautuvia yhteisöjä, joissa olisi hakemisto palveluntarjoajista. Virtuaalisen organisaation muodostajat voisivat valita näistä hakemistoista sopivat kumppanit tarpeen mukaan. VBE pysyvänä organisaationa auttaisi luottamuksen ja yhteisten toimintamallien rakentamiseen, ja dynaamisen virtuaalisen organisaation syntymisen [Cam03].

Idea kasvuympäristöistä ei kuitenkaan välttämättä toimi julkisen hallinnon kohdalla, jossa ei ole kysymys liiketoiminnasta ja parhaiden yhteistyökumppaneiden etsimisestä. Julkisen hallinnossa virtuaalisten organisaatioiden koostumus on pysyvämpi, ja tämä tuo ajan mittaan samoja etuja kuin mitä VBE:llä haetaan. Julkinen hallinto tosin tekee yhteistyötä yksityisenkin sektorin kanssa, ja näissä yhteyksissä kumppaneita saatetaan vaihtaa. Kuten Evangelidis ym. [EAT02] mainitsevat: ”*..next generation of eGovernment is .. result of partnership between the public and private sector*”. VBE sopisi julkisen hallinnon ja yksityisen sektorin rajapintaan, mutta ei välttämättä julkisen hallinnon tahojen sisäiseen yhteistoimintaan. Kasvuympäristö voidaan nähdä myös tekniikoiden ja ”best practices” käytöjen kasvualustana, vaikka alkuperäinen lähde ei sitä tällaisena näekään. Tällaisessa mielessä kasvuympäristön käsite sopii mielestäni paremmin julkiseen hallintoon. Onhan tavoitteena kuitenkin asiakaslähtöisyys, ja juuri ”best practices” tähtää parempaan asiakaslähtöisyyteen. Julkisen hallinnon eri tahot ovat nimittäin tässä mielessä varmasti eri tasolla.

### 3.4.3 Työnkulkuihin liittyviä projekteja

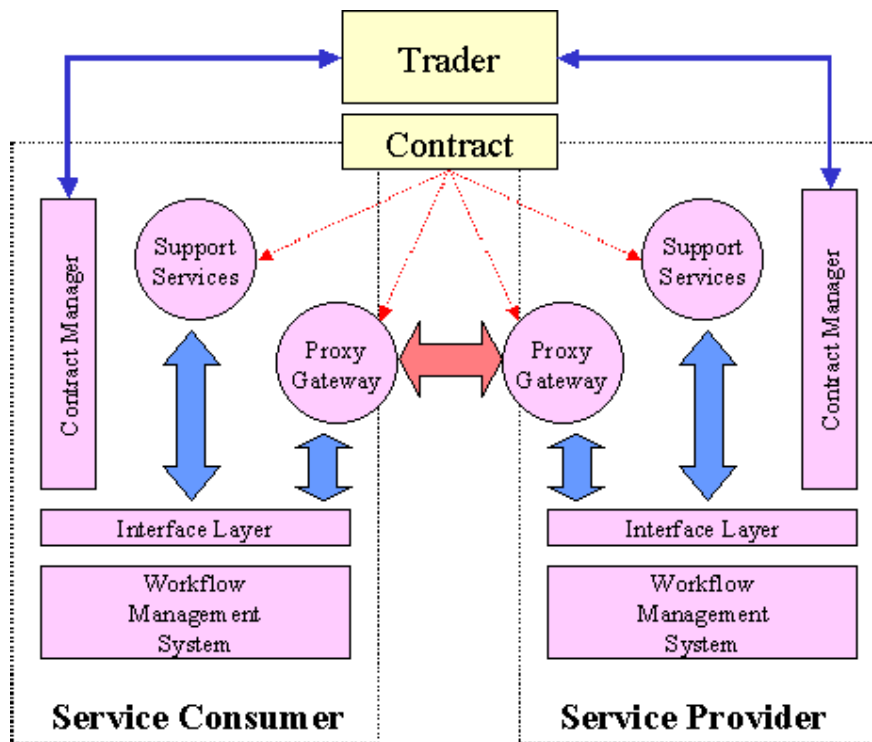
CrossFlow-projekti perustuu aikaisemminkin esille tulleeseen tuottaja-kuluttaja malliin. Projektissa on kehitetty työnkulkujen hallintajärjestelmää virtuaalisille organisaatioille. Projektin liittyvässä artikkelissa [GAH00] ei puhuta kasvuympäristöstä, mutta mallissa on samantapaisia piirteitä kuin edellä mainitussa VBE:ssä. CrossFlow:n virtuaalisessa organisaatiossa osapuolet löytävät toisensa dynaamisesti, ja kuluttaja voi ”ulkoistaa” osia prosessistaan tuottajalle (dynamic service outsourcing). Ennen kuin suhde kahden osapuolen välillä syntyy, tehdään CrossFlow-sopimus jossa määritellään palvelu (contract-based service specification). Sopimuksen yksityiskohdat riippuvat täysin osapuolista, eikä sopimusneuvotteluiden aloittaminen tarkoita välttämättä, että sopimus syntyy. Sopimuksen teke-

minen mahdollistaa monipuolisemman integroinnin ja vuorovaikutuksen kuin mitä tapahtuisi, jos toinen osapuoli tarjoaisi vain kiinteän rajapinnan palveluun. Lisäksi vuorovaikutusta parantavat CSS-palvelut (cooperation support services). Näiden palveluiden avulla työkulkujen valvonta ja hallinta on mahdollista. Kuvassa 9 on esimerkki eräästä työkulusta CrossFlow-järjestelmässä.



Kuva 9. CrossFlow [GAH00].

CrossFlow:n tekninen arkkitehtuuri [GAH00] perustuu työkulkujen hallinnan osalta IBM:n MQSeries Workflow- tuotteeseen, joka on erotettu rajapintakerroksella muusta CrossFlow-järjestelmästä. Kun palvelun tarjoaja haluaa julkaista palvelun, se ottaa yhteyttä *sopimuksen hallintaohjelmaan* (contract manager), joka ottaa yhteyttä CORBA-pohjaiseen *välittäjään* (trader). Palvelun kuluttaja toimii samalla tavalla etsiessään palvelua. Kun vaatimukset täyttävä palvelu löytyy, syntyy *sähköinen sopimus* (contract). Sopimuksen sisällön perusteella muodostuu sopimusympäristö, jossa välityspalvelinyhdyskäytävät (proxy gateway) hallinnoivat kommunikointia, ja *tukipalvelut* (support services) mahdollistavat vuorovaikutuksen. Kun sopimus on täytetty, sopimusympäristö taetaan. Kuvassa 10 on edellä kuvattu tekninen arkkitehtuuri.



Kuva 10. CrossFlow, tekninen arkkitehtuuri [GAH00].

CrossFlow perustuu samanlaiseen dynaamisuuteen kuin VBE-ajatuskin. CrossFlow'n paras idea, dynaamiset sopimukset, ei välttämättä tuo lisäarvoa tarkoin säädellyille julkisen hallinnon prosesseille. Käyttökelpoisuus julkisen hallinnon tiukoissa rakenteissa ei ole siis selvää. Toisaalta dynaamisemmat mallit ovat tervetulleita julkiseen hallintoon, ja tätä asiaa onkin jo käsitelty luvussa 2.

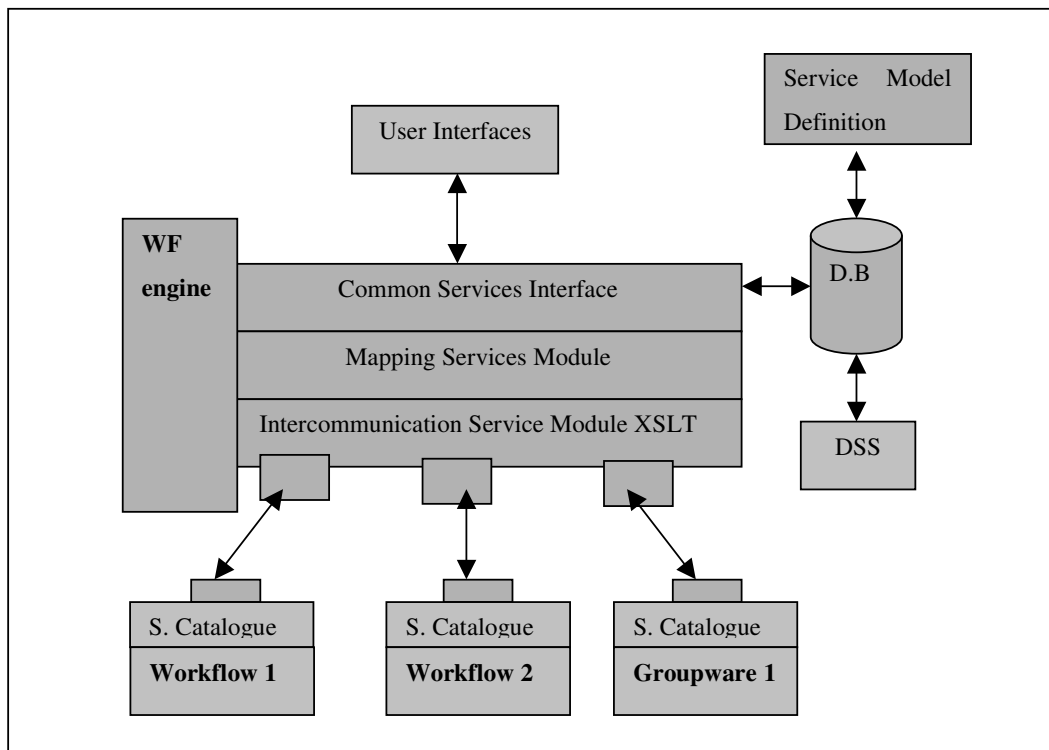
IMPULSE (IMproving PUBlic SERVICES) –projektissa [PDE02] sen sijaan on toteutettu nimenomaan julkiseen hallintoon tarkoitettu työkalu työnkulkujen hallintaan virtuaalisissa organisaatioissa. Myös IMPULSE'n integrointitekniologia nojaa CORBAan ja XML-kieleen. Ulkoiset sovelluksien ja järjestelmien kanssa kommunikoidaan käyttäen FORO-arkkitehtuuriin perustuvia sovellusajureita, joita kutsutaan liittimiksi (connector). Seuraavat metodikutsut (kuva 11) kuvaavat yksinkertaisesti liittimen toimintaa.

```
get_required_informationDef();  
set_input_information(in ParameterList params);  
need_moreinfo();  
get_returned_informationDef();  
get_output_information(inout params: ParameterList);  
release();
```

Kuva 11. Liittimen metodikutsut IMPULSEssa [PDE02].

Liittimet toimivat myös liikkuvina agenteina jos esimerkiksi loppukäyttäjä haluaa tietää prosessin käsittelyn tilan. Tällöin liitin käy kysymässä jokaiselta prosessiin liittyvältä taholta prosessin tilasta. Tässäkin projektissa siis kokeillaan orastavaa agenttitekniologiaa.

IMPULSE on suunniteltu erityisesti prosesseihin, jotka koostuvat useista erillisistä pienistä prosesseista virastojen omissa työnkulku (WF)- tai työryhmäohjelmistoissa (GW) [PDE02]. IMPULSE ei sisällä palveluhakemistoa, johon palveluliittomalli perustui. Oletuksena on, että palveluhakemisto on olemassa jossain muodossa järjestelmän ulkopuolella. Mallin ydin on keskitin, joka koordinoi ja valvoo loppukäyttäjien portaalien ja erillisten työnkulku- ja työryhmäohjelmistojen välissä. Edellä kuvaillun IMPULSE-mallin graafinen esitys on kuvassa 12.



Kuva 12. IMPULSE-malli [PDE02].

Tyypillinen tapahtumaketju on seuraava [PDE02]:

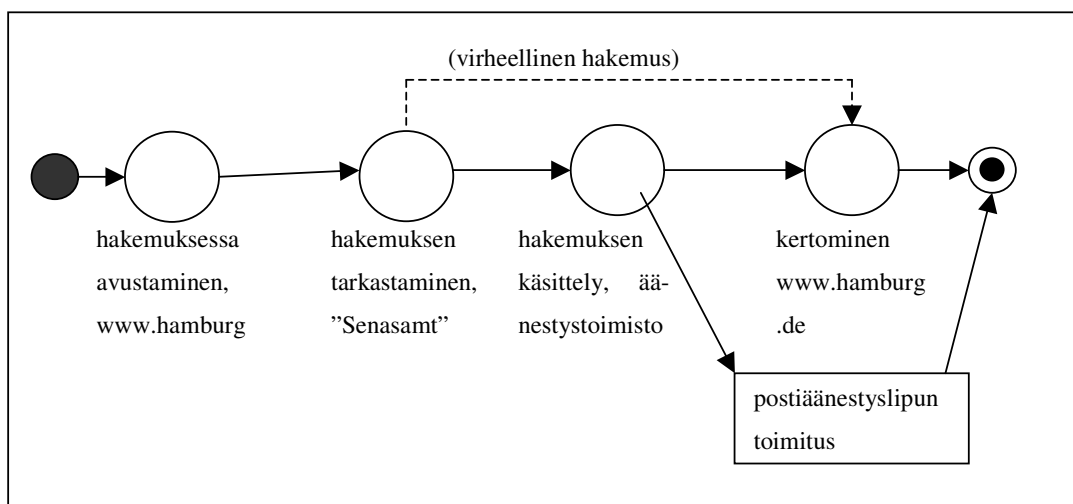
1. Kun ulkoista palvelua tarvitaan, intercommunication-kerros herätetään.
2. Intercommunication-kerros lähettää pyynnön kyseisen palvelun liittimelle.
3. Liitin luo työnkulkuilmentymän tai suorittaa toiminnon (GW).
4. Vastaava palvelin suorittaa halutun palvelun.
5. Liitin saa tulokset ja lähettää ne Intercommunication-kerrokselle.
6. Tulokset lisätään vastaavan IMPULSE-työnkulun jonoon ja lähetetään pyytäjän liittimelle.

Tiedon muunnokset tehdään, kuten muissakin esitellyissä projekteissa, XML:n avulla. Tästä huolimatta artikkelissa [PDE02] mainitaan yhdeksi ratkaisemattomaksi ongelmaksi vaihtelevat termistöt ja kunnollisten ontologioiden puutteen. Myös tuki palveluntarjoajien vertailuun ja palvelun laadun arviointiin IMPULSEsta puuttuu (vrt. VBE).

Hampurin kaupungissa tehdyssä sähköisen hallinnon projektissa [KIW02] sen sijaan näkökulma on aivan toisenlainen. Taustalla on sama luottamuksen saamisen idea kuin kasvuympäristöjen yhteydessäkin, ja palvelu eli kansalaisten tarpeiden tyydyttäminen. Mo-

nimutkaisissa transaktioissa kansalainen on tottunut siihen, että joku ihminen vastaa hänen asiastaan. Tekijöiden mukaan sähköisten palveluiden kohdalla näin ei enää ole. Projektissa on myös todettu, ettei dynaaminen tuottaja-kuluttaja ympäristö sovi julkiseen hallintoon. Tämä liittyy aiemminkin käsiteltyyn julkisen hallinnon staattisuuteen.

Projektissa on kehitetty konsepti nimeltään serviceflow management (SFM), eli palvelunkulkujen hallinta. Projektissa toteutettiin postiäänestysihakemus kaupungin portaalin kautta. Palvelunkulku koostuu sarjasta palvelupisteistä, joissa jokaisessa tapahtuu jokin palveluun liittyvä osa. Palvelupisteet voivat olla useissa eri organisaatioissa. Esimerkki palvelunkulusta on kuvassa 13.



Kuva 13. Palvelunkulku postiäänestysihakemuksessa portaalissa [www.hamburg.de](http://www.hamburg.de) [KIW02].

Projektissa tehtiin poikkeava toteamus verrattuna moneen muuhun projektiin. Mitään keskitin, kuten työkulkumoottori tai palvelin ei tulisi kyseeseen liian erilaisten järjestelmien takia [KIW02]. Tämä on yllättävä toteamus, koska juuri kyseiseen ongelmaan uudet integrointiteknologiat pyrkivät vastaamaan. Kyse saattaa tosin Hampurin tapauksessa olla myös rahasta, integrointi ei ole halpaa.

Hampurin projektissa tiedon siirtäminen on toteutettu omalla XML-määrittelyllä [KIW02]. XML-dokumentit siirtyvät organisaatioiden välillä ja tällä saadaan aikaan haluttua integrointia. Ainoa vaatimus organisaatioille on siis se, että ne kykenevät käsittelemään ja luomaan tietyn DTD:n mukaisia dokumentteja. Projektin yhteydessä on kehitetty myös kerrosmalli palvelunkulkujen prosessointiin, mutta myös oman arkkitehtuurin käyttö on sallittua, kunhan se tukee vaadittuja dokumentteja. Tämä tuo joustavuutta palvelunkulkuihin.

Myöskään mitään muita palveluja, kuten sopimusten tekoa tai palvelun laatua, ei suoraan tueta. Kolmena pääperiaatteena ovat 1. läpinäkyvyys kansalaisille, 2. joustavuus hallinnon henkilökunnalle ja 3. synergia ulkoisten palveluntarjoajien kanssa [KIW02].

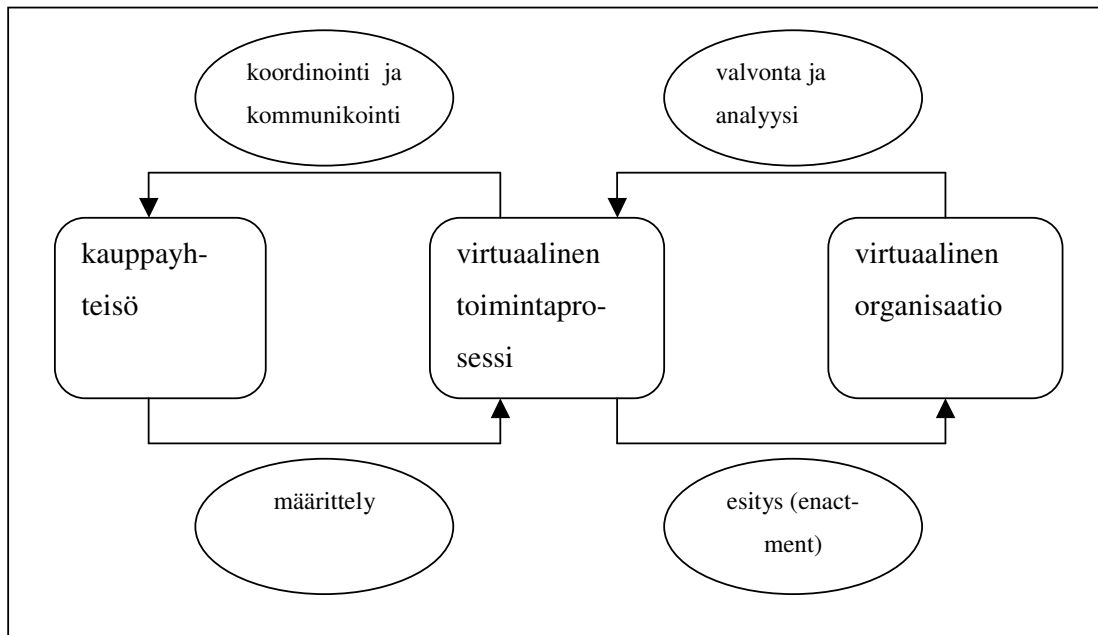
Hampurin projekti on poikkeuksellinen siinä mielessä, että lähtökohtana on kansalaisen hyvä palveleminen. Projektissa ei myöskään ole rakennettu massiivisia IT-infrastruktuureja, vaan on päädytty yksinkertaiseen ja joustavaan XML-dokumenttien siirtoon.

WISE (Workflow based Internet SErvices) on kiinnostava työnkulkuihin ja virtuaalisiin organisaatioihin liittyvä projekti. WISE-malli perustuu abstraktioon, jonka osat ovat *virtuaaliset toimintaprosessit* (virtual business processes), *virtuaaliset organisaatiot* (virtual enterprises) ja *kauppayhteisöt* (trading communities) [LAS00]. Termistöstä huomaa, että malli on suunniteltu sähköiseen kaupankäyntiin. Mallissa ei ole kuitenkaan mitään sellaista, miksi se ei soveltuisi julkisen hallinnon palveluihin.

Virtuaalinen toimintaprosessi kuvaa kokonaisen prosessin jonkin tavoitteen saavuttamiseksi. Se ei ole sidottu organisaatioiden rajoihin [LAS00]. Tyypillisesti eri virastot tuottavat osia virtuaaliseen prosessiin. Näiden aliprosessien yksityiskohtia ei tarvitse tietää, niihin suhtaudutaan kuin kapselointiin olio-ohjelmoinnissa. Vain aliprosessin rajapinta on tiedettävä. Tällainen kapselointiajatus selkeyttää prosessin rakennetta, mutta vähentää dynaamisuutta jota joissain projekteissa on korostettu.

Jotta virtuaalinen toimintaprosessi voidaan kuvata, tarvitaan sille konteksti. Konteksti kuvaa tavoitteet, säännöt, vaatimukset, rajoitteet ja resurssit [LAS00]. Tässä kontekstia kutsutaan virtuaaliseksi organisaatioksi. Kolmantena osana mallissa kuvataan toimijat, joita kutsutaan kauppayhteisöksi. Kauppayhteisöt ovat tahoja jotka ovat osallisena virtuaalisessa yrityksessä.

Mallista on kehitetty ohjelmistomalli, jossa on neljä moduulia: prosessin määrittely, prosessin esitys (enactment), prosessin valvonta ja analyysi, sekä koordinointi ja kommunikointi (Kuva 14) [LAS00].



Kuva 14. WISE-järjestelmän määrittely.

Prosessin määrittelyssä eri osapuolet julkaisevat palvelunsa www-luettelona. WISEssä tämä tehdään käyttämällä Java-tekniikoita [LAS00]. Kokonainen prosessi on mahdollista tehdä erityisen käyttöliittymän kautta drag and drop -tyyliin käyttämällä luetteloita. Eri kauppayhteisöt voivat määrittellä prosessinsa omalla kielellä, jonka WISE-järjestelmä kääntää yhteiselle OCR (Opera Canonical Representation)-kielelle. OCR-kuvaus toimii lopullisena prosessin kuvauksena.

Prosessin esitys ja ajo tapahtuu WISE-moottorissa. Moottori myös tallentaa tietoja prosesseista valvontakäyttöön ja mahdollisten vikatilanteiden varalta. Moottorin ominaisuuksissa on erityistä huomiota kiinnitetty turvallisuuteen, palvelun laatuun ja vikasietoisuuteen [LAS00].

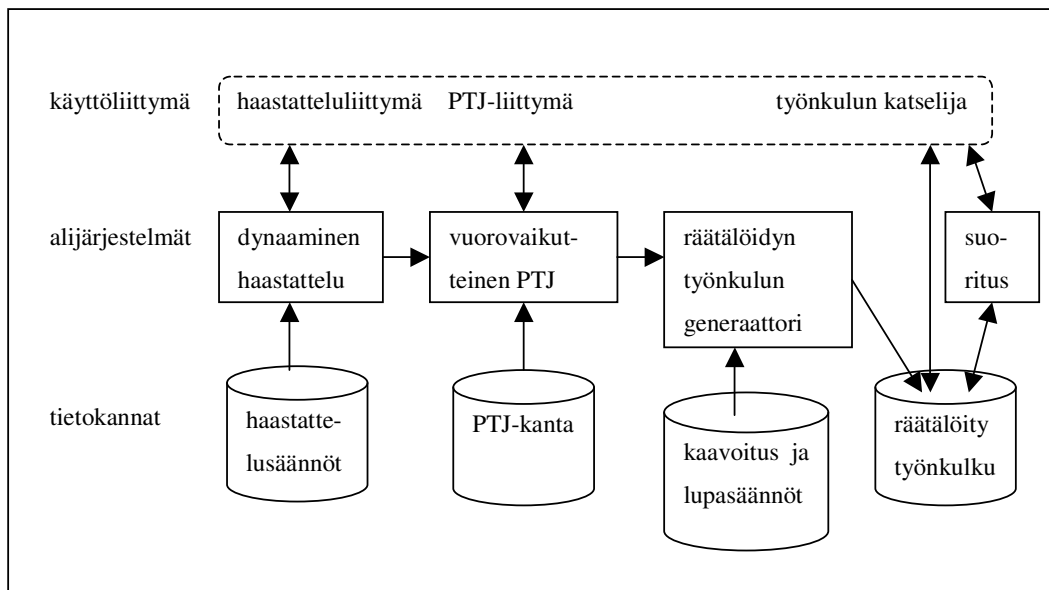
WISE-järjestelmä tallentaa historiatietoja prosesseista mallintamistarkoituksiin [LAS00]. Historiatietojen avulla parannetaan järjestelmän ajonaikaista päätöksentekoa, sekä voidaan analysoida historiatietoja. Käyttäjien on mahdollista valvoa prosesseja erityisten valvontatyökalujen avulla. Yhteistyöhön ja kommunikointiin ei ole kehitetty omaa ohjelmistoa, vaan järjestelmässä käytetään jo olemassa olevaa CoBrow (Collaboration Browsing in Information Resources)-ohjelmistoa [LAS00].



WISEn toimintoihin kuuluu myös poikkeusten hallinta, aliprosessien kanssa kommunikointi, palvelun laatuun (QoS) liittyviä toimintoja ja transaktio-ominaisuuksia [LAS00]. Transaktioiden osalta ei noudateta tiukkoja ACID-ominaisuuksia. WISEssä käytetään erilaisia alueita atomisuudesta, oikeellisuudesta ja eristyvyydestä. Prosessi voidaan jakaa eri alueisiin käyttäjän toimesta. Lisäksi käytetään TCA (Transactional Coordination Agents)-kääreitä vikasietoisuuden parantamiseen transaktioiden yhteydessä [LAS00].

WISEn näkökulma on prosessilähtöinen, ja tarjoaa varteenotettavan vaihtoehdon ainakin suunnittelun alkuvaiheeseen. Abstraktiot, kuten virtuaalinen yritys, helpottavat kokonaisuuden hahmotusta. Myös toiminnallisuutta on suunniteltu, mutta WISEn suurin arvo on prosessien ja työnkulkujen määrittelypuolella.

Julkinen päätöksenteko voi olla hyvin alueellista, tyypillisimpänä esimerkkinä on kuntien maankäyttö. Yhteispalvelussa työnkuluja jotka liittyvät maankäyttöön, tulisi valtava määrä, koska säännökset vaihtelevat valtavasti yhden kunnankin sisällä. Eräs kehitysryhmä [HCA01] on rakentanut julkisen hallinnon pilottijärjestelmän juuri tätä taustaa silmäläpitiäen. Järjestelmään on integroitu paikkatietojärjestelmä, jonka avulla esimerkiksi maankäyttöhakemuksen tekeminen helpottuu. Järjestelmä mahdollistaa myös räätälöinnin käyttäjän perusteella. Työnkulut eivät ole staattisia, vaan muodostuvat sijainnin, paikallisten säännösten, ja kunkin käyttötapauksen mukaan [HCA01]. Järjestelmän arkkitehtuuri näkyy kuvassa 15.



Kuva 15. Työnkulku-paikkatietojärjestelmän arkkitehtuuri [HCA01].

Verkossa toimiva sovellus haastattelee ensin käyttäjää, ja tämän perusteella ottaa yhteyttä paikkatietojärjestelmään. Tämän jälkeen paikkatietojärjestelmä antaa graafista tietoa halutuista alueista, ja käyttäjä voi vielä muokata hakuetojaan. Haastattelun ja paikkatietojen perusteella luodaan uniikki työnkulku, joka tallennetaan käyttäjän tunnuksella. Työnkulun katselijalla käyttäjä voi vertailla erilaisia luomiaan työnkuluja ja valita suoritettava työnkulku [HCA01]. Prototyypin web-käyttöliittymässä on kolme kehystä, yksi kullekin käyttöliittymätoiminnolle.

Järjestelmä tarjoaa ratkaisua yhteen todelliseen ongelmaan, ja paikkatietojärjestelmän käyttö on selvästi hyödyllistä. Pilotti kuitenkin ratkaisee vain pienen osan yhteispalveluiden ongelmasta, eikä siinä ole kovin kehittyneitä toimintoja esimerkiksi poikkeustilanteille. Integroituna johonkin kokonaisvaltaisempaan järjestelmään, jossa paikkatietoja tarvitaan, se toisi selvää lisäarvoa.

## 4 Kohti sähköisiä yhteispalveluita

Yhteispalveluiden kehittäminen on aktiivisen keskustelun kohteena oleva aihe julkisen hallinnon piirissä ympäri maailman. Keskusteluista on kuitenkin pitkä matka toimiviin järjestelmiin. Luvussa 4.1 esitellään teknisiä prototyyppejä ja pilottiprojekteja, jotka tähtäävät todellisiin yhteispalveluihin. Kenttä on nuori ja suppea, joten raportoituja pilotteja on vähän. Valintaa esiteltävien pilottien välillä ei ole siis päästy tekemään. Luvussa 4.2 kootaan kokemuksia projekteista, jotka eivät suoranaisesti pyri yhteispalveluihin, vaan ovat pienimuotoisempia askeleita kohti integroidumpia järjestelmiä. Luvun sisältö on vähemmän teknistä. Viimeisessä alaluvussa (4.2) käsitellään tietoturvaa ja tietosuojaa.

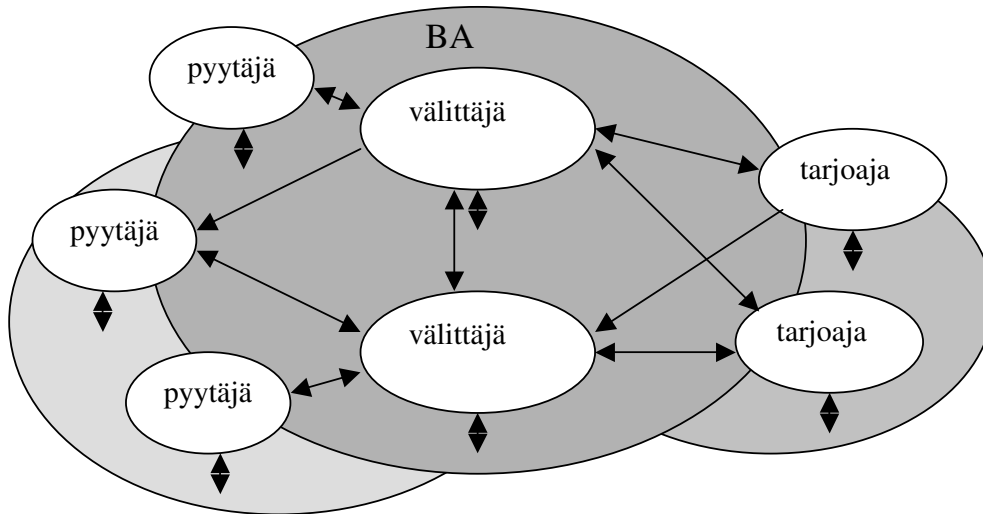
### 4.1 Hankkeissa käytettyjä teknisiä ratkaisuja

Tähän lukuun on koottu sellaisia projekteja, joissa on kehitetty teknistä arkkitehtuuria tarpeeksi pitkälle. Tällä tarkoitetaan riittävää teknistä konkreettisuutta. Kovin abstraktilla tai yleisellä (esimerkiksi [Gla02]) tasolla olevia malleja ei ole otettu tarkempaan tarkasteluun. Kentän kehittymättömyys näkyy projektien vähyytenä, ja tämän vuoksi seuraavassa esiteltävien projektien teknisen kuvauksen tarkkuus vaihtelee.

#### 4.1.1 ADSS-OSCAR

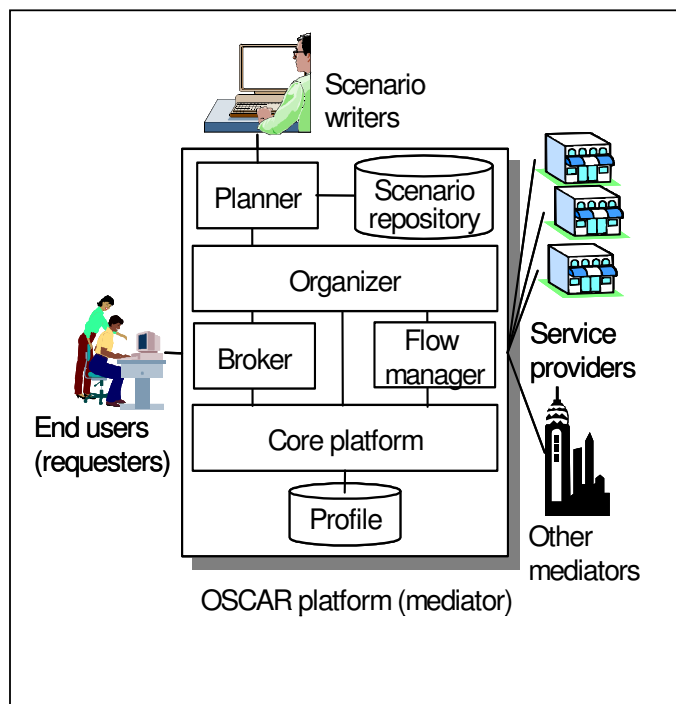
Japanilaisessa Information HUB-projektissa on ehdotettu käytettäväksi ADSS-mallia ja OSCAR-alustaa. ADSS (Autonomous Decentralized Service System) on Hitachin ja saksalaisen GMD/Focus instituutin kehittämä väliohjelmistoalusta, joka pohjautuu TINA-malliin ja CORBAan [Hit00, FST00]. Taustalla on VBE:n kaltainen dynaaminen malli. Myös yhtäläisyyksiä CrossFlow- projektiin löytyy, lähinnä sopimisen suhteen. ADSS-mallissa [TFS01] pääpaino on välittäjissä (mediator). Muut osapuolet ovat tarjoaja (provider) ja pyytäjä (requester). Mallissa on jostain syystä poikettu tuottaja-kuluttaja nimeämistavasta. Lisäksi mallissa on välityspaikka, joka on nimetty Baksi (japaniksi paikka, ”Ba”). Bassa välittäjät tarjoavat välityspalveluita kaikille kolmelle osapuolelle. Tarjoajat ja pyytäjät rekisteröivät palvelunsa välittäjille, jotka saattavat rekisteröintitietojen perusteella tietyt osapuolet yhteen. Tiedon välityksessä käytetään SOAP-sanomia, ja palvelutarjoukset ovat hyvinmuodostettuja XML-dokumentteja. Palvelutarjoukset koostuvat kolmesta

osasta: järjestelmäosasta (system part), käyttäjäosasta (user part) ja tiedusteluosasta (query part). ADSS-malli näkyy kuvassa 16.



Kuva 16. ADSS – malli [TFS01].

Välittäjiä varten on toteutettu OSCAR (Open Service Collaboration pLAtfoRm) alusta, jota kuvaa kuva 17.



Kuva 17. OSCAR välittäjäalusta [TFS01].

*Planner* muuntaa jonkun määrittelemät palveluskenaariot suunnitelmiksi, jotka voivat koostua useista palveluista, tarjoajista ja pyytäjistä. *Broker* etsii näistä suunnitelmista tilanteeseen sopivat. *Flow manager* tulkitsee suunnitelmat ja toimii työkulujen hallintajärjestelmänä. *Core platform* hallinnoi käyttäjien (tarjoajat ja pyytäjät) profiileja ja tarkkailee palveluiden käyttöä. Web-palvelin, sekä muiden laitteiden tuki, (esimerkiksi kannettavat laitteet) sisältyvät ydinalustaan. *Organizer* hallinnoi kaikkia toimintoja ja virhetilanteita.

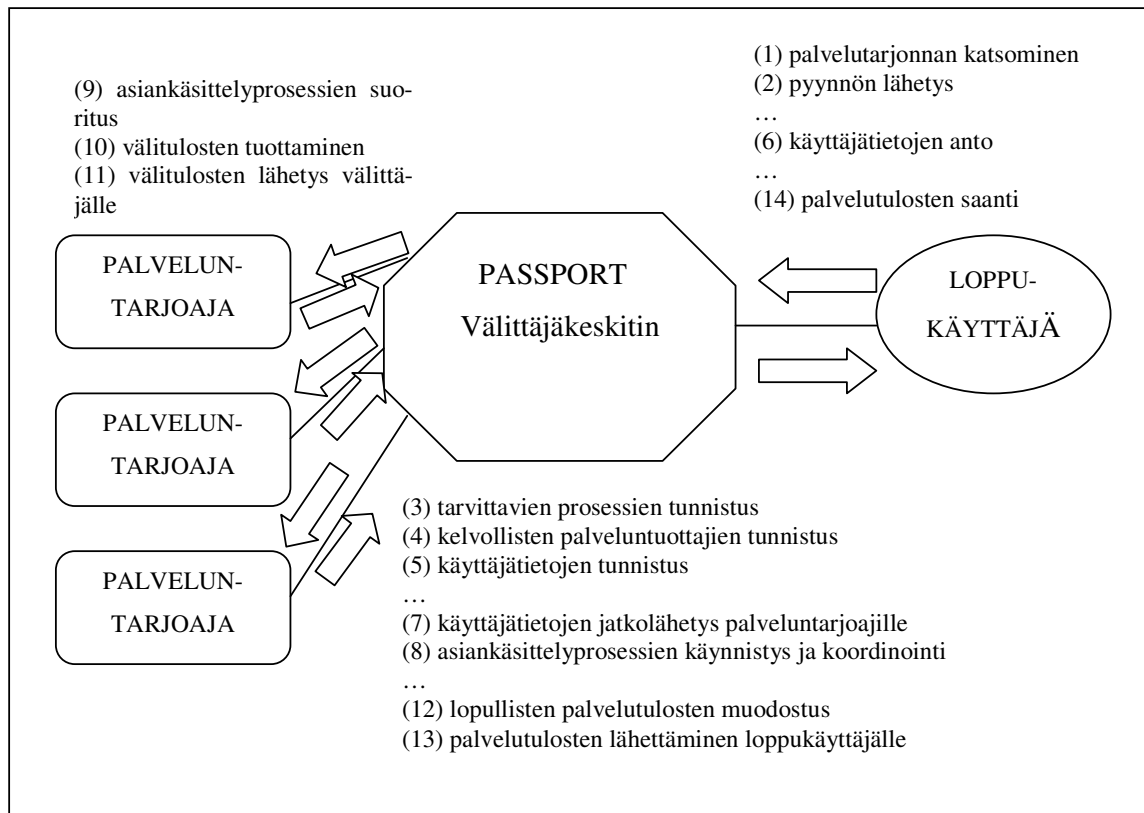
Sijoitettuna viitemalliin (luku 2.5), *planner*, *broker* ja *flow manager* luovat, toimeenpanevat ja valvovat työkulukuja. Kuitenkin vain *flow manager* sijoittuu ylimpään koordinoitinkerrokseen, ja *planner* sekä *broker* sijoittuvat palvelun viitemalliin. Lisäksi erilaisille skenaarioille on oma varastonsa (*scenario repository*). Tämä ei ole sama kuin palvelutarjousvarasto. Palvelun kutsumalliin ei tässä järjestelmässä toistaiseksi oteta kantaa muuten, kuin että sanomat ovat SOAP-muotoisia. *Core platform* ja *organizer* ovat lähinnä käyttäjien hallintaan ja järjestelmän yleiseen valvontaan liittyviä osia.

ADSS-OSCAR prototyyppi tarjoaa dynaamisuutta samassa mielessä kuin VBE (virtual breeding environment), ja työkulujen hallintaa. Sitä ei ole kuitenkaan testattu todellisessa ympäristössä ja kehitystyö jatkuu. Vaikka OSCARin yksityiskohtaisuus ylittää moduulitasolle, koko projektin kuvaus jättää vaikutelman, jonka mukaan useat yhteispalveluiden kannalta olennaiset asiat puuttuvat.

#### **4.1.2 PASSPORT/CB-BUSINESS**

CB-BUSINESS projekti (1.4.2002-31.3.2004) on tällä hetkellä pisimmälle vietyjä todellisen yhteispalvelun projekteja [Cro02a]. Projektissa on toteutettu PASSPORT-nimistä arkkitehtuuria lähinnä G2B-tarkoituksiin EU:n sisällä. Arkkitehtuurin on tarkoitus toimia maiden rajat ylittävän tuntumattoman palvelun pohjana [GMG01]. Arkkitehtuurin ytimenä on välittäjäkeskitin (intermediation hub) ja lähtökohtaisesti topologia on tähdenmuotoinen [GMG01]. Loppukäyttäjä asioi vain keskittimen kanssa, vaikka työnkulku vaatisikin usean palveluntarjoajan toimintaa. Tuntumattomuus toteutuu siis siten, kuin todellisissa yhteispalveluissa täytyykin. Myös oman palvelupyynnön tilaa on mahdollista tarkkailla.

PASSPORTissa yksittäiset palveluntarjoajat määrittelevät palvelunsa, tarvittavan laatutason, tiedonkulut ja kommunikointikanavan. Näiden määrittelyjen perusteella välittäjäkeskus käsittelee loppukäyttäjiltä tulevat pyynnöt [GMG01]. Yhteistä kommunikointitapaa tai rajapintaa ei siis tarvitse olla. Tässä mielessä arkkitehtuuri poikkeaa monesta muusta esitellystä. Palveluntarjoajan sisäisiin työkulkuihin ei puututa, ne voivat olla jopa paperipohjaisia [GMG01]. Kunhan kommunikaatio välittäjäkeskittimeen toimii. Kuvaus arkkitehtuurista ja tapahtuman kulusta on kuvassa 18.



Kuva 18. PASSPORT välittäjäarkkitehtuuri [GMG01].

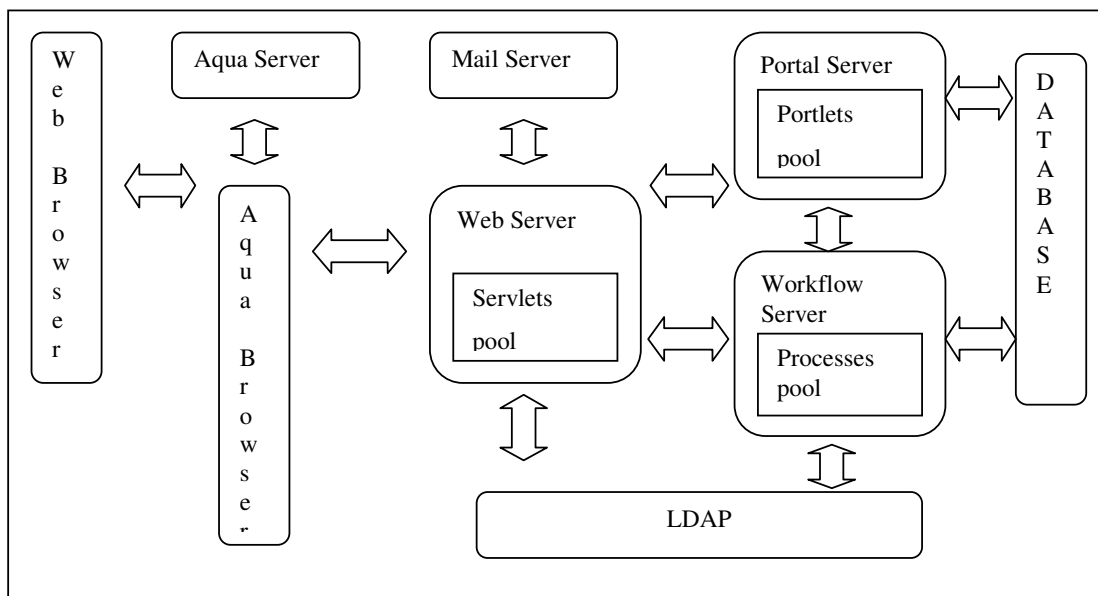
Myös PASSPORTissa palveluntarjoajat voivat julkaista palveluitaan välittäjälle (keskitin), jolloin ne ovat käytettävissä.

Koska tähdenmuotoinen topologia saattaa muodostua pullonkaulaksi, järjestelmässä on mahdollista käyttää myös liittotopologiaa. PASSPORTissa tässäkin olisi kuitenkin keskus, joka valvoisi globaalisti toimintaa [GMG01]. Jos kuormantasausta ei ole, tällaiselle yhdelle keskukselle asetettavat suorituskykyvaatimukset ovat isossa järjestelmässä valtavat.

PASSPORTin toimintalogiikka ja kommunikointi on erotettu toisistaan. Toisin kuin esimerkiksi web-palveluissa, kommunikointikanavaa ei ole standardoitu. Kommunikointi on toteutettu erillisillä kommunikaatiomoduuleilla, jotka peittävät kommunikoinnin yksityiskohdat [GMG01].

Välittäjäkeskus koostuu neljästä kerroksesta, jotka ovat: istuntokerros, työnkulkukerros, palvelukerros ja kommunikointikerros [VGL03].

Komponenttitasolla arkkitehtuurin tärkeimpiä osia ovat työnkulkujenhallintamoottori ja web-käyttöliittymä. Käyttöliittymän kautta loppukäyttäjät voivat tehdä palvelupyynnöjä ja etsiä tietoa, mutta myös palveluntarjoajat voivat valvoa omaa tilannettaan [VGL03]. Palveluntarjoajat voivat myös julkaista palveluitaan palveluvarastoon käytettäväksi. Käyttäjien ja palveluiden hallinnoimiseen käytetään LDAP (Lightweight Directory Access Protocol)-komponenttia [VGL03]. Kuvassa 19 on graafinen kuvaus järjestelmän komponenteista.



Kuva 19. CB-BUSINESS arkkitehtuuri komponenttitasolla [VGL03].

CB-BUSINESS projektissa on käytetty jo aiemmin esille tulleita konsepteja, kuten palveluiden julkaiseminen, työnkulut ja portletit. Järjestelmä myös peittää alla olevan heterogeenisyyden, mutta ei kuitenkaan välttämättä vaadi palveluntarjoajilta muutoksia omiin työnkuluihin. Järjestelmästä on olemassa kokeiluversio, jonka pilottitestausta on käynnissä.

Testaus on kuitenkin mahdollista vain CB-BUSINESS organisaation jäsenille (toukokuu 2004).

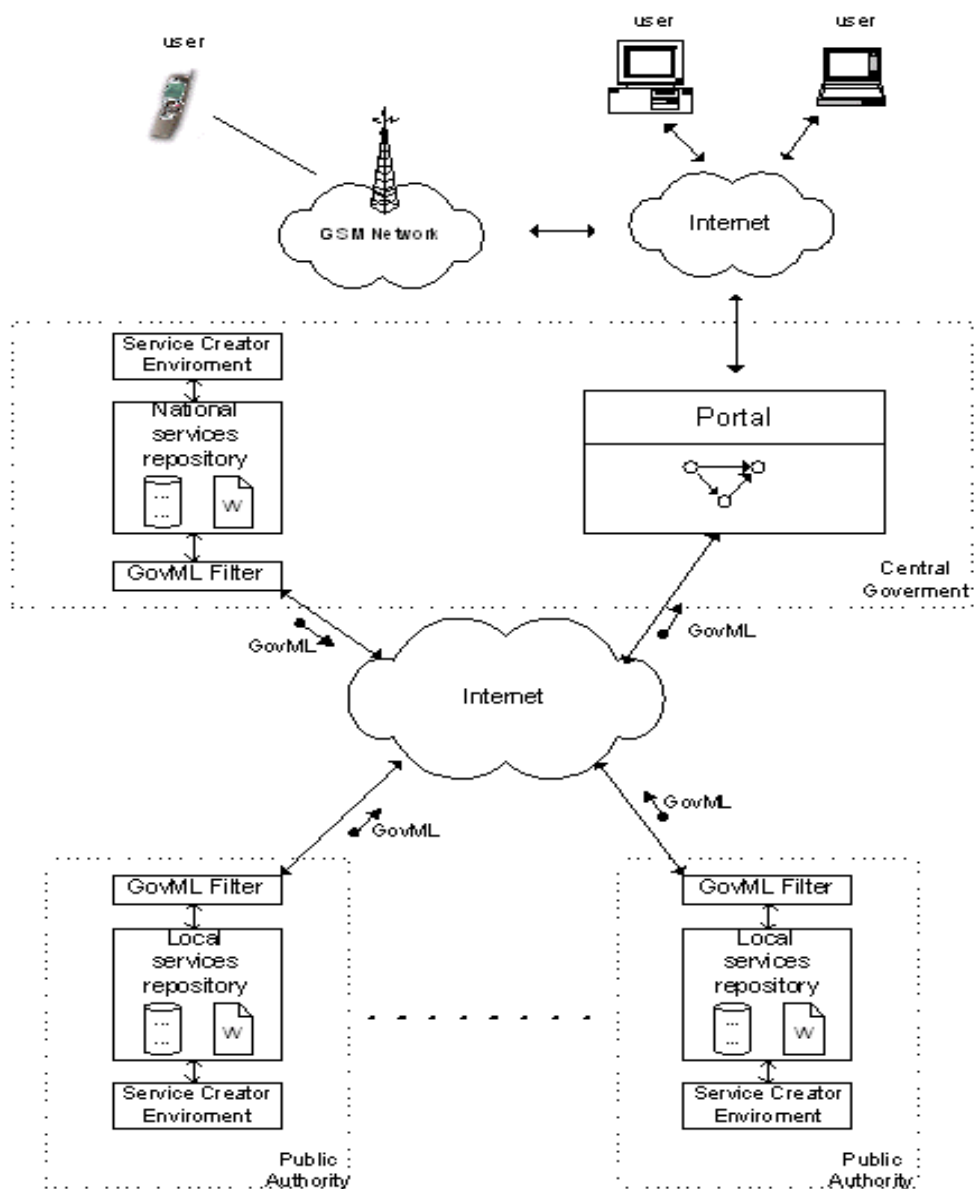
Koska luvussa 2.5 kuvattu viitemallia on kehitetty CB-BUSINESS projektin yhteydessä, noudattelee arkkitehtuuri kyseisessä viitemallissa kuvattua abstraktia mallia. Välittäjäkeskuksen neljän kerroksen toiminta noudattelee OSP-kerrosmallia. Kahden ylimmän kerroksen järjestys on tosin vaihdettu [VGL03], eikä kaikkiin mallissa kuvattuihin kohtiin löydy ratkaisua. Tämä kuvaakin hyvin jo aiemmin esille tuotua ongelmaa kerrosten järjestyksestä, eli ei ole aina kovin selvää missä järjestyksessä kerrosmallien kerrokset tulisi esittää.

### 4.1.3 eGOV

EU:n rahoittamassa eGOV-projektissa (2001-2003) on kehitetty integroitua yhteispalvelualustaa. Suomesta mukana on ollut TietoEnator. Projektin päätavoitteena ovat olleet seuraavan sukupolven julkisen hallinnon portaalien ja verkkoarkkitehtuurin kehittäminen, palveluvarastot ja palveluiden luontiympäristö, sekä GovML (Governmental Markup Language)-kieli [Tam01a]. Tuotettavan portaalien ominaisuuksiin kuuluu muun muassa räätälöinti, kielituki, käytettävyys erilaisista laitteista ja digitaalinen allekirjoitus [Tam01a]. Palvelut kuvataan elämäntapahtumina (life events). Elämäntapahtumat on määritelty seuraavasti [KaT03]: *”life events describe situations of human beings that trigger public services”*. Elämäntapahtumilla tarkoitetaan samaa kuin aikaisemmin esillä olleilla elämäntilanteilla. Projektissa on käytetty myös huomattava määrä aikaa käyttäjävaatimusten ja elämäntapahtumien selvittämiseen, ja prosessien rakentamiseen näiden pohjalta. Erityisesti on kiinnitetty huomiota toiminnallisuuteen, tietoturvaan, luotettavuuteen, kielitukeen ja käytettävyyteen [TSK02, s. 21]. Selvitystä edellä mainittujen vaatimusten suhteen on tehty kyselytutkimuksissa ja haastatteluilla Itävallassa, Kreikassa ja Sveitsissä [TSK02, s. 22-27].

Alustassa (Kuva 20) loppukäyttäjänä ovat kansalaiset ja yritykset. Yksi keskusviranomaisen ylläpitää kansallista portaalaa ja –palveluvarastoa. Muita toimijoita ovat muut julkisen hallinnon tahot, esimerkiksi paikallishallinto. Näillä tahoilla on omat palveluvarastonsa.





Kuva 20. EGOV järjestelmä [Tam01a].

EGOVin tekninen arkkitehtuuri voidaan jakaa kahteen osaan, portaaliin ja väliohjelmistoon. Portaalin komponentit ovat palveluhakemisto (service directory), sisältöhakemisto (content directory) ja ympäristö sisällön metadatan luomiseen [Pap03, s. 11-12]. Palveluhakemisto hyväksyy vain WSDL-muotoisia kuvauksia [Erk02, s. 54]. Portaalin teknisenä alustana on Weblogic sovellus- ja portaalipalvelin, ja portaali on toteutettu portlettien avulla [Pap03, s. 13]. Väliohjelmisto koostuu palvelun ajonaikaisesta ympäristöstä (SRE), palvelutarjousvarastosta (SR) ja palvelun luontiympäristöstä (SCE) [Pap03, s. 12]. Myös väliohjelmistokerroksen sovelluspalvelimena on Weblogic, ja palveluhakemistona toimii

Sunin UDDI-hakemisto [Pap03, s. 13]. Toteutuksessa on käytetty J2EE:tä, web-palveluita, ja varsinkin RDF (*Resource Description Framework*) –standardia, jota käytetään metadatta-kuvauksiin [Erk02, s. 39-40].

Tiedon kuvauksessa käytetty GovML-kieli on XML-pohjainen. Taustalla on ajatus julkisten palveluiden ja elämäntapahtumien standardoinnista. GovML tietorakenne sisältää kolme sanastoa. Ensimmäinen sanasto kuvaa julkisia palveluita yleisellä tasolla, toinen kuvaa spesifisiä palveluita, ja kolmas kuvaa elämäntapahtumia tai liiketoimintatilanteita (business situation) [KaT03]. GovML-dokumenttien tietoturva- ja -suoja on tarkoitus toteuttaa vielä tekeillä olevilla W3C:n XML-spesifisillä suosituksilla [TSK02, s. 45]. Seuraavassa on esimerkki GovML dokumentista (kuva 21), joka kuvaa yhden elämäntapahtuman (naimisiin meno).

```
<?xml version="1.0" encoding="UTF-8"?>
<govml:GovML xmlns:govml="http://www.egov-project.org/GovML.Schema/"
xmlns:xsi="http://www.w3c.org/2001/XMLSchema-instance"
xsi:schemalocation="http://www.egov-
project.org/GovML.Schema/file:///C:/temp/GovML.Schema.xsd">
  <description xsi:type="govml:SpecificLifeEventDescription">
    <identifier>ABC1234H</identifier>
    <language>EN</language>
    <title>Description of the life event "getting married"</title>
    <description>Getting married</description>
    <attention>This life event concerns only adults</attention>
    <faq-list>
      <item>
        <question>Is there possibility to get married
          online?</question>
        <answer>Yes. Visit the national governmental portal </answer>
      </item>
    </faq-list>
    <related-services>
      <item>
        <title>Issuing a birth certificate</title>
        <uri>http://www.egovproject.org\birth# </uri>
      </item>
      <item>
        <title>Online payment</title>
        <uri>http://www.egovproject.org\online_payment#</uri>
      </item>
    </related-services>
    <law>Law with number FRC-234</law>
  </description>
</govml:GovML>
```

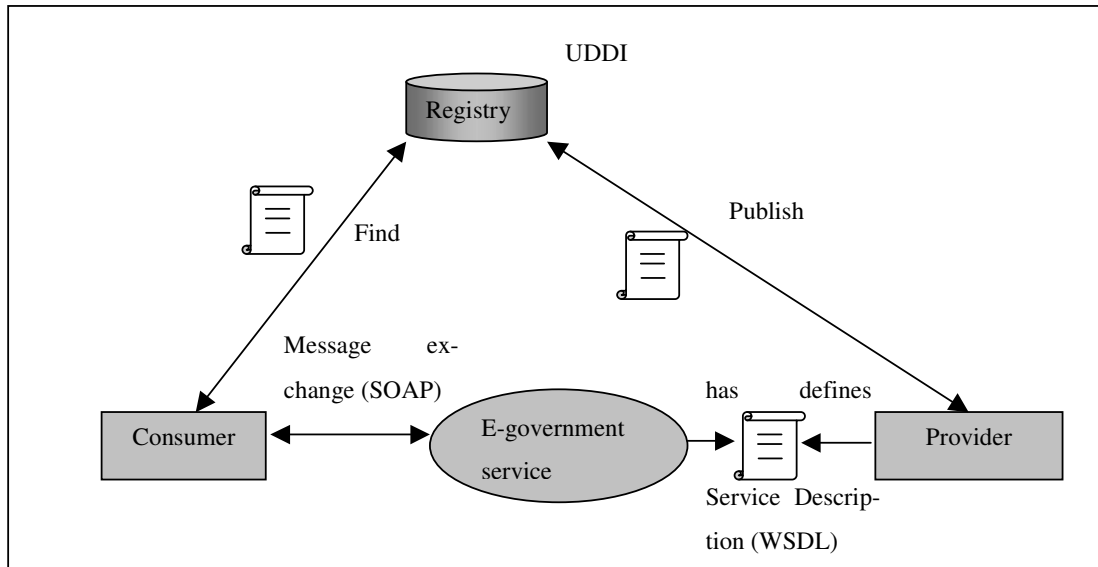
Kuva 21. Esimerkki GovML-dokumentista [KaT03].

EGOV-projektin ansiot ovat GovML:n kehittämisessä ja elämäntapahtuma-konseptissa. Erityisesti käyttäjävaatimuksiin on kiinnitetty aidosti huomiota. Myös tekninen arkkitehtuuri on riittävän yksityiskohtainen. Kun eGOV:ia verrataan OSP-malliin, nähdään että kolme alinta kerrosta ovat yhteneväisiä. Elämäntapahtumilla haetaan samaa kuin palveluiden nimeämisellä ja palveluontologioilla. Varsinkin palvelun kutsumalliin on kiinnitetty huomiota (GovML). Sen sijaan OSP-mallin ylin kerros, koordinointimalli, puuttuu. Työnkulkua ei aineistossa mainita. Prosesseista kyllä puhutaan, mutta hieman eri merkityksessä kuin työnkulkujen yhteydessä.

#### **4.1.4 WebDG**

WebDG (Web Digital Government) projektissa on kehitetty yhteispalveluarkkitehtuuria sosiaalipalveluiden tarpeisiin. Arkkitehtuurin suunnittelussa painopisteet ovat olleet palveluiden muodostamisessa (composing e-government services) ja yksityisyyden turvaamisessa [MRB03].

WebDG-arkkitehtuurissa on kolme erilaista toimijaa: tuottaja (provider), rekisteri (registry) ja kuluttaja (consumer) (Kuva 22). Tuottajat voivat olla julkisen hallinnon tahoja tai ulkopuolisia palveluntarjoajia. Kuluttajat ovat kansalaisia tai julkisen hallinnon virkailijoita. Tuottajat kuvaavat tarjoamansa palvelut käyttäen WSDL-kieltä. Kuvaukset on talletettu UDDI-rekisteriin ja kommunikointiin käytetään SOAP-sanomia. Kyseessä on siis puhdas web-palvelumalli.



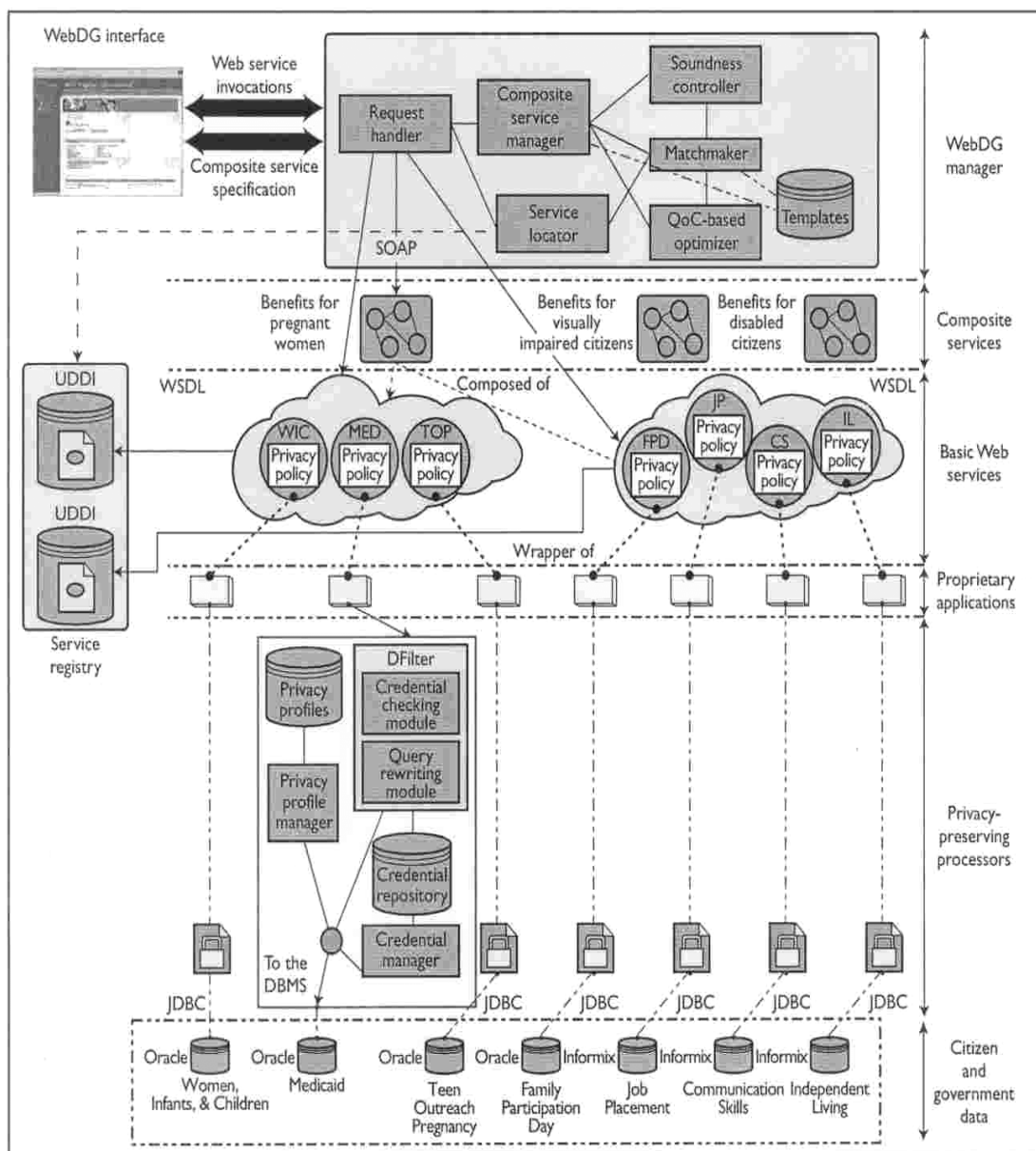
Kuva 22. WebDG palveluiden väliset suhteet [MRB03].

Palveluiden muodostamisessa on kyse siitä, että eri tahojen palvelut voidaan yhdistää yhdeksi kokonaisuudeksi. Palveluiden muodostamisessa tutkitaan syntaktisia ja semanttisia piirteitä. WSDL ei mahdollista palveluiden semanttista kuvaamista, joten kuvauksessa on käytettävä ontologioita. WebDG:ssä jokaisessa palveluoperaatiossa on kaksi ontologiaa, Category ja Type [MRB03]. Näiden avulla voidaan tiettyjä päättelysääntöjä käyttäen määrittää palveluiden keskinäinen muodostus. Lisäksi tutkitaan yhdistettävien palveluiden eheys (composition soundness) [MRB03]. Tässä yhteydessä eheydellä tarkoitetaan sitä, että yhdistetty palvelu tuottaa lisäarvoa verrattuna eriytettyihin palveluihin. Apuna käytetään muodostusmallia (composition template), joka kuvaa muodostussuunnitelman yleisen rakenteen, sekä tallennettuja malleja (stored templates) [MRB03]. Tallennetut mallit ovat valmiiksi tehtyjä ihannemalleja. Jos muodostumalli on tallennetun mallin osajoukko tai yhtä kuin tallennettu malli, on se eheä [MRB03]. Mallit kuvataan verkkoina.

Järjestelmässä on oletuksena, että tietoturva on jo kunnossa, ja täten voidaan keskittyä tietosuojan (privacy) turvaamiseen. Tässä ongelmia aiheuttaa kansalaistietojen jako eri virastojen kesken ja kansalaisten erilaiset vaatimukset tietosuojan suhteen. WebDG käyttää kolmitasoista tietosuojamallia, jossa tietosuoja voidaan jakaa käyttäjään, palveluun ja dataan [MRB03]. Käyttäjät voivat määrittellä henkilökohtaisen tietosuojaprofiilin, kullakin palvelulla on tietyt tietosuojasäännökset ja tietokannoista on saatavilla erilaisia näkymiä samoista dataolioista. Kaikki kyselyt ohjautuvat *tietosuojasuodattimen* (DFilter) läpi. Tie-

*tosuojasuodatin* koostuu kahdesta moduulista, jotka ovat nimeltään *CCM* (credential checking module) ja *QRM* (query rewriting module). Ensimmäinen tarkistaa valtuudet ja jälkimmäinen muokkaa tarvittaessa kyselyä sen mukaan. Lisäksi *PPM* (privacy profile manager) pitää huolta käyttäjien tietosuojaprofiilien toteutumisesta.

Kuvassa 23 näkyy WebDG-toteutus kokonaisuudessaan [MRB03]. Käyttöliittymä on toteutettu Java-servleteillä ja HTML-kielellä. WSDL-kuvaukset tuottaa Axis Java2WSDL-ohjelma ja kuvaukset julkaistaan UDDI-rekisterissä. Palvelunhallinta asiakkaana (service management client) on Apache SOAP 2.2, joka tarjoaa palvelimen pään infrastruktuurin ja sovellusliittymän asiakkaalle. Järjestelmän ytimessä on *WebDG manager*. *Service locator* etsii UDDI-rekisteristä WSDL-kuvauksia. Kun *request handler* havaitsee palvelun, se kutsuu palvelun operaatioita SOAP-tyngän (stub) kautta. Ennen tietokantakyselyjä pyynnöt menevät vielä tietosuojasuodattimen läpi. *Composite service manager* jäsentää JAXP:in avulla palveluiden XML-kuvaukset. Jäsennetyt kuvaukset menevät *matchmaker*-moduulille, joka toteuttaa aiemmin mainitun palveluiden muodostamisen *service locatorin* kanssa. Muodostamissuunnitelmat menevät *optimizer*-moduulille, joka palauttaa hyväksytyt suunnitelmat *matchmakerille*. Ennen kuin suunnitelmat menevät *composite service managerille*, *soundness controller* tarkistaa niiden eheyden.



Kuva 23. WebDG arkkitehtuuri [MRB03].

Kuvan 23 yksityiskohtaisuus hävittää alleen kokonaiskuvan, eikä kuvan graafinen asettelu (kerrokset) ole paras mahdollinen, mutta kuva 22 kertoo paremmin, mistä WebDG:ssä pohjimmiltaan on kyse. Myös OSP-kerrosmallin osat löytyvät. Palveluiden nimeämiseen ja ontologioihin on kiinnitetty erityistä huomiota (kuva 23, WebDG manager). Palvelutarjousvarasto on toteutettu UDDI:lla (Kuva 21, Basic Web services). Palvelun kutsumalli on toteutettu SOAP:lla. Työnkuluista (palvelun koordinoitumalli) ei suoraan puhuta, mutta WebDG Manager kokonaisuudessaan toimii myös työnkulkujen hallintamoottorina. Ja onhan järjestelmän toinen painopiste palveluiden muodostaminen, jossa selvästi on kyse

työnkulkujen muodostamisesta. Lisänä OSP-malliin, on toinen painopiste, tietosuoja, jonka hallintaan on erityisesti panostettu. WebDG:n kehitys jatkuu edelleen.

## 4.2 Sähköisen hallinnon hankkeista saatuja kokemuksia

Edellä esitetyt pilotit eivät vielä ole olleet tuotantokäytössä. Niinpä todellisten yhteispalveluiden käytöstä ei ole vielä saatavilla kokemuksia. Sen sijaan pienimuotoisempia hankkeita on toteutettu. Nämä ovat usein yksittäisten virastojen tarjoamia palveluita. Näistä hankkeista saadut kokemukset on myös hyvä ottaa huomioon yhteispalveluja rakennettaessa. Teknisiin valintoihin vaikuttavat myös ei-tekniset seikat, joten ne on otettava myös huomioon.

YK:n arvion (2002) mukaan 169 maan julkisella hallinnolla on jonkinlaiset verkkosivut. Kuitenkin vain 17 maassa julkinen hallinto tarjoaa jonkinlaisia sähköisiä transaktiopalveluita [Pra03]. Muutamien tutkimusten perusteella n. 85% kaikista sähköisen hallinnon projekteista joko epäonnistuvat täysin, tai ainakin osittain [Pra03]. Prattipatin vuonna 2002 tehdyssä tutkimuksessa [Pra03] on tutkittu sähköisen hallinnon palveluita 30 maassa. Tutkimuksen mukaan palveluita käytetään eniten Ruotsissa, Norjassa, Tanskassa, Singaporessa ja Suomessa. Palveluiden käyttö korreloi siis internetin käytön kanssa, joka näissä maissa on korkealla tasolla. Tutkimuksessa ei huomioitu sitä, että pohjoismaissa julkinen hallinto on paljon suuremmassa roolissa kuin monissa muissa maissa. Myös julkisten internet-palveluiden tarjoaminen (esimerkiksi kirjastot) näyttää vaikuttavan vahvasti sähköisen hallinnon palveluiden käyttöön [Pra03]. Myös palveluiden lisäarvo vaikuttaa siihen, käytetäänkö niitä. Palveluiden käyttäminen sähköisesti täytyy todella tuottaa kansalaiselle jotain lisäarvoa, verrattuna vanhoihin käytäntöihin. Prattipatin arvion mukaan todellisten yhteispalveluiden toteuttaminen on enemmän kiinni poliittisesta tahdosta kuin rahasta.

Euroopassa tehdyn tutkimuksen [HaK00, s.16] mukaan integroiduimpia palveluita löytyy rakennuksiin ja kiinteistöihin liittyvistä prosesseista. Tätä asiaa vasten paikkatietojärjestelmät saattavat tulevaisuudessa muodostua oleelliseksi osaksi yhteispalveluita. Lisäksi esimerkiksi Italiassa liiketoimintalupien haku on säädetty lailla yhdeltä luukulta toimivaksi, joten Italiasta löytyy paljon palveluita tältä alueelta [HaK00, s.16]. Tutkimuksessa havaittiin kuusi aluetta, jotka tuottavat ongelmia projektien menestyksellisessä toteuttamisessa [HaK00, s.21-22]. Projekteja viivästyttävät: 1) organisaatioiden välisen yhteistyön

puute ja 2) puutteellinen lainsäädäntö (muun muassa digitaaliset allekirjoitukset ja tietosuoja). Ennen toteutusta on oltava riittävät infrastruktuurit 3) teknologiassa ja 4) työprosesseissa. Jotta projektit etenevät, on löydettävä tarpeeksi 5) rahoitusta ja 6) poliittista tukea. Tutkimuksessa varoitetaan myös ongelmasta, joka syntyy kun palveluita integroidaan [HaK00, s.27]. Kuka omistaa ja on vastuussa palvelusta, jonka synnyttämiseen tarvitaan useaa tahoja? Tämä on oleellinen kysymys, varsinkin kun mukana on yksityisiä osapuolia.

Toisessa, EU:n rahoittamassa projektissa [Hei02], tutkittiin myös kokemuksia sähköisen hallinnon projekteista. Tämän tutkimuksen mukaan keskeisimmät ongelmat liittyvät todentamistekniikoihin. Valtaosa tämän hetken ratkaisuista käyttää tunnus/salasana käyttöä. Jatkossa lupaavia menetelmiä ovat julkisen avaimen menetelmä ja älykortit. Biometriseen tunnistukseen ei tutkimuksessa uskota. Kehityksen jatkuessa, huolenaiheita ovat myös tiedon luotettavuuden väheneminen, ihmiskontaktien väheneminen ja ihmisten jakautuminen (niihin jotka voivat hyödyntää teknologiaa ja niihin jotka eivät tähän kykene, ”digital divide”). Projektien edistämiseksi yksi tekijä näyttäisi nousevan ylitse muiden: vahva johtajuus [Hei02]. Iso-Britanniassa on myös kiinnitetty huomiota ”digital divide” ongelmaan [ARS02]. Esimerkiksi Koillis-Englantiin on perustettu julkisesti saatavilla olevia internet-kioskeja, joista on pääsy useisiin julkisiin palveluihin. Ihmiset, jotka eivät käytä internetiä tai tietotekniikkaa, eivät ole kuitenkaan ottaneet kioskeja omakseen. Osasyynä on ainakin se, että ajavana voimana ei ole ollut asiakaslähtöisyys, vaan kioskit on toteutettu, koska keskushallinto on niin halunnut [ARS02].

Alankomaissa tehdyissä selvityksissä [Hoo02, Kra02] on todettu samoja ongelmia kuin edellä mainituissa tutkimuksissa, kuten radikaalit organisaatio- ja prosessimuutokset, lainsäädäntö ja organisaatioiden välinen yhteistyö. Yhteistyötä vaikeuttivat piintyneet toimintatavat, mutta myös epäselvät työn- ja vastuunjaot. Varsinkin paikallishallintojen ongelmana on monissa palveluissa pieni transaktioiden määrä. Yhteispalveluiden kehittäminen tällaisille palveluille ei yksinkertaisesti kannata. Lisäksi ongelmia tuottivat yksityisyys, todentaminen ja tietoturva. Monet tietokannat ovat keskushallinnon hallussa, ja täten paikallishallinnon on vaikea omissa projekteissaan järjestää tietojen oikea käyttö. Alankomaissa ongelmana nähdään myös muun muassa kollektiivisten ja yksilöiden etujen ristiriita, ja tietosuojan ja läpinäkyvyyden ristiriita. Alankomaissa julkinen hallinto onkin lähentynyt suuntaan, jossa front-office (asiakasrajapinta) ja back-office (tuotanto) ovat selvästi erottuneet toisistaan. Modulaariset transaktiot mahdollistavat myös sen, että asiakasraja-



pinta voi valita erilaisista tuotantoyksiköistä parhaat. Tuotantoa voidaan tehdä myös yksityisellä puolella. Tuotantoyksiköiden on tehtävä yhteistyötä toistensa kanssa, sen sijaan erilaiset asiakasrajapinnat kilpailevat keskenään. Näiden seurauksena organisaatioiden rajat häipyvät ja kilpailun toivotaan parantavan palveluita. Tässäkin on siis kyse virtuaalisesta organisaatiosta ja palvelumarkkinatyypisistä lähestymistavasta. Tällaiseen dynaamiseen toimintaan liittyy kuitenkin omat vaaransa, ja niinpä Alankomaissa suunnitellaankin lainsäädäntöä tähän tarkoitukseen [Hoo02]. Kehittäminen on lähtenyt alhaalta ylöspäin, kuten monessa muussakin maassa, mutta tästä aiheutuneet ongelmat ovat niin suuria että Kraaijenbrink ehdottaa artikkelissaan [Kra02], että keskitetympi palveluiden kehitys olisi järkevämpää. Muun muassa koordinoinnin ja rahoituksen suhteen tämä olisikin järkevää, mutta tietoteknisesti kovin massiivisen järjestelmän kehitys ei ole järkevää.

Evangelidis ym. [EAT02] listaavat sähköisen hallinnon riskitekijöitä, jotka perustuvat Iso-Britanniasta saatuihin kokemuksiin. Mainitsen tässä vain tekniikkaan ja toteutukseen liittyvät riskitekijät: 1. projektin läpivieminen poliittisen hyödyn toivossa, 2. palveluiden tunnistamisen ja määrittelyn puute, 3. vaatimuksia ei osata tunnistaa, 4. suunnittelun ja ylläpidettävyyden taso, 5. erityisosaamisen puute julkisella sektorilla, 6. integroinnin vaikeus olemassa olevien järjestelmien kesken, 7. osaamisen puute projektien johtamisessa, 8. liian isot projektit, 9. riskien hallinnan puute, 10. tietoturva.

Brasilian G2G-projekteista saatujen kokemusten perusteella löydettiin kolme menestystekijää: tietoturva, organisaatiokulttuurien yhteensovittaminen ja henkilöstön koulutus [Joi03]. Tässä on tosin käsitelty vain G2G-toimintaa, ja toisekseen Brasiliassa tehty tutkimus ei sinällään ole yleistettävissä. Tosin organisaatiokulttuurien yhteydessä huomattiin työnkulkujen määrittämisen tärkeys, jota tässä tutkielmassa on painotettu.

Pardon ja Schollin artikkelissa [PaS02] tarkastellaan suurien sähköisen hallinnon projektien epäonnistumisen syitä. Tutkimusten mukaan projektien epäonnistuminen aiheutuu hyvin harvoin teknisistä tekijöistä, ja valtaosassa epäonnistuminen johtuu inhimillisistä syistä. Artikkelin ajaa sosioteknistä näkemystä, jossa teknologialla on vain pieni rooli. Aihe on kuitenkin tämän tutkielman ydinalueen ulkopuolella, ja ei-tekniisiä ongelmia on jo lyhyesti käsitelty luvussa 2.

### 4.3 Tietoturva ja tietosuoja

Tietoturvaan ja yksityisyyteen liittyviin asioihin on jo viitattu useassa kohtaa aiemmin tässä tutkielmassa, joten tämä luku keskittyykin asian tekniseen puoleen. Kovin syvälle aiheeseen ei mennä, ja koko aihealue on laaja ja käymistilassa.

Tietosuojalla (privacy) pyritään kansalaisen yksityisyyden turvaamiseen. Yksityisyyttä ja tietosuojaa käsitellään tässä tekstissä synonyymeinä. Tietosuoja on erityisen tärkeä julkisessa hallinnossa, koska se on usein lakisääteistä. Laki saattaa määrätä tietyt kriteerit tietosuojan tasolle [SaP97]. Tässä suhteessa virastojen erillisuus on perusteltua. Tiedot eivät saa kulkea virastosta toiseen, ellei siihen ole perusteltua syytä [Hof03]. Lisäksi kaikista muutoksista tietoon on jäätävä lokimerkintä, ja jokainen dataan tehty kirjoitus-operaatio on pystyttävä kohdentamaan muutoksen tekijään ja syyhyn [Hof03]. Tavoitteena on, että kansalainen kokee sähköisen asioinnin yhtä turvalliseksi kuin henkilökohtaisen asioinnin. Käytännössä tämä tarkoittaa *henkilön oikeaksi todentamista, valtuutusta* (kansalainen pääsee vain niihin tietoihin, mihin hänen kuuluukin päästä), *kiistattomuutta* (kumpikaan osapuoli ei voi myöhemmin kiistää transaktiota) ja *transaktion eheyttä* [CCD03]. Nämä neljä ovat toisistaan erillisiä asioita.

Tietojenkäsittelytieteessä tietoturvalle viitataan keinoihin, joilla taataan tiedon *saatavuus, eheys ja luottamuksellisuus*. Saatavuus ja eheys taataan yleensä epäsuorasti jollain mekaniismilla (esimerkiksi palomuurit ja käyttäjän todentaminen), joka suojaa tietoa [Tam01b, s.63]. Sen sijaan luottamuksellisuus taataan yleensä kryptografian eli salauksen avulla [Tam01b, s.63]. Salauksessa lähetettävä sanoma muunnetaan toiseksi jollain algoritmilla ja avaimella. Vastaanottaja purkaa sanoman vastaavasti algoritmin ja avaimen avulla. Salauksen vahvuus riippuu siitä, kuinka vahva algoritmi on, ja pysyykö avain salassa [Tam01b, s.63].

Avain voi olla joko symmetrinen tai julkinen. Jos avain on symmetrinen, sanoman salaus ja purku tehdään samalla avaimella [Tam01b, s.64]. Julkisen avaimen menetelmässä (PKI) sanoma salataan kaikille julkisella avaimella, mutta sen purku onnistuu vain yksityisellä avaimella [Tam01b, s.64]. Julkisen avaimen menetelmää pidetään turvallisempana, mutta sen laskennallinen vaatavuus tekee siitä hitaan menetelmän [Tam01b, s.64-65]. Näiden menetelmien hybridinä käytetäänkin PGP (Pretty Good Privacy)-menetelmää. PGP:ssä käytetään symmetristä avainta salaamiseen ja purkuun, ja lisäksi yksityistä istunto-avainta

[Tam01b, s.65]. PGP on nopeampi kuin julkisen avaimen menetelmä, mutta silti sitä pidetään erittäin turvallisena [Tam01b, s.65].

Webocrat nimiseen EU:n rahoittamaan järjestelmään on kehitetty kehys nimeltään e-GOV-OFSR (Organizational Framework for the Security Requirements of e-government services) [LGD03]. Arkkitehtuurissa on käytetty vain PKI-menetelmää, ja kehittäjien mukaan suurin osa julkisen hallinnon portaalien tietoturvaongelmista voidaan ratkaista käyttämällä julkisen avaimen menetelmää. Kehyksen olennaisimpia asioita on riskikartoituksen tekeminen ja riskitasojen arviointi, sekä toisaalta tietoturva vaatimusten suhde edellisiin. Projektin tuloksena on toteutettu CSAP (Communication, Security, Authentication and Privacy)-arkkitehtuuri, joka on täysin PKI-pohjainen, ja se on integroitu Webocrat-järjestelmään [LGD03]. Kuten nimikin kertoo, arkkitehtuurilla voidaan tietoturvan lisäksi puuttua myös tunnistamiseen ja tietosuojaan. Koko kehyksen lähtökohtana on kuitenkin se, että organisaatioiden tietojärjestelmien rajat on tiukasti määritelty. Perinteisesti julkisessa hallinnossa tämä pitää paikkansa, mutta asia on kuitenkin ristiriidassa uusien integrointikeinojen, kuten virtuaalisten organisaatioiden kanssa.

Sanomien lähetyksessä tulisi varmistaa että sanoman lähettäjä on varmasti se, kuka tämä väittää olevansa, sekä sanoman koskemattomuus. Tähän tarkoitukseen käytetään sähköiseksi allekirjoitukseksi kutsuttua menetelmää, joka perustuu julkisen avaimen menetelmään. Julkisen avaimen menetelmässä taas on oltava varmoja siitä, että käytettävissä oleva julkinen avain on aito. Tässä käytetään hyväksi sähköisiä sertifikaatteja [Tam01b, s.66].

XML-kielestä löytyy spesifikaatiot sähköisen allekirjoituksen käytölle. Sähköiset allekirjoitukset esitetään XML-kielessä `signature` elementeillä [Tam01b, s.67]. Myös salaukselle löytyy syntaksi. Elementillä `EncryptedData` voidaan joko viitata (referencing) tai sisällyttää (enveloping) salattua tietoa [Tam01b, s.70]. W3C:n tietosuojasuositukset ovat kuitenkin kokonaisuudessaan alkutekijöissään. Konkreettisia keinoja esimerkiksi web-palveluiden tietosuojaan ei ole esitetty [ROB02].

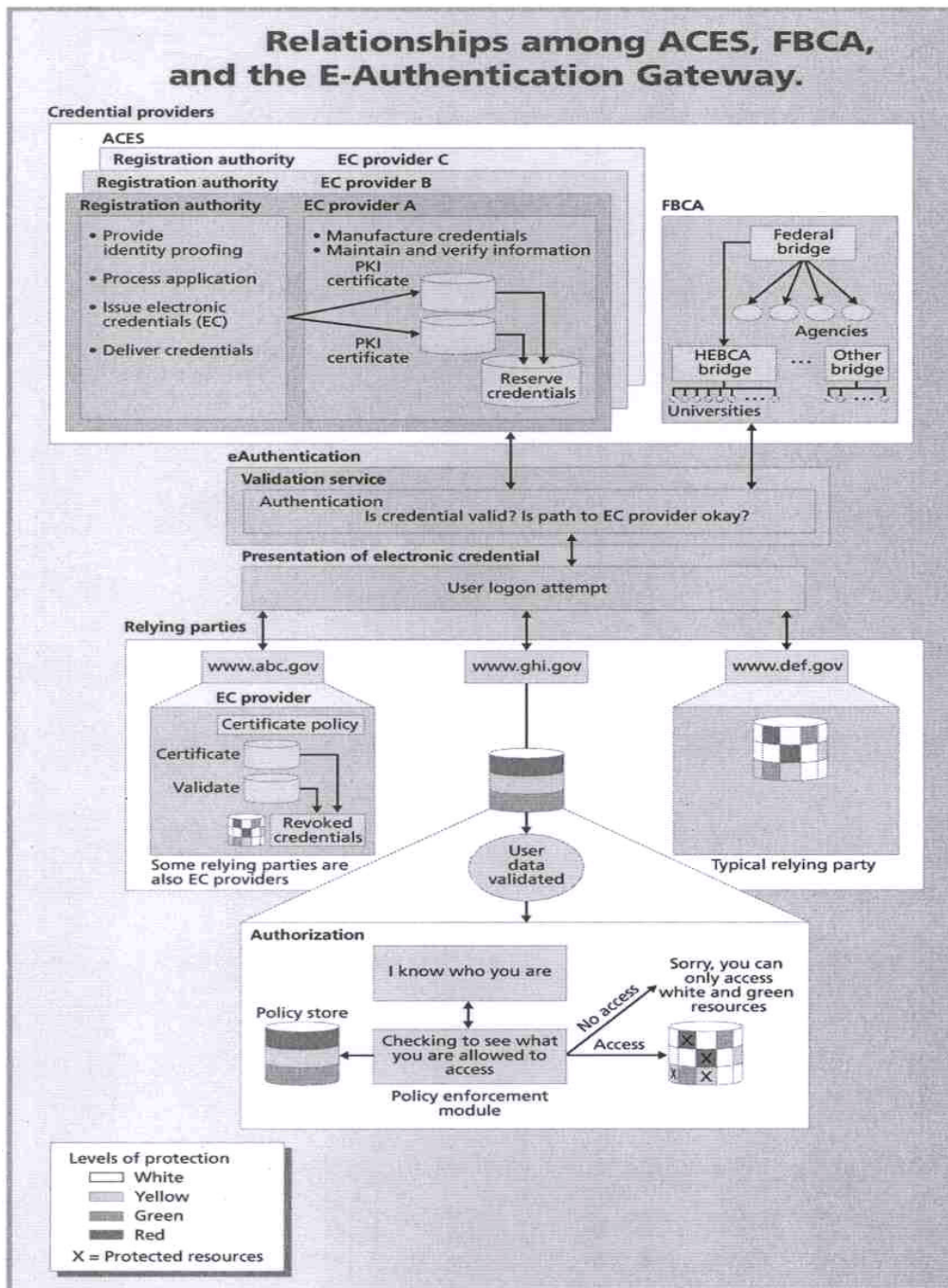
Luvussa 4.1.4 esitellyssä WebDG-järjestelmässä [ROB02] on kiinnitetty erityistä huomiota tietosuojaan. Siinä tietosuoja on jaettu kolmeen osaan: käyttäjään, dataan ja palveluun. Käyttäjän tietosuojassa käyttäjä itse voi määritellä oman tietosuojaprofiilinsa. Palvelun

tietosuoja määrittelee yksittäisen palvelun tietosuojan kaikille käyttäjille. Datan tietosuoja määrittelee erilaiset näkymät eri käyttäjille samasta tieto-olioista. Tieto-olioista, joiden tietosuojaprofiili on samanlainen, voidaan muodostaa tietosuojarykelmiä (*privacy cluster*), joilla on yksi globaali profiili. Malli perustuu valtakirjoihin (*digital privacy credentials*), suodattimiin (*data filters*) ja liikkuviin agenteihin (*mobile privacy enforcement agents*).

Jotta kansalainen voi käyttää palvelua, hänen on ensin saatava *valtakirja* (DPC), joka määrittelee käyttäjän käyttöoikeudet. Palvelupyynnön tullessa, järjestelmä tarkistaa ensin valtakirjan. Sitten pyyntö menee *suodattimelle* (DFilter), joka käsittelee pyyntöä (tämä on kuvattu luvussa 4.1.4). *Rykelmienhallinta* (CM, Clusters Manager) ylläpitää tietosuojarykelmiä. Edellä kuvatut toiminnot ovat kuitenkin keskitetyssä systeemissä, joten tietosuoja tulee taata myös paikallisesti pyytävän osapuolen päässä tai kolmansien osapuolien ollessa kyseessä. Tätä varten WebDG-järjestelmässä käytetään liikkuvia tietosuojan takaavia agenteja (MPPA) [ROB02]. Kun palvelupyynnön tulos on muodostettu, *agenttien hallinta* (AM) muodostaa *agentin*, joka lähetetään pyytäjälle. *Agentti* koostuu kahdesta moduulista, datasta ja koodista. *Agentti* säätelee vastaanottavassa päässä pääsyä dataosioon. *Agentit* on toteutettu käyttäen Aglets SDK:ta, joka on Java-pohjainen liikkuvien agenttien alusta. *Aglet* on liikkuvaa koodia ja tilatietoja, jotka voivat liikkua internetissä. Palvelimen päässä *agletteja* ajaa siihen erityisesti suunniteltu DGWebServer. Liikkuviin agenteihin liittyy kuitenkin vielä tietoturvaongelmia, joista osa on vielä ratkaisematta. Lisäksi paikallisten tahojen on toteutettava oikein *aglettien* vaatima ympäristö. Tämä onkin parhaillaan jatkotutkimuksen kohteena [ROB02].

USA:ssa on hallituksen aloitteesta käynnistetty useita tietosuojan parantamiseen tähtäviä ohjelmia (kuva 24) [CCD03]. *ACES* (Access Certificates for Electronic Services)-ohjelman tuloksena kolme PKI-menetelmän tarjoajaa tuottaa julkiselle hallinnolle varmennus ja sertifikaattipalveluita. Hallinnon yksiköiden ei ole tarvinnut tuottaa omia PKI-järjestelmiä, vaan kaikilla oli mahdollisuus käyttää projektissa tuotettua järjestelmää. Projektissa tuotettiin avoimen lähdekoodin ohjelmisto *CAM* (Certificate Arbitration Module), joka huolehtii eri tahojen sertifikaattien yhteistoiminnasta. Ohjelman ulkopuolella monet julkisen hallinnon tahot käynnistivät omia PKI-ohjelmiaan. Jotta nämä saataisiin toimimaan yhteen *ACES*-ohjelman sertifikaattien kanssa, muodostettiin *FBCA* (Federal Bridge Certification Authority). Eri PKI-toimialueet toimivat *FBCA*:n kautta, joka muodostaa luottamuspolkua (trust paths) toimialueiden välille. Keskustelua on myös käyty *FBCA*:n käytöstä maail-

manlaajuisesti. Kolmas laaja ohjelma on nimeltään *E-Authentication Gateway*. Ohjelmas-  
sa toteutettu infrastruktuuri tarjoaa todentamislustan, jonka eri julkisen hallinnon tahot  
voivat halutessaan ottaa käyttöön. *ACES* ja *FBCA* siis lähinnä tarjoavat valtakirjan ja serti-  
fikaatin, kun taas *EAG* tarjoaa itse kirjautumisrajapinnan palveluihin.



Kuva 24. ACES, FBCA ja E-Authentication Gateway [CCD03].

USAssa on siis myös lähdetty siitä, että jokainen julkisen hallinnon taho voi kehittää omia järjestelmiään. Keskushallinto yrittää sitten edellä kuvatulla tavalla koordinoida eri tahoja yhteistyöhön.

Bistro-projektissa [GCC03] on kehitetty turvallista ja skaalautuvaa tiedonsiirtojärjestelmää julkiselle hallinnolle. Laajoissa alueverkoissa (WAN) käytetyissä sovelluksissa yleisesti käytössä oleva TCP/IP johtaa yksittäisten asiakkaiden (client) kohdalla helposti huonoon suorituskykyyn [GCC03]. Bistrot toimivat eräänlaisina välityspalvelimina, jotka levittävät tietoliikennekuormaa, ja täten parantavat sekä asiakkaan että palvelimen suorituskykyä. Lisäksi Bistron tietoturvaominaisuudet on rakennettu niin, että vain salattu tieto kulkee niiden läpi [GCC03]. Bistron kehitys on vielä kesken, ja tekijöiden suurista toiveista huolimatta, sitä ei ole testattu todellisessa ympäristössä. Lisäksi Bistron toiminta liittyy matalammalle tasolle (tietoliikenne ja tietoturva) kuin tämän tutkielman näkökulma.

Edellä esitettyjen hankkeiden perusteella voi todeta, että ainakin Yhdysvalloissa ideoita ja pilotteja on paljon. Seurauksena on kuitenkin lähinnä sekasotku ja koordinoinnin vaikeus, kun erilaisia ideoita yritetään saada toimimaan keskenään. Järkevämpää olisi sopia tietyistä linjauksista, jotka kaikki tahot hyväksyisivät. PKI-menetelmä näyttäisi tällä hetkellä salauksen kannalta vahvimmalta ehdokkaalta, mutta isoissa järjestelmissä laskennallinen vaativuus voi aiheuttaa ongelmia. PGP on tässä suhteessa parempi vaihtoehto.

## 5 Johtopäätökset

Tämän tutkielman tarkoituksena oli selvittää, onko julkisen hallinnon sähköisiin yhteis- palveluihin muodostunut vakiintuneita standardiratkaisuja, ja mitkä ongelmat ovat vielä ratkaisematta. Lisäksi tarkastelun kohteena ovat luvussa 2.4 asetetut vaatimukset yhteis- palveluille.

Seuraavassa on arvioitu vaatimusten toteutumista esitellyissä projekteissa. Koska arviointi on tehty artikkeleissa esitettyjen tietojen perusteella, on siihen suhtauduttava varauksella. Monessa tapauksessa nimittäin artikkelin kirjoittajat ovat itse vastuussa järjestelmän toteu- tuksesta ja/tai suunnittelusta, ja saattavat siten kuvata järjestelmää subjektiivisen hyvässä valossa. Toisaalta kuvauksessa on voitu jättää kertomatta asioita, joita kirjoittaja ei pidä järjestelmän painopistealueina. Taulukossa 1 on merkattu nollalla ne vaatimukset, joista ei ole mainintaa, tai joita järjestelmästä ei löydy. Ykkösellä on merkitty kohdat, jotka maini- taan, tai selvästi löytyvät kyseisestä järjestelmästä. Tässä on huomioitava, että sitä ei ole arvioitu, miten hyvin jokin vaatimus järjestelmässä on toteutunut.

	ADSS-OSCAR	PASSPORT	eGOV	WebDG
kohderyhmien tarpeet	0	0	1	0
monikanavaisuus	1	0	1	0
monimutkaisuus/työnkulut	1	1	0	1
integrointi	1	1	1	1
tietoturva	0	0	1	0
tiedon suojaus ja läpinäkyvyys	0	1	1	1
luotettavuus ja käytettävyys	0	0	1	0

Taulukko 1: Vaatimusten toteutuminen neljässä yhteispalvelujärjestelmässä.

Kohderyhmien tarpeet on vaikeasti arvioitava vaatimus. Ainoastaan eGOVin yhteydessä siihen on otettu selvästi kantaa, ja projektin yhteydessä onkin tehty tarvekartoitusta muun muassa haastattelututkimuksilla. Muiden projektien kohdalla on huolestuttavaa, että koh- deryhmien tarpeista ei puhuta, koska juuri tarpeiden tulisi olla järjestelmän toiminnalli- suuden määrittäjänä ja lähtökohtana.

Monikanavaisuuden toteutuminen on helpommin arvioitava vaatimus, ja sille löytyy tukea ainakin ADSS-OSCARista ja eGOVista. Työnkulkujen hallintaan on välineitä kaikissa

muissa toteutuksissa, paitsi eGOVissa. EGOVissa elämäntapahtumat määrittelevät prosessit, mutta varsinaisesta työnkulkujen hallinnasta ei löytynyt mainintaa. Tarvittava yhden pysähdyksen integraatio löytyy jokaisesta projektista. Sitä, miten hyvin yhden pysähdyksen integraatio on toteutunut, on kuitenkin vaikeaa täysin arvioida näillä tiedoilla.

EGOV on ainoa projekti, jossa on suoraan kiinnitetty huomiota tietoturvaan. WebDG:ssä taas lähdetään suoraan siitä oletuksesta, että tietoturva on jo kunnossa. Tietoturva ei mielestäni kuulukaan yhteispalvelujärjestelmien ydinalueeseen. Tietosuojaan taas on otettu kantaa WebDG:ssä ja eGOVissa. WebDG:ssä tietosuoja on yksi järjestelmän ydinalueista ja eGOVissa siihen pureudutaan XML:n omien välineiden avulla. Mahdollisuus oman prosessin seuraamiseen (läpinäkyvyys) mainitaan vain PASSPORTissa.

Luotettavuus seuraa muun muassa tietoturvasta ja tietosuojasta. Käytettävyyteen on viitattu vain eGOVissa, jossa asiaan on kiinnitetty laajemmin huomiota. Käytettävyys on kytköksissä ensimmäiseen vaatimukseen, käyttäjien tarpeisiin. Toisaalta käytettävyys on tietoturvan ohella toinen tekijä, joka ei kuulu tämän tutkielman ja sovellusintegraation ytimeen.

Jos vaatimusten toteutumista arvioidaan edellisen taulukon perusteella, parhaiten on pärjännyt eGOV. Kokonaisuutena myös PASSPORT ja WebDG jättävät hyvän vaikutelman yksityiskohtaisuudellaan. Pois lukien eGOV, kaikissa projekteissa on puutteita kohderyhmien todellisten tarpeiden huomioimisessa, joka on kytköksissä myös läpinäkyvyyteen ja käytettävyyteen.

Seuraavassa taulukossa (taulukko 2) arvioidaan neljän projektin sijoittumista luvussa 2.5 esitettyyn OSP-malliin. Työnkulkujen hallintamoottori löytyy kaikista muista, paitsi eGOVista. WebDG:ssä on lisäksi eritelty tarkemmin työnkulkujen muodostamista. Abstrakti palvelun kutsumalli on kolmessa projektissa XML-pohjainen, vain PASSPORTissa on jätetty yhteinen kanava sopimatta, ja kutsut pyritään hoitamaan erityisillä kommunikatiomoduleilla. eGOVissa on kehitetty täysin oma XML-toteutus, GovML. Abstraktiin palvelun viitemalliin on otettu kantaa täydellisimmin WebDG:ssä ja eGOVissa. ADSS-mallin lokerointi OSP-mallin mukaan on vaikeampaa. Ba toimii kuitenkin virtuaalisena yhteisönä ja Planner-komponentti palveluiden nimeäjänä, ja katson näiden kuuluvan abstraktiin palvelun viitemalliin.



	ADSS-OSCAR	PASSPORT	eGOV	WebDG
abstrakti palvelun koordinointimalli	Flow Manager	Workflow Server	-	WebDG manager, muodostusmallit
abstrakti palvelun kutsumalli	XML, SOAP	Communication Gateway	Useita, mm. SOAP	SOAP
abstrakti palvelun viitemalli	Ba, Planner, Broker	-	GovML, life-events, WSDL, UDDI, RDF	WSDL, UDDI, Category ja Type ontologiat
viestintäalusta	HTTP	HTTP, sähköposti, WAP jne.	HTTP, GSM jne.	HTTP

Taulukko 2: OSP-kerrosmalli neljässä yhteispalvelujärjestelmässä.

Vaikka OSP-malli on kehitetty PASSPORTin taustatyönä, tekninen toteutus näyttää viitemallia vasten vaillinaiselta. WebDG sen sijaan täyttää mallin jokaisen kohdan.

Virtuaalisten organisaatioiden näkökulmasta kaikki neljä hanketta noudattavat palveluliit-to-lähestymistapaa. Kerrospohjainen lähestymistapa on jäämässä pois vanhentuneena, ja agenttitekniologiat ovat vasta kehityksessä. Tosin WebDG:ssä on käytetty myös agenttitekniologiaa lisänä. Myös kasvuympäristön käsitettä on jossain määrin sovellettu, mutta julkinen hallinto asettaa rajoituksia kumppaneiden valinnalle, kuten pohdin luvussa 3.4.2. Toisaalta EU-alueella julkisen hallinnon on kilpailutettava hankkeensa, ja tässä mielessä sovellettu kasvuympäristön käsite voisi toimia. Tässä ei kuitenkaan enää puhuta aivan samasta asiasta kuin mitä alkuperäisellä kasvuympäristön käsitteellä tarkoitettiin.

IMPULSE-projektin (Luku 3.4.3) työnkulkujen hallintamoottori on yksityiskohtaisuuden tasolla pitkälle viety arkkitehtuuri. Kyseinen järjestelmä on kuitenkin ensisijaisesti työnkulkujen hallintaan painottuva, eikä sitä voida pitää yhteispalvelujärjestelmänä. IMPULSE perustuu agenttitekniologiaan, ja olettaa että muun muassa palveluhakemisto löytyy järjestelmän ulkopuolelta. Täten IMPULSEN ja jonkin yhteispalveluhankkeen yhdistäminen (esimerkiksi eGOV) voisi tuottaa lisäarvoa sähköisten yhteispalveluiden kehittämiseen. Yhteispalveluprojektit ovat kuitenkin ensisijaisesti integrointiprojekteja, vaikkakin työnkulkujen hallinta liittyy läheisesti integrointiin. WISE-malli taas soveltuu ainakin prosessien mallintamiseen.

Tutkielman tutkimusongelmana oli: ”onko yhteispalveluiden toteutuksessa muodostunut jo vakiintuneita standardiratkaisuja, ja mitä ongelmia on vielä ratkaisematta?”. *Vakiintuneena ratkaisuna on XML, ja erityisesti siihen pohjautuva web-palvelut, joka koostuu WSDL, UDDI ja SOAP – toteutuksista.* Teknisemmällä puolella vahvimmin ovat esille J2EE ja sovelluspalvelimet. Web-palvelut arkkitehtuuria on käytetty joko täydellisenä, tai sitten osia siitä. Myös muut XML-pohjaiset ratkaisut ovat vahvasti esillä, kuten GovML. Edellisiä toteamuksia ei voi kuitenkaan pitää lopullisina. Aihe-alueen nuoruudesta johtuen, konkreettisia hankkeita on vasta muutama. Artikkeleissa esitetyt arviot kuitenkin tukevat edellä mainittujen tekniikoiden mahdollisuuksia. J2EE:n asema näyttää tällä hetkellä vahvalta, mutta nopea teknologinen kehitys voi muuttaa tilannetta nopeasti.

Ratkaisemattomia ongelmia on sen sijaan paljon, ja monet niistä ovat aihe-alueen laajuuden vuoksi tämän tutkielman rajauksen ulkopuolella. Suurimmat ongelmat ovat kuitenkin ei-tekniisiä, ja liittyvät esimerkiksi poliittiseen päätöksentekoon ja organisaatioiden hallintaan. Tekniset työkalut sähköisten yhteispalveluiden rakentamiseen ovat olemassa.

Koska tuotantokäytössä olevia täydellisiä yhteispalvelujärjestelmiä ei ole, kaikkia sähköisiin yhteispalveluihin liittyviä ongelmia ei vielä edes tiedetä. Yhteispalveluiden perimmäisenä tarkoituksena on kansalaisten parempi palveleminen. Useimpien hankkeiden ajavana voimana on kuitenkin jokin muu asia, vaikka lähes kaikissa kirjallisissa lähteissä painotetaan asiakaskeskeisyyden tärkeyttä. Samaan asiaan liittyviin läpinäkyvyyteen ja käytettävyyteen, ei ole myöskään kiinnitetty tarpeellista huomiota. Käyttöliittymätutkimuksessa tehty edistys olisi saatava myös sovellusintegraation pariin.

Työnkulkujen hallintaan on olemassa valmiita ohjelmistoja, jotka saadaan upotettua laajempaan järjestelmään. Ongelmana eivät olekaan työnkulkujenhallintaohjelmistot. Suurin ongelma on työnkulkujen määrittely, joka julkisen hallinnon kaltaisessa pirstoutuneessa ympäristössä on vaikeaa. Tässä tulisivikin hakea oppia erityisesti työnkulkujen ongelmiin liittyviltä tahoilta. Luvuissa 3.4.2 ja 3.4.3 on kuvattu tätä ongelmakenttää esimerkiksi virtuaalisten organisaation ja kasvuympäristön käsitteiden avulla.

Tietoturvaan liittyvät ongelmat ovat yleisiä, eivätkä vain yhteispalveluihin liittyviä. Tietoturvan katson kuuluvaksi yhteispalveluiden ydinalueen ulkopuolelle. Tämä ei tarkoita sitä, että tietoturva olisi vähäpätöinen asia, koska se on kansalaisten luottamuksen kannalta

todella tärkeässä roolissa. Yhteispalveluiden kehittämisen kannalta tietoturva tulisi kuitenkin nähdä annettuna asiana, eikä siihen tulisi uhrata suurta osuutta kehitystyön henkilötyöpäivistä. Tietosuojaa sen sijaan liittyy lähemmin yhteispalveluiden ytimeen. WebDG on malliesimerkki siitä, kuinka tietosuojaa kuuluu luonnollisena osana muuhun yhteispalvelujärjestelmään.

Tämän tutkielman kannalta olennaisin osa-alue, itse integrointi on jatkuvassa kehitystilassa. On olemassa useita käyttökelpoisia korkean tason malleja, kuten web-palvelut, joka tässä tutkielmassa on todettu standardiksi. Alemman tason mekanismeja on vielä lukuisampi määrä, ja monet niistä ovat yksityisellä puolella jo hyviksi havaittuja. Vaikka yhteispalveluprojekteissa on kehitetty täysin uutta tekniikkaa, ratkaisun pohjalla on usein jokin tunnettu standardi, kuten J2EE tai CORBA. Lähdeaineistossa ei ole kiinnitetty paljoa huomiota suorituskyykyyn, vaikka Java ja XML eivät ole oikeita välineitä silloin, kun järjestelmältä halutaan ensisijaisesti suorituskyykyä. Suorituskyykyongelmat ilmenevät usein vasta sitten kun laaja järjestelmä otetaan tuotantokäyttöön (tästä on Suomestakin useita esimerkkejä, kuten [www.helmet.fi](http://www.helmet.fi)). Pilottikäytössä olevat järjestelmät eivät yleensä aiheuta suorituskyykyongelmia, joten kuormantasaus, topologiat ja tietoliikenteen optimointi on ratkaistava ennen tuotantoon ottoa.

Tämä tutkielma on tuonut lisää tietoa uudesta aihealueesta, julkisen hallinnon sähköisistä yhteispalveluista. Se on ensimmäinen suomenkielinen katsaus aihealueeseen. Tutkielmassa on myös pyritty nivomaan yhteen vielä varsin hajanaista tietoa. Vaikka tarkastelunäkökulma on tietojenkäsittelytieteilijän, on mukaan otettu myös paljon poikkitieteellisempää materiaalia. Tutkielma toimii täten ensisijaisesti state-of-the-art katsauksena aihealueeseen. Pääosa materiaalista koostuu tieteellisistä artikkeleista. Tästä johtuu myös näkökulman abstraktius ja korkealla tasolla oleminen. Syvälle tekniikkaan on menty vain muutamassa tapauksessa. Koska projekteja tehdään myös akateemisen maailman ulkopuolella, voi vain arvailla minkälaisia ratkaisuja on kehitteillä. Toisaalta julkisen hallinnon ”julkisuus” tuo tällaiset mahdolliset hankkeet esille ennen pitkää.

Vastaukseni tutkimusongelmaan oli lyhyt standardien osalta, ja liian laaja ratkaisemattomien asioiden osalta. Vaikka tiesin kentän suppeuden ja nuoruuden, varsinkin konkreettisten projektien vähäisyys yllätti. Lähteitä löytyy paljon, mutta niistä monet ovat aivan liian yleisellä tasolla, ja toistavat samoja asioita, jotka lisäksi usein alkavat sanalla ”pitäisi”.

Tämä näkyy myös väistämättä tässä tutkielmassa, erityisesti luvussa 2. Vaikka materiaalia on useasta maanosasta, ei johtopäätöksiä pidä yleistää. Pilottiprojektien vähäisyys ja se, että yksikään järjestelmä ei ole laajassa tuotantokäytössä, estää tekemästä pitkälle menevää yleistämistä.

Jatkotutkimuksen kannalta aihealue vaatisi lisää teknisiä artikkeleita ja uusi projekteja. Myös omien mallien ja pilottien kehittäminen tässä vaiheessa olisi hedelmällistä. Lisäksi aihealueen laajuus tarjoaa mahdollisuuden rajatumpiin näkökulmiin. Esimerkiksi tietoturva tai tietosuoja ovat tällaisia. Toisaalta, jos kontekstina on julkinen hallinto, lähdemateriaalin niukkuus tuottaa ongelmia, joten tietoa on sovellettava yleisemmistä lähteistä.

## Lähteet

- Apa04 Apache <Web Services /> Project – Axis (2004).  
<http://ws.apache.org/axis/> [1.4.2004].
- ARS02 Ashford, R., Rowley, J., Slack, F., Electronic Public Service Delivery through Online Kiosks: The User’s Perspective. *Lecture Notes in Computer Science*, 2456 (2002), sivut 169-172.
- Cam03 Camarinha-Matos, L., Infrastructures for virtual organizations – where we are, *Proceedings of ETFA’03 – 9<sup>th</sup> Int. Conf. On Emerging Technologies and Factory Automation* (2003), sivut 405-414.
- CCD03 Caloyannides, M., Copeland, D., Datesman, G., Weitzel, D., US E-Government Authentication Framework and Programs. *IT Professional*, 5, 3 (2003), sivut 16-21.
- Cro02a Cross-Border BUSiness Intermediation through Electronic Seamless Services, D6.1 Project Presentation (2002). <http://www.cb-business.com/documents/D6.1%20Project%20Presentation.pdf>  
 [28.4.2004].
- Cro02b Cross-Border BUSiness Intermediation through Electronic Seamless Services, D6.1.1 State of the Art Analysis (2002). <http://www.cb-business.com/documents/D6.1.1%20-%20State%20of%20the%20Art%20Analysis%20-%20final.pdf>  
 [28.4.2004].
- DoM02 Donnelly, V., Merrick, R., Community Portals Through Communitization. *Proceedings of the 2003 Conference on Universal usability*, (2002), sivut 9-14.
- EAT02 Evangelidis, A., Akomode, J., Taleb-Bendiab, A., Taylor, M., Risk Assessment & Success Factors for e-Government in a UK Establishment. *Lecture Notes in Computer Science*, 2456 (2002), sivut 395-402.

- Erk02 Erkkilä, T., (ed.), An Integrated Platform for Realising Online One-Stop Government (eGOV): Platform and Network Architecture Functional Specifications, D111, (2002).
- FST00 Funabashi, M., Strick, L., Toyouchi, J., Service Integration Platform based on TINA 3-tier Model and Interfaces. *Proceedings of TINA 2000 Conference*, (2000).
- GaG02 Gant, J., Gant, D., Web portal functionality and State government E-service. *Proceedings of the 35<sup>th</sup> Annual Hawaii International Conference on System Sciences (HICSS-35'02)*, sivut 1637-1646.
- GAH00 Grefen, P., Aberer, K., Hoffner, Y., Ludwig, H., CrossFlow: Cross-Organizational Workflow Management in Dynamic Virtual Enterprises; *International Journal of Computer Systems Science & Engineering*, 15, 5, (2000), sivut 277-290.
- GCC03 Golubchik, L., Cheng, W., Chou, C., Khuller, S., Samet, H., Wan, Y., Bistro: A Scalable And Secure Data Transfer Service For Digital Government Applications. *Communications of the ACM*, 46, 1 (2003), sivut 50-51.
- Gla02 Glassey, Olivier, A One-Stop Government Prototype Based On Use Cases and Scenarios. *Lecture Notes in Computer Science*, 2456 (2002), sivut 116-123.
- GLL02 Gouscos, D., Laskaridis, G., Lioulias, D., Mentzas, G., Georgiadis, P., An Approach to Offering One-Stop e-Government Services – Available Technologies and Architectural Issues. *Lecture Notes in Computer Science*, 2456 (2002), sivut 264-271.
- GMG01 Gouscos, D., Mentzas, G., Georgiadis, P., PASSPORT A Novel Architectural Model for the Provision of Seamless Cross-Border e-Government Services. *12th International Workshop on Database and Expert Systems Applications* (2001), sivut 318-322.

- HaK00 Hagen, M., Kubicek, H., (ed.), One-Stop-Government in Europe: Results from 11 national surveys. University of Bremen, 2000. <http://infosoc2.informatik.uni-bremen.de/egovernment/cost/one-stop-government/home.html> [14.2.2004].
- HCA01 Holowczak, R., Chun, S., Artigas, F., Atluri, V., Customized Geospatial Workflows for E-Government Services. *Proceedings of the ninth ACM international symposium on advances in geographic information systems*, (2001), sivut 64-69.
- Hei02 Heinderyckx, F., Assessing e-Government Implementation Processes: A Pan-European Survey of Administration Officials. *Lecture Notes in Computer Science*, 2456 (2002), sivut 111-115.
- Hit00 ADSS: Autonomous Decentralized Service System – An agent based Software Architecture for Large Scale and Ever Changing Information Market Places. *Hitachi Review*, 49, 3 (2000), sivu 128. [http://www.hitachi.com/rev/2000/revnov00/pdf/r3\\_105.pdf](http://www.hitachi.com/rev/2000/revnov00/pdf/r3_105.pdf) [23.4.2004].
- Ho02 Ho, A. Reinventing Local Governments and the E-Government Initiative. *Public Administration Review*, 62 (2002), sivut 434-444.
- Hof03 Hof, S. Security Aspects within E-Government. *Lecture Notes in Computer Science*, 2739 (2003), sivut 266-271.
- Hoo02 Hoogwout, M., Organizing for Online Service Delivery: The Effects of Network Technology on the Organization of Transactional Service Delivery in Dutch Local Government. *Lecture Notes in Computer Science*, 2456 (2002), sivut 33-36.
- Hov03 Hovy, E., Using an Ontology to Simplify Data Access. *Communications of the ACM*, 46, 1 (2003), sivut 47-49.

- Joi03 Joia, L., Key Success Factors for Electronic Inter-organisational Cooperation between Government Agencies. *Lecture Notes in Computer Science*, 2645 (2003), sivut 76-81.
- KaT03 Kavadias, G., Tambouris, E., GovML: A Markup Language for Describing Public Services and Life Events. *Lecture Notes in Computer Science*, 2645 (2003), sivut 106-115.
- KIW02 Klieschewski, R., Wetzal, I., Serviceflow Management: Caring for the Citizen's Concern in Designing E-Government Transaction Processes. *Proceedings of the 35<sup>th</sup> Annual Hawaii International Conference on System Sciences (HICSS-35'02)*, (2002), sivut 1637-1646.
- Kra02 Kraaijenbrink, Jeroen, Centralization Revisited? Problems on Implementing Integrated Service Delivery in The Netherlands. *Lecture Notes in Computer Science*, 2456 (2002), sivut 10-17.
- Kre02 Krenner, J., Reflections on the Requirements Gathering on an One-Stop Government Project. *Lecture Notes in Computer Science*, 2456 (2002), Sivut 124-128.
- Lam03 Lambrou, M., Advancing the One-Stop Shop E-Government Paradigm. *Engineering Management Conference IEMC '03. Managing Technologically Driven Organizations: The Human Side of Innovation and Change* (2003), sivut 489-493.
- LAS00 Lazcano, A., Alonso, G., Schuldt, H., Schuler, C., The WISE approach to Electronic Commerce. *International Journal of Computer System Science and Engineering*, 15, 5 (2000), sivut 343-355.
- LGD03 Lambrinoukadis, C., Gritzalis, S., Dridi, F., Pernul, G., Security requirements for e-government services: a methodological approach for developing a common PKI-based security policy. *Computer Communications*, 26 (2003), sivut 1873-1883.



- Len02 Lenk, K., Electronic Service Delivery – A driver of public sector modernization. *Information Polity*, 7 (2002), sivut 87-96.
- LeT02 Lenk, K., Traunmüller, R., Electronic Government: Where Are We Heading? *Lecture Notes in Computer Science*, 2456, (2002), sivut 1-9.
- Lin01 Linthicum, D., *B2B Application Integration*, Addison-Wesley (2001).
- MaM03 Marche, S., McNiven, J., E-Government and E-Governance: The Future Isn't What It Used To Be. *Canadian Journal of Administrative Sciences*, 20 (2003), sivut 74-86.
- MeP01 Mecella, M., Pernici, B., Designing wrapper components for e-services in integrating heterogeneous systems. *The VLDB Journal*, 10 (2001), sivut 2-15.
- MRB03 Medjahed, B., Rezgui, A., Bouguettaya, A., Ouzzani, M., Infrastructure for E-Government Web Services. *IEEE Internet Computing*, 7, 1 (2003), sivut 58-65.
- MSB03 Marchionini, G., Samet, H., Brandt, L., Digital Government. *Communications of the ACM*, Vol. 46, No. 1 (2003), sivut 24-27.
- PaM04 Pappa, D., Makropoulos, C., Designing a Brokerage Platform for the Delivery of E-government Services to the Public. *Lecture Notes in Computer Science*, 3035, (2004), sivut 178-189.
- Pap03 Pappa, D. (ed.), An Integrated Platform for Realising Online One-Stop Government (eGOV): Platform integration and deployment report, descriptions of the trials, D312 (2003) [http://www.egov-project.org/egovsite/eGOV\\_D312.zip](http://www.egov-project.org/egovsite/eGOV_D312.zip) [24.10.2004].

- PaS02 Pardo, T., Scholl, H., Walking Atop the Cliffs: Avoiding Failure and Reducing Risk in Large Scale E-Government Projects. *Proceedings of the 35<sup>th</sup> Annual Hawaii International Conference on System Sciences (HICSS-35'02)*, (2002).
- PDE02 Pasic, A., Diez, S., Espinosa, J., IMPULSE: Interworkflow Model for e-government. *Lecture Notes in Computer Science*, 2456, (2002), sivut 472-479.
- PHA01 Papantoniou, A., Hattab, E., Afrati, F., Kayafas, E., Loumos, V., Change Management, a Critical Success Factor for e-Government. *Database and Expert Systems Applications. Proceedings. 12<sup>th</sup> International Workshop*. (2001), sivut 402-406.
- Pra03 Prattipati, S., Adoption of e-Governance: Differences between countries in the Use of Online Government Services. *Journal of American Academy of Business, Cambridge*, 3 (2003), sivut 386-391.
- Ric99 Ricci, A., Measuring Information Society 98. Information Society Project Office.  
<http://europa.eu.int/ISPO/polls/98/poll198/> [18.2.2004].
- ROB02 Rezgui, A., Ouzzani, M., Bouguettaya, A., Medjahed, B., Preserving Privacy in Web Services. *Proceedings of the fourth international workshop on web information and data management*, (2002), sivut 56-62.
- Ros04 RosettaNet Overview (2004), <http://www.rosettanet.org> [1.4.2004].
- SaP97 Saarenpää, A., Pöysti, T., Tietoturvallisuus ja laki. Näkökohtia tietoturvallisuuden oikeudellisesta sääntelystä. *Valtionvarainministeriö ja Lapin yliopiston oikeusinformatiikan instituutti. Hallinnon kehittämissosasto.*(1997).
- ScE03 Scherlis, W., Eisenberg, J., IT Research, Innovation and E-Government. *Communications of the ACM*, Vol. 46, No. 1 (2003), sivut 67-68.

- Yht04 Yhteispalvelu, sisäasianministeriön sivusto.  
<http://www.yhteispalvelu.fi> [21.2.2004].
- Tag01 Tagg, R., Workflow in Different Styles of Virtual Enterprise, *Proceedings of the Workshop on Information Technology for Virtual Enterprises* (2001), sivut 21-28.
- Tam01a Tambouris, E., An Integrated Platform for Realising Online One-Stop Government: The eGOV Project. *12th International Workshop on Database and Expert Systems Applications* (2001), sivut 359-363.
- Tam01b Tambouris, E., (ed.), An Integrated Platform for Realising Online One-Stop Government (eGOV), Project Presentation and State-of-the-Art, D011 (2001).
- TaS02 Tanenbaum, A., van Steen, M., *Distributed Systems, Principles and Paradigms*, Prentice Hall (2002).
- TFS01 Toyouchi, J., Funabashi, M., Strick, L., Born, M., Kanai, A., Uchihashi, T., Hakomori, S., Yoshida, E., Komoda, N., Development of Service Integration Platform for One-stop Service Applications. *Third International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS '01)* (2001), sivut 123-125.
- TrW01 Traummüller, R., Wimmer, M., Directions in E-Government: Processes, Portals, Knowledge. *Database and Expert Systems Applications. Proceedings. 12<sup>th</sup> International Workshop* (2001), sivut 313-317.
- TSK02 Tambouris, E., Spanos, E., Kavadias, G., (ed.), An Integrated Platform for Realising Online One-Stop Government (eGOV): Services and Process models functional specifications, D121 (2002).
- Udd02 UDDI Spec TD: UDDI Version 3.0, UDDI Spec Technical Committee Specification, 19 July 2002. <http://uddi.org/pubs/uddi-v3.00-published-20020719.htm> [1.4.2004].

- VGL03 Verginadis, G., Gouscos, D., Legal, M., Mentzas, G., An Architecture for Integrating Heterogeneous Administrative Services into one-stop e-Government, *eChallenges 2003 Conference, Bologna, October 22-24 (2003)* [http://www.cb-business.com/documents/eChallenges\\_paper\\_published.pdf](http://www.cb-business.com/documents/eChallenges_paper_published.pdf) [5.5.2004].
- ViM02 Virili, F., Sorrentino, M., Reconfiguring the Political Value Chain: The Potential Role of Web Services. *Lecture Notes in Computer Science*, 2456, (2002), s. 61-68.
- W3C01 Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001. <http://www.w3.org/TR/wsdl> [1.4.2004].
- W3C03 SOAP Version 1.2 Part 2: Adjuncts, W3C Recommendation 24 June 2003. <http://www.w3.org/TR/2003/REC-soap12-part2-20030624/> [1.4.2004].
- W3C04a Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation 04 February 2004. <http://www.w3.org/TR/2004/REC-xml-20040204> [31.3.2004].
- W3C04b Web Services Architecture, W3C Working Group Note 11 February 2004. <http://www.w3.org/TR/ws-arch/> [1.4.2004].
- Web03 Web Services for Remote Portlets Specification, Version 1.0, OASIS (2003). <http://www.oasis-open.org/committees/download.php/3343/oasis-200304-wsrp-specification-1.0.pdf> [14.10.2004].
- Wes04 West, D., E-Government and the Transformation of Service Delivery and Citizen Attitudes. *Public Administration Review*, 64 (2004), sivut 15-27.
- WiT00 Wimmer, M., Traunmüller, R., Trends in Electronic Government: Managing Distributed Knowledge. *Database and Expert Systems Applications. Proceedings. 11<sup>th</sup> International Workshop*. (2000), sivut 340-345.