

# **Integraatoratkaisu joukkoviestintäverkkojen esittämiseen paikkatietojärjestelmässä**

Tuomas Suni

## **Sähkötekniikan korkeakoulu**

Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi  
diplomi-insinöörin tutkintoa varten Espoossa 12.4.2016.

**Työn valvoja:**

Prof. Jukka Manner

**Työn ohjaaja:**

DI Heikki Isotalo

Tekijä: Tuomas Suni		
Työn nimi: Integraatoratkaisu joukkoviestintäverkkojen esittämiseen paikkatietojärjestelmässä		
Päivämäärä: 12.4.2016	Kieli: Suomi	Sivumäärä: 8+104
Tietoliikenne- ja tietoverkkotekniikan laitos		
Professuuri: Tietoverkkotekniikka		
Työn valvoja: Prof. Jukka Manner		
Työn ohjaaja: DI Heikki Isotalo		
<p>Paikkatiedon ja paikkatietojärjestelmien käyttö on laajentunut ammattilaiskäyttöä osaksi joka paikan tietotekniikkaa. Paikkatiedon yleistymistä ovat vauhdittaneet Open Geospatial Consortiumin määrittelemät avoimet standardit paikkatiedon tallentamisessa ja jakamisessa. Toisaalta internet-sovelluksista on tullut yhä enemmän hajautettuja järjestelmiä, jotka hyödyntävät monimuotoisia tietolähteitä ja palveluita. Internet-sovellusten hajautettavuutta on edistänyt World Wide Web Consortium erilaisilla XML- ja Web Service -standardeillaan.</p> <p>Tässä työssä tehdään tietoliikenneyritys Digita Oy:lle paikkatietojärjestelmän vaatimusmäärittely. Sen pohjalta suunnitellaan sitä tukeva ohjelmistoarkkitehtuuri ja toteutetaan taustajärjestelmäratkaisu, joka hyödyntää tietolähteinä yrityksen olemassa olevia ohjelmointirajapintoja, tietojärjestelmiä ja aineistoja mahdollisimman tehokkaasti.</p> <p>Työn tuloksena määriteltiin televisio- ja radiopalveluiden kuluttajille sovellus, joka visualisoi palveluiden saatavuuden karttaesityksenä sekä tiedottaa häiriö- ja huoltotilanteesta alueellisesti ja reaaliaikaisesti. Toteutettu taustajärjestelmäratkaisu rakennettiin palvelukeskeisen arkkitehtuurimallin mukaisesti. Arkkitehtuurin ytimenä toimii palveluväyläsovellus, joka hyödyntää tässä työssä toteutettuja rajapintapalveluita sekä reitittää ja muuntaa tietoa kohdejärjestelmään soveltuvaksi. Ratkaisu vastasi erinomaisesti kaikkiin sille asetettuihin vaatimuksiin. Taustajärjestelmän julkiset rajapinnat suunniteltiin uudelleenkäytettäväksi myös mahdollisissa tulevaisissa sovelluksissa, mutta taustajärjestelmäratkaisun suorituskyvyn ja tiedon reaaliaikaisuuden kanssa voi tulla eteen haasteita. REST-arkkitehtuurin hyödyntäminen ja koordinaattimuunnosten määrän pienentäminen voisivat vaikuttaa suorituskykyhaasteisiin edullisesti.</p>		
Avainsanat: paikkatieto, gis, palvelukeskeinen arkkitehtuuri, integraatio, palveluväylä, esb, eai, väliohjelmisto, soa, xml, web service, soap, wsdl, xml-skeema, xquery, xpath		

Author: Tuomas Suni
Title: Integration solution for visualizing mass communications networks on geographic information system
Date: 12.4.2016                      Language: Finnish                      Number of pages: 8+104
Department of Communications and Networking
Professorship: Networking technology
Supervisor: Prof. Jukka Manner
Advisor: M.Sc. Heikki Isotalo
<p>The use of geospatial data and geographic information systems has been extended from professional use to the part of ubiquitous computing. Open Geospatial Consortium has defined several open standards for storing and sharing geospatial data, which has promoted the development and utilization of geographic data and systems. At the same time, internet applications are becoming distributed systems that exploit diverse data sources and services. Different XML and Web Service standards defined by World Wide Web Consortium have advanced the development of distributed systems.</p> <p>In this master's thesis a requirement specification is performed for a geographic information system of telecommunication company Digita Oy. Based on requirement specification, supporting software architecture is designed and backend system is implemented. This system utilizes Digita's existing application programming interfaces, information systems and data as a data source.</p> <p>The result of this study was the requirement specification for an application that is directed to consumers of television and radio services. The application visualizes availability of the services as well as outages and maintenances in the network geographically in real time. Designed architecture of the backend system was based on the model of service-oriented architecture. A core of the architecture is service bus application, which uses the services implemented in this study, route messages and transform data to a schema of a target system. The solution fulfilled the requirements that were set very well. Interface of the backend system was planned for reusing for future applications, but there might be some challenges in performance and real time data. REST architecture and minimizing the number of coordinate transforms could reduce the probability of performance issues.</p>
Keywords: geospatial data, gis, service-oriented architecture, integration, enterprise service bus, esb, eai, middleware, soa, xml, web service, soap, wsdl, xml schema, xquery, xpath

## Esipuhe

Diplomityön kirjoittaminen lähes kaksi vuosikymmentä opintojen aloittamisen jälkeen oli erittäin kasvattava kokemus. En suosittelen sitä muille, mikäli se nykyisessä opiskeluaikojen lyhentämiseen tähtäävässä ilmapiirissä on edes mahdollista.

Haluan kiittää valvojaani professori Jukka Mannerta täydestä luottamuksesta työtäni kohtaan. Riittävän vapaat kädet ja rohkaisevat sanat epävarmuuden hetkillä olivat juuri se mitä tarvitsin. Kiitän ohjaajaani Heikki Isotaloa mahdollisuudesta toimia tässä Digita Oy:n projektissa projektipäällikkönä ja sitä kautta erittäin mielenkiintoisesta työn aiheesta. Haluan myös kiittää projektiryhmästäni Timo Hakulista, Anna-Riitta Lundia ja Matti Jokisaloa, joiden panos tässä työssä oli ehdottoman tärkeä. Lämpimimmät kiitokset kuuluvat kuitenkin avovaimolleni Veralle, joka suhtautui äärimmäisen ymmärtäväisesti työn pitkittyneeseen kirjoitusprosessiin.

Helsinki, 4.4.2016

Tuomas Suni

# Sisällysluettelo

<b>Tiivistelmä</b>	<b>ii</b>
<b>Tiivistelmä (englanniksi)</b>	<b>iii</b>
<b>Esipuhe</b>	<b>iv</b>
<b>Sisällysluettelo</b>	<b>v</b>
<b>Symbolit ja lyhenteet</b>	<b>viii</b>
<b>1 Johdanto</b>	<b>1</b>
<b>2 Maanpäälliset joukkoviestintäverkot ja -palvelut</b>	<b>4</b>
2.1 Digitaalinen televisio Suomessa . . . . .	4
2.2 Analoginen radio Suomessa . . . . .	4
2.3 Signaalin eteneminen . . . . .	5
2.4 Yhteenveto maanpäällisistä joukkoviestintäverkoista . . . . .	7
<b>3 Paikkatieto ja paikkatietojärjestelmät</b>	<b>8</b>
3.1 Paikkatiedon historia . . . . .	8
3.2 Paikkatietojärjestelmät . . . . .	9
3.3 Tietoliikennetekniikan paikkatietojärjestelmät . . . . .	10
3.4 Paikkatietoteknologia . . . . .	12
3.4.1 Paikkatieto ja sen muunnokset . . . . .	13
3.4.2 Extensible Markup Language (XML) . . . . .	13
3.4.3 Paikkatietojärjestelmien tietomuodot . . . . .	14
3.4.4 Geographic Markup Language (GML) . . . . .	14
3.4.5 Web Map Service (WMS) . . . . .	16
3.4.6 Web Feature Service (WFS) . . . . .	16
3.5 Yhteenveto paikkatiedosta . . . . .	16
<b>4 Hajautettujen järjestelmien arkkitehtuurimallit</b>	<b>17</b>
4.1 Arkkitehtuuri . . . . .	17
4.2 Palvelukeskeinen arkkitehtuuri . . . . .	18
4.2.1 Palvelu . . . . .	22
4.2.2 Soveltaminen paikkatietojärjestelmissä . . . . .	23
4.3 Representational State Transfer (REST) . . . . .	25
4.4 Mashup . . . . .	26
4.5 Yhteenveto hajautettujen järjestelmien arkkitehtuurista . . . . .	27
<b>5 Web Services -teknologiat</b>	<b>28</b>
5.1 XML-standardit . . . . .	28
5.1.1 XML Schema Definition Language (XSD) . . . . .	28
5.1.2 XPath . . . . .	29
5.1.3 XQuery . . . . .	30

5.2	Protokollat	31
5.2.1	Web Services (WS)	31
5.2.2	Simple Object Access Protocol (SOAP)	32
5.2.3	Web Service Description Language (WSDL)	33
5.2.4	Universal Description Discovery and Integration (UDDI)	34
5.3	Yhteenvedo Web Services -teknologioista	34
<b>6</b>	<b>Järjestelmäintegraatiot</b>	<b>36</b>
6.1	Enterprise Application Integration (EAI)	36
6.2	Väliohjelmistot	38
6.3	Palveluväylä	40
6.4	Yhteenvedo järjestelmäintegraatioista	41
<b>7</b>	<b>Karttapalvelusovelluksen vaatimusmäärittely</b>	<b>43</b>
7.1	Johdanto	43
7.2	Yleiskuvaus	43
7.3	Tiedot ja tietokanta	45
7.4	Toiminnot	45
7.4.1	Sijainnissa vastaanotettavissa olevat palvelut	45
7.4.2	Suuntaa vastaanottoantenni	48
7.4.3	Etsi paikkakunnalla toimiva antenniasentajayritys	49
7.4.4	Selvitä lähetyksessä oleva häiriö	49
7.4.5	Palveluiden peittoalueet	49
7.4.6	Digitan kuluttajapalvelu	50
7.4.7	Taajuustalkoiden neuvontapalvelu	51
7.4.8	Asiakkaan tai yhteistyökumppanin oma karttapalvelu	51
7.5	Ulkoiset liittymät	52
7.5.1	Laitteistoliittymät	52
7.5.2	Tietoliikenneliittymät	52
7.5.3	Ohjelmistoliittymät	53
7.6	Muut ominaisuudet	54
7.7	Yhteenvedo karttapalvelusovelluksen vaatimusmäärittelystä	55
<b>8</b>	<b>Karttapalveluratkaisu</b>	<b>56</b>
8.1	Taustajärjestelmät ja niiden rajapinnat	56
8.1.1	TIHA	56
8.1.2	PAJA	58
8.1.3	HP Service Manager	61
8.1.4	Microsoft Exchange Server	63
8.1.5	Palveluiden peittoalueet	64
8.1.6	Paikkatietokanta	66
8.1.7	GeoServer	67
8.2	Järjestelmäarkkitehtuuri	68
8.3	Järjestelmän sisäiset integraatiot	69
8.4	Ulkoiset rajapinnat	70

8.5	Yhteenveto karttapalveluratkaisusta . . . . .	75
<b>9</b>	<b>Tulokset ja johtopäätökset</b>	<b>75</b>
9.1	Vertailu vaatimusmäärittelyyn . . . . .	75
9.2	Julkinen ohjelmointirajapinta . . . . .	76
9.3	Paikkatiedon käsittely . . . . .	78
9.4	Arkkitehtuuri . . . . .	80
<b>10</b>	<b>Yhteenveto</b>	<b>83</b>
<b>A</b>	<b>Digitan karttapalvelun Web Services -rajapinnat</b>	<b>90</b>

# Symbolit ja lyhenteet

## Lyhenteet

AM/FM/GIS	Automated Mapping/Facilities Management
API	Application programming interface
BPM	Business Process Model
BPMN	Business Process Modeling Notation
BPMS	Business Process Management System
BPEL	Business Process Execution Language
CIA	Coordinator Integration Adapter
CSV	Comma-Separated Value
DDL	Data Definition Language
DTD	Document Type Definition
EAI	Enterprise Application Integration
ESB	Enterprise Service Bus
EWS	Exchange Web Service
GDF	Geographic Data Files
GIF	Graphic Interchange Format
GIS	Geographic Information System
GML	Geography Markup Language
GPS	Global Positioning System
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JCA	Java Connector Architecture
JMS	Java Message Service
JPEG	Joint Photographic Experts Group
MD	Message Drive
MOM	Message-Oriented Middleware
OGC	Open Geospatial Consortium
PNG	Portable Network Graphics
QoS	Quality of Service
REST	Representational State Transfer
RIA	Rich Internet Application
RPC	Remote Procedure Call
SCA	Service Component Architecture
SOA	Service Oriented Architecture
SOR	System of Record
SOAP	Simple Object Access Protocol
TIFF	Tagged Image File Format
UDDI	Universal Description Discovery and Integration
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
WFS	Web Feature Service
WMS	Web Map Service
WPS	Web Processing Service
WSCI	Web Service Choreography Interface
WWW	World Wide Web
XML	Extensible Markup Language
XSD	XML Schema Definition Language



# 1 Johdanto

Kuluttajille tarjottavat televisio- ja radiopalvelut maanpäällisessä joukkoviestintä-verkoissa ovat vahvasti tiettyyn sijaintiin sidottuja palveluita. Digitaalisten televisiopalvelujen ja analogisten radiopalvelujen vastaanotto on mahdollista kaikkialla Suomessa, mutta palvelutarjonta poikkeaa monilta osin paikkakuntakohtaisesti. Televisiopalveluissa kanavatarjonta vaihtelee alueellisesti, signaalin vastaanotossa antenni on suunnattava lähetyksentantennin kohden ja kanavien lähetyksenaajuus vaihtelee lähettimittäin. Radiopalveluissa paikkakuntaisuus on vielä suurempaa kuin televisiopalveluissa, koska vain muutaman valtakunnallisen radiokanavan lisäksi on tarjolla suuri joukko paikallisia radiokanavia.

Joukkoviestintäpalveluiden saatavuuden kuvaaminen kuluttajille on hankalaa, koska palvelu näyttäytyy jokaisessa sijainnissa hieman erilaisena. Yksi tehokas tapa kuvata joukkoviestintäpalveluiden saatavuutta on visualisoida televisiokanavien näkyvyysalueet ja radiokanavien kuuluvuusalueet karttapohjalla tarjoten näin kuluttajille työkalun selvittää joukkoviestintäpalveluiden saatavuutta eri sijainneissa. Lisäksi karttaesityksessä voidaan tarjota televisiolähettimien sijaintitiedot vastaanottoantennin suuntaamista varten. Mikäli joukkoviestintäpalvelutarjonnan visualisointi on reaaliaikainen, voidaan palveluiden saatavuuden lisäksi esittää verkon tila. Verkon tilalla voidaan ilmaista mahdolliset häiriöt ja tulevat huollot joukkoviestintäpalveluissa, ja informoida mikä on niiden palveluvaikutus ja mitä maantieteellistä aluetta ne koskevat. Viestintävirasto Suomessa on velvoittanut kaikkia teleyrityksiä, myös joukkoviestintäpalveluja tarjoavia, tiedottamaan verkoissaan olevista häiriöistä internetissä karttaesityksenä.

Joukkoviestintäpalveluiden maantieteellinen näkyvyysalue, kuuluvuusalue tai radiolähettimen sijainti on paikkatietoa. Paikkatieto on tietoa, jolle voidaan osoittaa tietty maantieteellinen sijainti. Sijainnin lisäksi paikkatieto pitää sisällään ominaisuustietoa, joka kertoo mitä tiettyssä sijainnissa on. Paikkatietoa tyypillisesti jaetaan, käsitellään tai kerätään paikkatietojärjestelmään (GIS). Paikkatietojärjestelmä on siten myös luonnollinen valinta maanpäällisen joukkoviestintäverkon palveluiden ja muun tarvittavan tiedon esittämiseen.

Tietoliikennealan yrityksen tavoitteena oli kehittää ensisijaisesti kuluttajille tarjottava paikkatietojärjestelmä, joka kuvaa karttaesityksenä joukkoviestintäpalveluiden saatavuuden Suomessa. Paikkatietojärjestelmän piti myös pystyä tiedottamaan verkon häiriöistä Viestintäviraston asettamien määräysten mukaisesti, sekä tarjota tietoa tulevista ja käynnissä olevista huolloista. Verkon huoltotöillä voi olla usein vaikutusta kuluttajan kokemaan palvelutasoon. Lisäksi järjestelmään haluttiin viellä muutakin joukkoviestintäpalveluihin liittyvää paikkatietoa, jonka nähtiin olevan arvokasta kuluttajille, asiakkaille tai yritykselle itselleen.

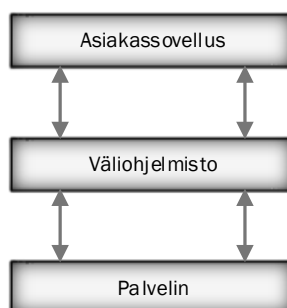
Paikkatiedon ja ominaisuustietojen välittäminen paikkatietosovellukselle koettiin tietoliikenneyrityksessä haasteellisena, koska kaikki tieto oli hyvin heterogeenista sijaiten eri tietojärjestelmissä, muodoissa ja koordinaattijärjestelmissä tai tieto ei ollut ensinnäkään paikkatiedoksi rinnastettavassa muodossa.

Tässä työssä selvitettiin mitä tietoa paikkatietosovellukselle tulisi välittää, suunniteltiin integraatioarkkitehtuuri rajapintoineen ja toteutettiin tarvittavat rajapin-

nat ja integraatiopalvelut tietoliikenneyrityksen tietojärjestelmistä ulkoiseen web-paikkatietosovellukseen. Integraatio toteutettiin palvelukeskeiseen arkkitehtuuriin (SOA) perustuen rakentamalla joukko erilaisia palveluita, joilla paikka- ja ominaisuustietoa voidaan hakea tietojärjestelmistä, yhdenmukaistaa ja rikastaa sekä lopulta tarjota avoimien standardien rajapintojen kautta web-sovellukselle. Toisin sanoen, tässä työssä suunniteltiin ja toteutettiin kokonaisvaltaisesti web-sovellukselle hajautettu taustajärjestelmäratkaisu.

Integraatoratkaisun komponenttien vuorovaikutus perustuu World Wide Web Consortiumin (W3C) määrittelemään Web Services -ohjelmistojärjestelmään, joilla toteutetaan palvelukeskeiseen arkkitehtuuriin perustuvia ratkaisuja. Vektorimuotoisen paikkatiedon välittämisessä järjestelmien välillä käytetään Open Geospatial Consortiumin (OGC) standardeihin, esimerkiksi Web Feature Service (WFS) -rajapintamäärittely. WFS:ää käyttäen voidaan tietoverkon kautta kysellä paikkatietokohteiden koordinaatteja ja tietoja. Tässä työssä paikkatietokohteita olivat esimerkiksi televisiopalveluiden näkyvyysalueet.

Tiedon välitys lähdejärjestelmistä paikkatietosovellukseen perustuu siis yleisesti tunnettuihin internet-standardeihin, jotka toimivat tietoverkossa synkronisen pyyntö-vastausmallin mukaisesti. Tyypillisesti tällaista dialogia hyödynnetään kutsuvan asiakassovelluksen ja vastaavan palvelinsovelluksen välillä kaksikerroksisessa järjestelmäarkkitehtuurissa. Tässä työssä malli on laajennettu sisältämään myös väliohjelmistokerroksen, joka vastaa kaikista sisältöön liittyvistä sanomamuunnoksista ja sanoman reitityksistä. Muunnokset voivat liittyä esimerkiksi tiedon XML-skeemamuunnokseen, perinteisen tiedon muuttamista standardin mukaiseksi paikkatiedoksi tai paikkatiedon muuttamista koordinaatistojärjestelmästä toiseen.



Kuva 1: Kolmikerroksinen palveluarkkitehtuuri.

Yhteenvedona voidaan siis todeta, että työssä hyödynnetään palvelukeskeisen arkkitehtuurin ja paikkatietoteknologian tarjoamia käytäntöjä, suosituksia ja standardeja, joiden pohjalta suunnitellaan ja toteutetaan toimiva integraatoratkaisu tietoliikenneyrityksen tietojärjestelmien ja web-karttapalvelusovelluksen välille.

Työn lopullisena tuotoksena syntyi siis tuotantokelpoinen hajautettu paikkatietojärjestelmä Digita Oy:lle, joka täytti sille asetetut toiminnalliset ja ei-toiminnalliset vaatimukset. Työn suurin haaste liittyi yrityksen olemassa olevien heterogeenisten

tietojärjestelmien integroimiseen osaksi järjestelmäratkaisua. Lisäksi työssä toteutettiin paikkatietojärjestelmän vaatimat integraatio- ja rajapintapalvelut, joiden toteutus tehtiin palvelukeskeisen arkkitehtuurin periaatteiden ja käytäntöjen mukaisesti. Tämän työn laajuuteen ei sisällynyt järjestelmän käyttöliittymäsovelluksen suunnittelu ja toteutus, vaan kokonaisuudesta vastasi ulkoinen taho, joka hyödynsi tämän työn tuottamaa vaatimusmäärittelyä ja taustajärjestelmän ohjelmointirajapintaa. Tuotettua järjestelmäratkaisua arvioitiin palvelukeskeisen arkkitehtuuriin tavoitteiden ja mahdollisten tulevien tarpeiden pohjalta. Hajautetun järjestelmän jatkokehityksessä pitäisi paremmin huomioida kaikkien rajapinta- ja integraatiopalveluiden uudelleenkäytettävyyttä, jotta tehtyä työtä ja investointeja voitaisiin hyödyntää tehokkaammin. Lisäksi havaittiin, että tässä työssä tuotetun ulkoisen rajapinnan olisi syytä olla paremmin itsensä kuvaava, jotta sen käyttö ulkopuolisissa sovelluksissa olisi yksinkertaisempaa ja houkuttelevampaa. Tulevia tarpeita ajatellen rajapinnan pitäisi pystyä paremmin palvelemaan reaaliaikaisia sovelluksia takaamalla riittävä kevyet siirtoprotokollat ja tietomuodot sekä rajapinnan riittävän suorituskyvyn.

Työn toisessa luvussa esitellään työn liiketoiminnallinen viitekehys – joukko- viestintäverkot ja -palvelut. Luvussa tuodaan esille liiketoiminnan erityispiirteitä, joihin järjestelmäratkaisun on mukauduttava. Liiketoiminnan pohjalta syntyy web-paikkatietosovelluksen vaatimusmäärittely ja sitä kautta vaatimukset koko hajautetulle järjestelmälle. Kolmannessa luvussa käsitellään paikkatietojärjestelmiä, paikkatietoa ja siihen liittyviä standardeja. Neljännessä luvussa käydään läpi erilaisia hajautettujen järjestelmien arkkitehtuurimalleja, joihin perustuen tämän työn ratkaisun voisi toteuttaa. Viidennessä luvussa siirrytään arkkitehtuuritasolta teknisemmälle tasolle esittelemällä työssä hyödynnettyjä teknologioita. Pääosissa on erilaiset XML-standardit, sekä palvelukeskeisessä arkkitehtuurissa tyypillisesti hyödynnetyt teknologiat. Työn kuudennessa luvussa esitellään järjestelmäintegraatioiden kehityspolkua ja erilaisia integraatiomalleja.

Työn seitsemännessä luvussa siirrytään teoreettisen viitekehysten alta toteutettuun ratkaisuun. Tämä luku pitää sisällään kuvauksen järjestelmän vaatimusmäärittelystä, jossa on erityisesti keskitytty toiminnallisiin vaatimuksiin, joiden vaikutus on heijastunut hajautetun järjestelmän muihin komponentteihin. Työn kahdeksannessa luvussa kuvataan toteutettu hajautettu taustajärjestelmä kokonaisuudessaan. Siinä kuvataan ratkaisun järjestelmäkomponentit, niiden rajapinnat ja vuorovaikutussuhteet sekä teknologiavalinnat.

Yhdeksännessä luvussa arvioidaan toteutettua paikkatietojärjestelmää vertailemalla sitä aiemmin tehtyyn vaatimusmäärittelyyn. Lisäksi arvioidaan tehtyjä suunnitteluvalintoja liittyen valittuihin arkkitehtuuriin ratkaisuihin sekä toteutettuihin rajapintoihin ja integraatioihin. Viimeisessä kymmenennessä luvussa on lyhyt yhteenveto koko työstä.

## 2 Maanpäälliset joukkoviestintäverkot ja -palvelut

Tämän luvun tarkoituksena on kuvata maanpäällisten joukkoviestintäverkkoihin liittyvää toimialaa ja tietoliikennepalveluita Suomessa. Luvussa keskitytään digitaalisen antennitelevisiopalveluun ja analogisen radiopalveluun sekä näiden palveluiden signaalien etenemiseen liittyvään teoriaan. Luvussa alustetaan niitä vaatimuksia ja ominaispiirteitä, joita maanpäällinen joukkoviestintäverkko asettaa sitä tukevalle karttapalvelusovellukselle.

### 2.1 Digitaalinen televisio Suomessa

Verkkopalvelun tarjoamiseen Suomessa antennitelevisioverkossa vaaditaan verkkotoimilupa [1]. Kaapelitelevisiotoimintaan sekä satelliittivälitteiseen televisiotoimintaan ei lupaa tarvita. Television verkkotoimiluvat Suomessa on myönnetty Digita Oy:lle, DNA Oy:lle sekä Anvia Oy:lle 31.12.2016 saakka. Digitalle toimiluvat on myönnetty kanavanippuihin A, B, C, D, E ja H. DNA:n kanavaniput ovat VHF A, VHF B ja VHF C ja Anvian kanavaniput ovat F ja G. Kaikki ohjelmakanavat eivät näy koko maassa. Toimiluvissa on selkeät vaatimukset sille, kuinka suurelle osalle väestöstä kukin televisiokanava on vähintään jaettava. Käytännössä näkyvyysalueet ovat kuitenkin vähimmäisvaatimuksia kattavammat. [2] Lisäksi jokainen lähetin tarvitsee vielä erikseen Viestintäviraston myöntämän radioluvan, jossa määritellään lähetyslaitteiden tekniset ominaisuudet. Valtakunnalliseen televisiotoimintaan on Suomessa varattu VHF-alueelta kanavat 5-12 (174-230 MHz) ja UHF-alueelta kanavat 21-60 (470-790 MHz). [1]

Digitaalinen televisio (DVB, Digital Video Broadcasting) tarkoittaa digitaaliseen jakeluun perustuvaa televisiojärjestelmää. DVB on standardoitu erikseen antennijakeluun eli maanpäälliseen jakeluun (DVB-T), kaapelijakeluun (DVB-C) ja satelliittijakeluun (DVB-S). Digitaalisessa televisiojaketussa ohjelmakanavia yhdistellään kanavanipuiksi eli mukseiksi. Jakelustandardeista on tehty uusia versioita. Antennijakelussa on Suomessa jo käytössä DVB-T2 ja sen käyttö laajenee jatkuvasti. Standardien päivittämisellä on saatu tehostettua televisioverkkojen jakelukapasiteettia. Uusilla tekniikoilla saadaan samalla kanavanippumäärällä jaettava enemmän sisältöä tai parannettua äänen tai kuvan laatua. [1]

### 2.2 Analoginen radio Suomessa

Radiotoiminta edellyttää lyhytaikaista ja pienimuotoista toimintaa lukuun ottamatta ohjelmistotoimilupaa. Lisäksi radiolähettimellä pitää olla radiolupa, jossa määritellään lähetyksessä käytettävät tekniset parametrit. Suurin osa suomalaisista radiotoimijoista on paikallisia. Ainoastaan Yleisradion ja Radio Novan kuuluvuusalueet kattavat lähes koko väestön. Koko maan jakelu perustuu Digita Oy:n omistamiin korkeisiin mastoihin ja niissä oleviin lähetysantenneihin sekä suuriin lähetystehoihin. [1]

Kaupallinen radiotoiminta käynnistyi Suomessa 1985. Viime vuosien aikana radioasemien määrä on ollut edelleen kasvussa. Radiotoiminta Suomessa perustuu

ULA-taajuuksien ja FM-modulaation käyttöön. Radion kuuluvuusalueiden laskennassa oletetaan vastaanoton tapahtuvan lähetysasemaa kohti suunnatulla, katolle asennetulla antennilla. Lähes vastaava kuuluvuus on myös kannettavilla radioilla ja autovastaanottimilla, vaikka ovatkin herkempiä maastoesteistä ja heijastuksista johtuville häiriöille. Radio voi myös kuulua hyvin kaukana varsinaisen kuuluvuusalueen ulkopuolella. [1]

### 2.3 Signaalin eteneminen

Signaalin eteneminen noudattaa perustapauksessa yksinkertaista geometristä mallia. Signaalin etenemismalli ei kuitenkaan noudata näitä oletuksia silloin, kun signaali kohtaa esteitä, erilaisia pinnanmuotoja ja muita tekijöitä, joihin käytännössä usein törmätään. Yksinkertaisimmassa signaalin etenemistapauksessa lähetetään muuttumatonta säteilyä pistemäisestä lähteestä eli isotrooppisesta säteilijästä. Tämä voidaan nähdä analogisena pistemäiselle valon lähteelle, jossa säteilevä energia välittyy tasaisella intensiteetillä imaginäärisen pallon pinnan läpi, joka sijaitsee säteen  $r$  etäisyydellä lähteestä. Tällaisen pinnan ala on  $4\pi r^2$  ja teho pinta-alayksikköä kohden  $W = P_t/4\pi r^2$ , missä  $P_t$  on lähteen säteilemä teho. Lähetystekniikassa ja radiopalveluissa on kuitenkin tavanomaisempaa puhua tehotehiheyden  $W$  sijaan sähkökentän  $E_0$  voimakkuudesta. Tehotehiheys vapaassa tilassa on yhtä suuri kuin kentän voimakkuuden neliö jaettuna väliaineen impedanssilla, eli [3]

$$W = \frac{E_0^2}{120\pi}, \quad (1)$$

jolloin

$$P_t = \frac{r^2 E_0^2}{30}. \quad (2)$$

Tavanomaisin muoto tästä yhtälöstä on sovellettavissa myös muihinkin lähetysantenneihin kuin isotrooppiseen säteilijään, joka on

$$E_0 = \frac{\sqrt{30g_t P_t}}{r}, \quad (3)$$

jossa  $g_t$  on tehovahvistus suunnatulle antennille verrattuna isotrooppiseen antenniin. Isotrooppinen antenni on käyttökelpoinen vertailumalli monimutkaisimmillekin antenneille, vaikka ideaalista isotrooppista antennia ei todellisuudessa ole olemassa. [3]

Maanpinta muuttaa radioaaltojen etenemistä siten, että vastaanotettu kentän voimakkuus on yleensä erilainen kuin se olisi ollut vapaassa tilassa. Maanpinta heijastaa ja absorboi osittain radioaaltoja ja kumpikin ominaisuus vaikuttaa energian jakautumiseen maan pinnan yläpuolella. Vastaanottoantenni vastaanottaa radioaallot suoraa polkua pitkin sekä maanpinnan kautta heijastuneen radioaallon. Näiden vastaanotettujen radioaaltojen välillä on vaihe-ero, johon vaikuttaa lähettävän ja vastaanottavan antennin välinen etäisyys sekä niiden korkeudet maanpinnasta. Maanpinnalla tapahtuva radioaallon vaimeneminen riippuu taajuudesta, antennipolarisaation tyypistä sekä maanpinnan vakioarvosta. [3]

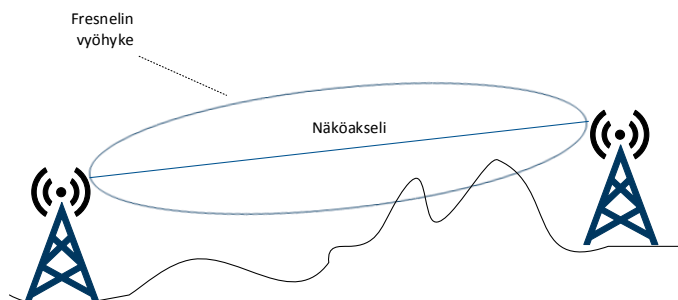
Maanpinnan kaarevuus vaikuttaa radioaaltojen etenemiseen silloinkin kun lähetys- ja vastaanottoantennit ovat näköakselilla. Ensimmäiseksi, kaarevan pinnan heijastuskerroin on erilainen kuin tasaisen pinnan kerroin. Toiseksi, energia hajaantuu voimakkaammin kaarevalla pinnalla kuin etäisyyden käänteiseen neliöjuureen perustuva laki osoittaa. Kolmanneksi, maan kaarevuus aiheuttaa sen, että heijastuspisteeseen piirretty tangenttitaso leikkaa lähetys- ja vastaanottoantennit, jolloin tasolta mitattu antennin korkeus on aina pienempi kuin antennin todellinen korkeus. [3]

Radioaallot etenevät maan pinnalla myös näköakselin ulkopuolelle. Tämä ilmiö perustuu aaltoliikkeen diffraktioon. Diffraktio on aaltoliikkeen perusominaisuus. Optiikassa diffraktio täydentää geometrista optiikkaa eli sädeoptiikkaa, jolloin lopputuloksena on täsmällisempää aalto-optiikkaa. Aalto-optiikassa jokainen piste aaltorintamassa toimii säteilevänä lähteenä. Kun aaltorintama on yhdenmukainen ja häiriötön, tuloksena on rintamaa kohtisuorassa olevaa polkua etenevä säde. Kun aaltorintamaa häiritään, seuraa siitä rintaman voimakkuuden ja suunnan muutoksia, jolloin säde voi vaimentua ja taittua. [3]

Kaareutuvan maan pinnalla lähettäminen näköakselin ulkopuolelle on mahdollista diffraktion avulla. Tämä kuitenkin aiheuttaa suurempia häviöitä kuin lähettäminen vapaassa tilassa tai tasaisella pinnalla. Häviön määrä lisääntyy lähetysetaisyyden ja taajuuden kasvaessa ja ne riippuvat jossain määrin antennin korkeudesta. Kentän voimakkuuden laskeminen tietyssä pisteessä näköakselin ulkopuolella on monimutkaista, eikä sitä yleensä tehdä kuin tietyillä sitä varten suunnitelluilla tietokoneohjelmistoilla. Näissä malleissa oletetaan, että maa on täysin tasainen pallo, jossa kentänvoimakkuus voidaan laskea erikseen näköakselin sisä- ja ulkopuolella ja interpoloimalla kentänvoimakkuus näiden välillä. Tällainen tarkastelu ei ota huomioon mäkiä, rakennuksia tai puita, joilla on niin monimutkainen vaikutus aaltoliikkeen etenemiseen, että on lähes mahdotonta laskea kentänvoimakkuus tietyssä pisteessä tai edes mediaania tietyllä pienellä alueella. Kuitenkin analysoimalla maanpinnan profilia aaltoliikkeen etenemää polkua pitkin ja tekemällä tiettyjä yksinkertaistuksia, ennustukset ovat täsmällisempiä kuin pelkästään tasaisen maanpinnan mukaan tehdyt laskutoimitukset. [3]

Radioaaltojen eteneminen terävän harjun tai mäen yli voidaan käsitellä veitsenterädiffraktiona. Häviön määrä lisääntyy sitä enemmän, mitä korkeampi tällainen este on lähetysantennista vastaanottoantenniin kulkevan näköakselin yläpuolella. Häviön määrä lähestyy nolaa, kun suora näköakseli on vapaa. Vapaalle välille on myös olemassa optimi, jota kutsutaan 1. Fresnelin vyöhykkeeksi. Siinä vapaata tilaa on suoraa näköyhteyttä enemmän, jolloin teoreettisesti voidaan saavuttaa 1,2 dB parempi siirto kuin vapaassa tilassa. 1. Fresnelin vyöhykkeen tila määräytyy siten, että aaltoliikkeen vaihesiirtymä lähetysantennista esteen huipulle ja sieltä vastaanotajalle on noin puolitoista aallonpituutta suurempi kuin vaihesiirtymä suoraa polkua pitkin. [3] Kuvassa 2 on esitetty tilanne, jossa antennien välillä on suora näköyhteys, mutta 1. Fresnelin vyöhykkeellä on esteitä. Tällöin ei saavuteta Fresnelin vyöhykkeen tuomaa etua siirrossa.

Rakennuksien vaikutus radiolähetyksiin on pieni muutamien hertsien taajuusalueella, jossa rakennusesteen koko on pieni verrattuna aallonpituuteen. Yli 30 MHz taajuuksilla radioaallon vaimennus sen kulkeutuessa esteen läpi tai yli, ei ole enää



Kuva 2: Fresnelin vyöhykkeellä oleva este ja vapaa näköakseli.

merkityksetön. Yleisradiojärjestelmien suunnittelussa ei ole kuitenkaan ollut käytännöllistä huomioida jokaista rakennusta omana geometrianaan, vaan rakennusten aiheuttamaan vaimennusta on käsitelty tilastollisesti. Esimerkiksi kaupungin keskustan korkeiden rakennusten aiheuttama vaikutus on erilainen kuin matalammin rakennettujen asuinalueiden. Rakennukset yleisesti ottaen ovat radioaalloille näkyvämpiä kuin maanpinnan muodot aiheuttaen siten pienempää vaimennusta. Toisaalta taas diffraktiokulmat rakennuksissa ovat suurempia kuin maastossa, joka taas omalta osaltaan lisää häviötä [3]

## 2.4 Yhteenveto maanpäällisistä joukkoviestintäverkoista

Antenniverkossa tapahtuva televisio- ja radiotoiminta Suomessa on luvanvaraista toimintaa, joka vaatii verkkotoimiluvan sekä lähettimen radioluvan. Maanpäällisessä jakelussa televisio- ja radiopalveluiden jakelu tapahtuu peittoalueen kattavalla maantieteellisellä alueella, joka rajautuu signaalin etenemämallin mukaan. Signaalin etenemiseen vaikuttavat mm. näköesteet, maan kaareutuminen, voimakkaasti heijastusta aiheuttavat terävät kulmat, kasvillisuus ja asutustyypit.

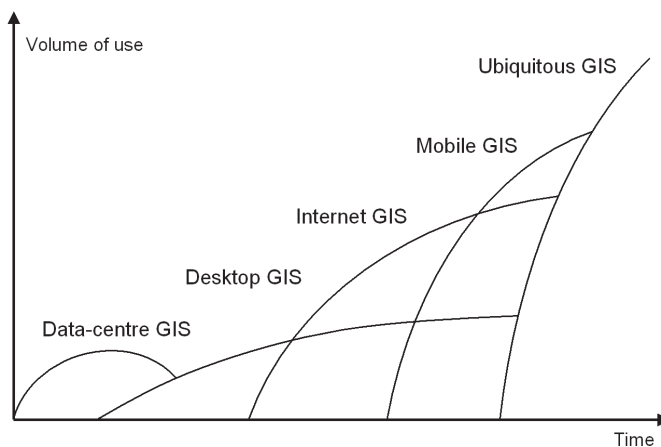
### 3 Paikkatieto ja paikkatietojärjestelmät

Tämän luvun tarkoituksena on esitellä paikkatietoon liittyviä käsitteitä; standardeja ja paikkatietojärjestelmiä. Luvussa käydään läpi paikkatietojärjestelmien kehittyminen viimeisen puolen vuosisadan aikana nykyiseen malliinsa. Paikkatietoa ja paikkatietoteknologiaa käsitellään ensisijaisesti tässä työssä toteutetun karttapalvelusovelluksen tarpeiden näkökulmasta. Erityishuomionsa saa myös selvitys siitä, miten paikkatietoa on tähän mennessä hyödynnetty tietoliikennealalla.

#### 3.1 Paikkatiedon historia

Paikkatietojärjestelmät (GIS) ovat kehittyneet merkittävästi tähän päivään mennessä – kehityksen nähdään alkaneen jo 1960-luvulla. Aluksi paikkatietojärjestelmät olivat tietokonekeskuksissa olevia järjestelmiä, josta ne muuttuivat vaiheittain pöytätietokoneiden sovelluksiksi, verkkosovelluksiksi, mobiilisovelluksiksi ja lopulta tulivat osaksi joka paikan tietotekniikkaa. Tämä on samalla tarkoittanut paikkatietojärjestelmien ja niiden käyttäjien lukumäärän merkittävää kasvua. [4]

1980-luvun puoleen väliin mennessä paikkatietojärjestelmiä käytettiin vahvasti luonnonvara- ja maanpuolustusaloilla, erityisesti valtionhallinnon toimipaikoilla. Sovellukset olivat omimmillaan paikkatietoanalyysissä, mutta tietoa ei voitu jakaa järjestelmien välillä tehokkaasti. [5] Perinteiset paikkatietojärjestelmät, jotka keskittyivät tarjoamaan toimintoja ammattilaiskäyttöön, rakennettiin suljetuiksi, erillisiksi ja yksittäisiksi järjestelmiksi [6].



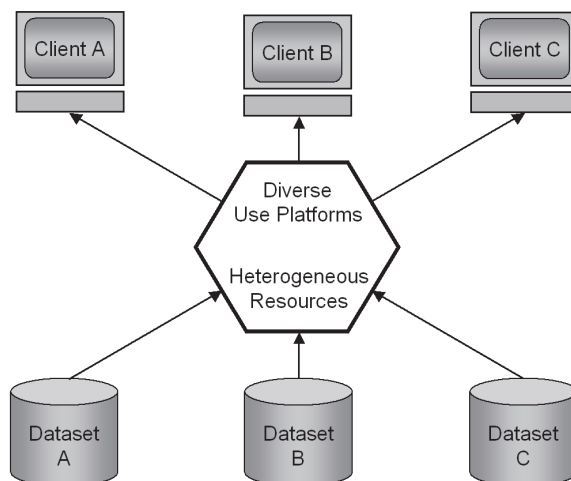
Kuva 3: Paikkatietojärjestelmien kehittymisen viisi vaihetta. [4]

Ensimmäisen sukupolven internet-sovelluksissa käytettiin yksinkertaisia valmiiksi visualisoituja karttanäkymiä. Paikkatietojärjestelmien kehittyessä tarvittiin hienostuneempia lähestymistapoja. Karttapalveluilta vaadittiin mahdollisuutta konfiguroida



niiden visualisoinnin tilaa dynaamisesti käytön ja käyttäjän tarpeiden mukaisesti [7]. Tämä tarkoitti sitä, että esimääritellyn sisällön sijaan karttasovelluksien piti pystyä prosessoimaan todellista tietosisältöä [4]. Tietotekniikan ja tietoverkkoteknologian kehittyessä paikkatietojärjestelmät alkoivat muuttua avoimiksi, hajautetuiksi ja monikoneympäristöissä toimiviksi järjestelmiksi. Samalla paikkatietojärjestelmien käyttö muuttui päivittäiseksi julkiseksi käytöksi. [6]

Internet-pohjaisen paikkatiedon käsittelyn lisääntyessä on tullut vastaan kahdenlaisia monimuotoisuuden haasteita. Ensimmäiseksi, erimuotoista geospaatialista tietoa on saatavilla web-palveluista ja tämän monimuotoisen tiedon yhtäaikainen prosessointi sovelluksissa on tullut haasteelliseksi [8]. Toiseksi, yhä erilaisemmat käyttäjäyhteisöt ovat kiinnostuneita paikkatietoa tarjoavista web-palveluista, jolloin niiden on tuettava useampia päätelaitteita ja alustoja. Näitä haasteita on paras lähestyä uudelleenkäytettävien ratkaisujen kautta. Tällaisilla ratkaisuilla maksimoidaan rajapintoihin tehtyjen investointien hyötyjä, ne mahdollistavat integraatiot ja luovat lisäarvoa uusien tuotteiden ja palveluiden kautta. [4]



Kuva 4: Kaksi internet-paikkatietojärjestelmien moninaisuushaastetta. [4]

Tiukasti määriteltyjen internet-standardien ja -teknologioiden omaksuminen paikkatietosovelluksissa on edistänyt niiden yhtenäistämiskäytäntöjä. Tämän seurauksena kohteisiin (feature) perustuva paikkatieto internetissä on mahdollistanut standardoidut menetelmät tiedon mallintamiseen, koodaukseen ja prosessointiin. [9, 10]

### 3.2 Paikkatietojärjestelmät

Paikkatietojärjestelmä (Geographic Information System, GIS) on maantieteellistä tietoa käsittelevä järjestelmä, joka mahdollistaa karttojen luonnin ja paikkatietoanalyysin. Ponce-Rojas ym. [11] on määritellyt sen tietojärjestelmäksi, joka keskittyy aikaisemmin tallennetun datan hallintaan ja prosessointiin. Se on tehokas tuottamaan

tarpeellista informaatiota tiedonkeruuta, tiedonhallintaa, paikkatietoanalyysiä ja tilastollista analyysiä varten. GIS on myös tehokas järjestelmä tiedon rikastamiseen ja integraatioihin. Käyttäjät voivat helposti kysellä tietojoukosta paikkatietokohteita ja ominaisuuksia. [12] Paikkatietojärjestelmä mahdollistaa tehokkaat, toistettavat ja standardoidut paikkatietokonsultaatiot, joilla saavutetaan lisäarvoa ylläpidettyyn tietoon [11]. Paikkatietojärjestelmät voivat olla myös joka paikan hajautettuja järjestelmiä. Tiukan määritelmän mukaan ne ovat tietokonejärjestelmiä, jotka voivat integroida, varastoida, editoida, analysoida, jakaa ja esittää maantieteeseen liittyvää informaatiota. Yleisemmin paikkatietojärjestelmä on työkalu, jolla käyttäjä voi tehdä vuorovaikutuksellisia kyselyjä, analysoida paikkatietoa, editoida dataa, karttoja ja esittää kaikkien näiden toimenpiteiden tuloksia käyttäjille. [5]

Paikkatietojärjestelmien käyttäjät hyödyntävät paikkatietoa liiketoiminnassa, hallinnossa, tieteessä tai yksityisesti. Paikkatiedon integrointi muiden tietojärjestelmien kanssa voi mahdollistaa rikastetun paikkatietosisällön. Kahden järjestelmän analyttisten ominaisuuksien yhdistäminen antaa paremmat lähtökohdat muuntaa data hyödylliseen muotoon. [13] Paikkatietojärjestelmät käyttävät hyväkseen erilaista maan pintaan liittyvää informaatiota. Esimerkiksi maanpinnan muotojen korkeusmalleja tai maantieteellistä paikkainformaatiota hyödynnetään paikkatietoanalyysissä, navigoinnissa tai paikkatiedon haussa. Paikkatietojärjestelmillä on erittäin läheinen suhde tietotekniikan kanssa. [6]

Paikkatietojärjestelmien pitää pystyä piilottamaan ammattimaiset maantieteeseen liittyvät ominaispiirteet ja tarjoamaan yksinkertainen palvelun käyttöliittymä käyttäjälle. Ammattimaisessa käytössä paikkatietojärjestelmien on pystyttävä jakamaan ja käyttämään uudelleen maantieteellistä dataa ja niiden edistyneitä toimintoja. [6]

### 3.3 Tietoliikennetekniikan paikkatietojärjestelmät

Paikkatietojärjestelmien käyttö tietoliikennetekniikassa alkoi AM/FM/GIS-sovellusten myötä. Termillä AM/FM/GIS usein viitataan paikkatietojärjestelmiin, joilla julkisen palvelun infrastruktuuriin liittyvää tietoa on hallittu ja analysoitu. Langattomassa teollisuudessa taas ensimmäiset paikkatietosovellukset liittyivät radioaaltojen etenemisen analyysiin. Yrity maailmassa käytetyt sovellukset olivat alkujaan yksiköiden sisäisessä käytössä olevia suunnittelutyökaluja. Sitten huomio paikkatietojärjestelmissä on liittynyt enemmän tiedon jakamiseen yksiköiden välillä, yrityksen sisäverkoissa sekä internetissä. Paikkatietojärjestelmien standardoinnin ja tietotekniikan lisääntyneen suorituskyvyn myötä, tietoliikennepalvelujen tarjoajat ovat pyrkineet saavuttamaan kilpailuetua sitomalla asiakkaita, myyntialueita, työnkulkuja ja peittoalueita sijaintitiedon kanssa yhteen. [14]

Tietoliikennealan yritykset hyödyntävät paikkatietoa monessa eri yhteydessä. Paikkatietojärjestelmä tukee yrityksen päätöksentekoa, minne verkkopalveluja pitäisi laajentaa ja mitä palveluita siellä tulisi tarjota. Paikkatietojärjestelmän rooli tietoliikenneyrityksissä on myös toimia keskitettynä tietokantana, jonne paikkatietoa tallennetaan, jossa sitä prosessoidaan ja josta sitä haetaan eri organisaatioyksiköiden toimesta. Näin yrityksen eri toiminnoilla on ajantasainen näkymä tietoon, joka

mahdollistaa nopeiden muutosten tekemisen verkkoon heti kun tarve siihen on tunnistettu. Lisäksi, tietoliikenneteollisuudessa toimiva asiakaspalvelu on välttämättömyys. Asiakaspalvelun kautta asiakkailta saadaan tietoa palveluiden saatavuudesta eri alueilla keskitettyyn paikkatietojärjestelmään. Samalla myös tallennettu tieto palvelee asiakaspalvelua ja mahdollistaa täsmällisen palveluiden saatavuuteen liittyvän tiedon välittämisen asiakkaille. Internet-palvelujen yleistyessä monet tietoliikennealan yritykset ovat rohkaistuneet jakamaan paikkatietoaan myös julkisessa verkossa yrityksen sisällä jaettavan paikkatiedon rinnalla. Internet-palvelut ryhtyivät tarjoamaan välitöntä ja ajantasaista tietoa samalla kun palveluiden käyttö yleistyi mobiililaitteissa. Mobiililaitteet taas puolestaan mahdollistivat sijaintiin sidottujen palvelujen tarjoamisen, koska laitteet kykenivät vaihtamaan tietoa GPS:n ja muiden paikannustekniikoiden kanssa. Esimerkiksi langattoman tietoliikennepalvelun laskutus voi perustua käyttäjän sijaintiin ja erilaisiin laskutusvyöhykkeisiin. [14]

Paikkatietojärjestelmiä on käytetty tietoliikenneyritysten asiakaspalvelutoiminnoissa jo kauan. Varsinkin langattomien verkkojen peittoalueet ja kapasiteetit muuttuvat jatkuvasti, jolloin asiakaspalvelun asiakkailleen tarjoaman tiedon on oltava maantieteellistä ja ajantasaista. Asiakkaille on esimerkiksi pitänyt tarjota ennakkoon tietoa langattoman verkon saatavuudesta, varsinkin silloin kun asiakas on lähdessä alueelle, jossa hän ei ole aikaisemmin käynyt. Toisena esimerkkinä paikkatietojärjestelmä voi tarjota asiakaspalvelulle tietoa alueellisista häiriöistä verkoissa ja niiden korjausajoista. Tällöin asiakaspalvelu pystyy palvelemaan yhteyttä ottavia asiakkaita täsmällisesti. Jos paikkatietojärjestelmä sisältää tiedon paikallisista palvelukeskuksista, voi asiakaspalvelu myös ohjata asiakkaan ottamaan yhteyttä tällaiseen, mikäli häiriö ei ole verkossa, vaan vastaanottavassa laitteessa. [14]

Tietoliikenteen paikkatietojärjestelmät eivät välttämättä palvele vain nykyisiä asiakkaita, vaan niitä voidaan hyödyntää uusien asiakkaiden hankinnassa. Järjestelmä voi tarjota myynnille tietoa esimerkiksi kuvaamalla paikkakuntakohtaiset asiakasmäärät, tulotasot, tarjolla olevat palvelut ja siten kohdistaa markkinointitoimenpiteitä halutuille alueille. [14]

Tietoliikenneyritykset ovat tunnistaneet monissa prosesseissaan paikkatietoelementin. Jakamalla tätä paikkatietoa eri yksiköiden välillä, on mahdollista hyödyntää tietoa entistä tehokkaammin. Aikaisemmin operaattorit tallensivat tietoa sijainneista, kokoonpanoista, kapasiteeteista ja verkon tilasta erilaisille kartoille, kuvaajille ja raportteihin. Nämä olivat tyypillisesti hajallaan organisaatiossa. Kun insinöörin tarvitsi päästä kiinni tietoon koskien jotain verkon osaa, hänen piti ensin löytää asianmukainen dokumentti ja tulkita toisen työntekijän tuotoksia. Usein yksityiskohdat olivat epätarkkoja tai jopa vanhentuneita. Tällöin saatettiin joutua tekemään kallisarvoisia kenttätutkimuksia ennen varsinaisen työn aloittamista. [15]

Kustannusten leikkaamisen ja palveluiden kehittämisen tarpeet ovat pakottaneet yritykset pohtimaan tiedon uudelleenjärjestelyä. Yhä enemmän yksiköt joutuvat kilpailemaan projektien rajoituksesta, yhtiöt eivät ole kiinnostuneita käyttämään rahaa laajamittaiseen tietojen muuntamiseen tai sen hankintaan ja vaativat näyttöä projekteihin sijoitetun pääoman tuotolle. [15]

Paikkatieto on nykyään perusteknologiaa tietoliikenneteollisuudessa. Tyypillisesti teknologiaa on käytetty tietoliikenneverkkojen ja siihen liittyvien palvelujen

suunnittelussa, rakentamisessa ja operoinnissa. Tietoliikenneteollisuuden lisäksi tietoliikenteen paikkatietojärjestelmiä käytetään tutkimus- ja koulutuslaitoksissa, kuten yliopistoissa, sähkötekniikan oppilaitoksissa sekä viestinnän ja markkinoinnin osastoilla kaupallisissa oppilaitoksissa. Näissä laitoksissa paikkatietojärjestelmiä käytetään verkkosuunnittelun, verkon operoinnin ja markkinoinnin opetuksessa. [15]

Teletuotemallin vapautuessa kilpailulle, tuli tietoliikenneyrityksille suurempi tarve kustannussäästöihin ja asiakashankinnalle. Tietoliikenneyritysten investoinnit muuttuivat laitteiden elinkaaripäivitysten sijaan investointeihin, jotka mahdollistivat uusien asiakkaiden tehokkaamman hankinnan. Tällä muutoksella oli vaikutuksensa myös siihen miten tietoliikenneyritykset ovat hallinneet tietoansa. Jos verkon suunnittelu ja operointi olivat olleet paikkatietojärjestelmän keskeisimpiä käyttötapoja, niin nykyään ne mahdollistavat esimerkiksi väestörakenteen visualisoinnin. Tällaista dataa ovat asukasluku, naapuruston tyytit, keskimääräiset ansiotulot ja muut asiakkaiden ominaisuudet. [15]

Tietoliikennepalveluita tarjoaville yrityksille paras tapa suuntautua uusille markkinoille on vertailla verkkoa nykyisiin ja potentiaalisten asiakkaiden tietoihin. Kaapeliverkkooperaattorit ovat olleet tietoliikenneteollisuudessa edelläkävijöitä paikkatietojärjestelmien hyödyntämisessä sekä teknisissä, että markkinointitarkoituksissa. Kaapeliopeeraattoreiden kahtena päätavoitteena on ollut rakentaa verkko taloudellisesti ja tehokkaasti sekä saavuttaa nopea sijoitetun pääoman tuotto priorisoimalla myyntitoimet suurimman välittömän potentiaalin alueille. [15]

Fry [15] näki tietoliikenteen paikkatietojen tulevaisuuden lupaavana jo 1990-luvun lopulla. Kovan kilpailun vuoksi tietoliikenneinfrastruktuurin on oltava nopeaa ja tehokasta, jolloin paikkatietojärjestelmät tulevat verkon suunnittelussa pienimpienkin yritysten käyttöön. Suuremmat yritykset taas pyrkivät saamaan paikkatietojärjestelmistä strategista etua standardoimalla paikkatietoa koko organisaatiossa. Tällöin sama insinöörien tuottama verkkosuunnittelun paikkatieto voi olla esimerkiksi markkinointihenkilöstön käytettävissä. Kaikki tieto on strukturoidussa tietokannassa, josta se on saumattomasti käytössä rajapintojen kautta eri yksiköiden tietojärjestelmissä. Tuoremmat tietoliikenneyritykset voivat ketterämmin implementoida uutta paikkatietoteknologiaa, koska niillä ei ole painolastina vanhoja perinnejärjestelmiä. [15]

Tässä työssä suunniteltu tietoliikenteen paikkatietojärjestelmä tukee ensisijaisesti viestinnän, markkinoinnin ja asiakaspalvelun tarpeita. Järjestelmän avulla voidaan saavuttaa tiettyjä kustannussäästöjä, kun osa viestinnästä on mahdollista automatisoida paikkatietojärjestelmässä.

### 3.4 Paikkatietoteknologia

Tässä työssä käsiteltävä paikkatieto on pääasiassa vektorimuotoista. Työn paikkatietojärjestelmälle tuotetaan vektorimuotoista paikkatietoa, järjestelmä muuntaa sitä sekä tarjoaa sitä standardien rajapinnan kautta XML-muotoisena. Tässä luvussa esitellään tämän työn kannalta oleellisia paikkatiedon standardoituja muotoja ja rajapintoja.

### 3.4.1 Paikkatieto ja sen muunnokset

Paikkatietosisällöllä voidaan tarkoittaa paikkatietoa sen visuaalisessa muodossa, kuten karttoja. Toisaalta paikkatietosisältö voi tarkoittaa myös lähdedataa, jonka pohjalta tietoa on mahdollista visualisoida. [4]

Skeemamuunnoksella tarkoitetaan prosessia, jossa paikkatieto muunnetaan toisesta skeemasta toiseen. Tällainen muunnosprosessi voidaan luokitella skemaattiseksi muunnokseksi. Skeemamuunnos voidaan jakaa kahdeksi eri vaiheeksi: ensimmäisessä vaiheessa määritellään muunnos ja toisessa suoritetaan muunnos. [4]

Reaaliaikaisella sisältömuunnoksella tarkoitetaan muunnosta, joka tehdään verkkopalveluympäristössä osana synkronista pyyntö-vastaus dialogia [4]. Tyypillisesti dialogi tapahtuu asiakassovelluksen ja palvelimen välillä kaksikerrosarkkitehtuurissa, mutta tässä työssä arkkitehtuuria on laajennettu välikerroksella, joka vastaa kaikista tarvittavista reaaliaikaisista sisältömuunnoksista. Tässä työssä sisältömuunnoksia tehdään reaaliaikaisesti, mutta ratkaisu perustuu myös tiettyihin ennalta tehtäviin sisältömuunnoksiin.

Vektorimuotoista paikkatietoa voidaan käsitellä joukkona kohteita, jotka vastaavat reaali maailmassa tunnistettavia objekteja ja ilmiöitä. Tietomalli taas on käsitteellinen rakenne, joka esittää tietokomponenttien suhteen reaali maailman objekteihin. Tietomalli kuvataan täsmällisesti formaalilla mallinnuskielellä, jota kutsutaan skeemaksi. [4]

### 3.4.2 Extensible Markup Language (XML)

Extensible Markup Language (XML) on W3C:n kehittämä syntaksi- ja toiminnallisuusmäärittely tiedon koodaamiseen tekstimuotoon. XML on suunnattu erityisesti web-käyttöön. XML määrittelee syntaksisäännöt ja perusrakennekomponentit XML-dokumenttien muodostamiseen. XML:n perusstandardi ei määrittele mitään konkreettisia sanastoja tiettyyn sovelluskäyttöön, joten XML voidaan käsittää metakielenä. Vuosien varrella on kehitetty satoja erilaisia XML-sanastoja erilaisten sovellusten käyttöön. Esimerkiksi Geographic Markup Language (GML) on parhaiten tunnettu XML-sanasto, joka on suunniteltu paikkatietosovelluksille. XML määrittelee myös ohjelmistokomponentin, XML-jäsentimien toiminnallisuusvaatimukset. XML-jäsenin lukee ja validoi XML-dokumentin. Perusstandardina XML yleisenä teknologiana muodostaa perustan tiedon, kuvien ja sanomien koodaukselle. [4]

Paikkatietojärjestelmien Web Services -rajapinnat ovat ohjelmistokomponentteja, joihin muiden sovellusten on mahdollista päästä kiinni internetin kautta. Web Services on ohjelmointirajapinta (API), joihin on pääsy verkon, kuten internetin kautta. Web Services -komponentit isännöidään ja suoritetaan etäjärjestelmässä. [5] Web Services -teknologiaa on esitelty tarkemmin tämän työn luvussa 5.

XML Information Set (XML Infoset) on W3C:n määritelmä XML-dokumenttien abstraktille tietomallille. Sen tarkoituksena on tarjota yhdenmukainen joukko määritelmiä muille standardeille, joilla on tarve viitata hyvin muotoiltuun XML-dokumenttiin. Sen tarkoituksena ei ole olla täydellinen, vaan siinä on pyritty ennakoimaan määrittelyjen tarve tulevilla standardeilla. XML-dokumentti toteuttaa XML Infosetin,

mikäli se on hyvin muotoiltu ja toteuttaa nimiavaruuden asettamat vaatimukset. XML voi kuitenkin olla validi, vaikka se ei toteuttaisi XML Infosettiä. [16]

XML-dokumentin Infoset koostuu informaatioyksiköistä. Hyvin muotoiltu XML Infoset sisältää ainakin yhden dokumentti-informaatioyksikön, sekä muutamaan muun informaatioyksikön. Informaatioyksikkö on abstrakti kuvaus jostain XML-dokumentin osasta. Jokaisella informaatioyksiköllä on joukko nimetty ominaisuuksia. Informaatioyksikköjä voidaan verrata tietojenkäsittelyssä käytettyihin tietorakenteisiin, puuhun ja sen solmuihin. Näitä termejä ei kuitenkaan XML Infosetin standardissa käytetä, jottei tapahtuisi sekoittumista tiettyihin muihin tietomalleihin. [16]

### 3.4.3 Paikkatietojärjestelmien tietomuodot

Kartat voidaan esittää kahdella erilaisella perustyypillä, rasterimuodossa tai vektorimuodossa. Rasterimuodossa kartta esitetään bittikarttakuvana ja se voidaan tarvittaessa pakata esimerkiksi JPEG, TIFF, GeoTIFF, GIF tai PNG-algoritmeilla. Vektorimuodot ovat monimutkaisempia, koska niissä karttatieto tallennetaan joukko- na koordinaattipisteitä, jotka muodostavat esitettäessä pisteitä, janoja ja monikulmioita. Näiden lisäksi vektorimuoto mahdollistaa myös muun kiinnostavan tiedon tallentamisen, kuten katujen nimien, kaistojen lukumäärän autoteillä, yksisuuntaiset kadut, tekemällä näin siitä käyttäjälle navigoitavamman. Rasterikartassa kaikki kartalla oleva informaatio on aina näkyvillä, mutta vektorimuotoisessa tieto on tallennettu kerroksittain, eikä kaikkien kerrosten tarvitse olla koko ajan näkyvillä. [17]

Vektorikarttamuotoja on olemassa myös useita. Osa näistä on patentoituja muotoja yrityksiltä, jotka tarjoavat myös paikkatietosovelluksia, kuten AutoCAD, MapInfo ja ESRI. Näiden lisäksi on olemassa kansainvälisiin standardeihin perustuvia muotoja. GDF on avoin standardi, jota on käytetty pääasiassa kuvaamaan ja siirtämään teihin liittyvää tietoa. Se on tekstimuotoisena standardina tehoton, koska digitaalisen tiedon tallennus vie paljon levytilaa ja on hidas lukea. [17]

### 3.4.4 Geographic Markup Language (GML)

Open Geospatial Consortium (OGC) tarjoaa toisenlaisen lähestymistavan vektorimuotoihin. OGC on yhdistys, joka toimii avoimien ja laajennettavien ohjelmointirajapintojen kehityksessä paikkatietojärjestelmille. OGC:n hyväksymät määrittelyt ovat julkisia ja ilmaisia. Tarkoituksena on tuottaa paikkatietoa ja paikkatietopalveluita, joiden käyttö on teknologiariippumaton. OGC on standardoinut Geographic Markup Language (GML). GML on XML-kielioppiin perustuva kieli, jota käytetään paikkatiedollisten ja paikkatiedottomien ominaisuuksien mallintamiseen, siirtämiseen ja varastointiin. GML määrittelee XML-skeeman muutosäännöt, mekanismit ja käytännöt, jotka tarjoavat avoimen ja toimittajariippumattoman kehyksen paikkatietosovellusten skeemojen ja objektien määrittelyyn. XML-pohjaiset kielet ovat soveltuvia tietojen siirtoon hajautettujen ja heterogeenisten järjestelmien välillä. XML on siis tarkoituksenmukainen paikkatietojärjestelmien kehittämisessä ja mahdollistaa tietointegraation hajautetuista lähteistä. [17]

GML:n perusstandardi esittelee joukon yleisiä XML-elementtejä, jotka on tarkoitettu paikkatietosovellusten käyttöön. Nämä ovat esimerkiksi geometriatyyppejä, joita esiintyy paikkatiedon tietojoukoissa. GML taas ei määrittele elementtityyppejä kohdeluokille. Kohdeluokka voi olla esimerkiksi tie, jolloin paikkatiedon kaikki tiedot jakavat tämän kohdeluokan määrittelemät attribuutit. Nämä määritellään aina paikallisesti sovelluksen skeemassa, jotka noudattavat GML-perusstandardin sääntöjä. Uusien GML-sovelluksissa määriteltyjen elementtityyppien on periydyttävä GML:n abstrakteista perustyypeistä. GML-sovellusten skeemat voivat esitellä sekä yksinkertaisia, että komplekseja ominaisuuksia määriteltyille kohdetyypeille. Kohteella voi olla useita geometrisia ominaisuuksia, joka on tallennettu geometrian tietoihin. Määrittely sisältää myös konseptin kohdejoukoista, jolla useita kohteita voidaan paketoita yhdeksi joukoksi. [4]

GML:ää on sovellettu laajasti paikkatietosisällön koodausmenetelmänä. GML on myös ISO-standardi (ISO 19136:2007). ISO-standardin määrittelemällä XML-skeemalla pyritään tarjoamaan avoin ja toimittajariippumaton viitekehys paikkatietojen siirtämiseen ja tallentamiseen XML-muodossa. [18]

Paikkatietojärjestelmien haasteina ovat olleet suorituskyky ja järjestelmien yhteensopivuus. Paikkatiedon käsittelyyn paikkatietojärjestelmissä on käytetty paljon relaatiotietokantoja ja niiden paikkatietolaajennuksia. Tietokantojen kehittyneitä indeksointitekniikoita hyödyntäen paikkatiedon hakeminen tietokannasta on ollut tehokasta. Monet paikkatietojärjestelmät ovat tukeneet XML:n ja GML:n käyttöä tiedon viennissä ja tuonnissa. Yhteensopivuusongelmia on yleensä tullut siitä, kun paikkatietojärjestelmien ja tietokantojen valmistajat käyttävät omia tietokantaskeemoja ja tiedostomuotoja. Toinen järjestelmä ei välttämättä ymmärrä toisen järjestelmän luomaa dataa, eikä siten voi hyödyntää sitä suoraan. Paikkatiedon käsittelyssä relaatiotietokannassa on myös haasteita. Reaalimaailman ilmiöt ovat luontaisesti monimutkaisia ja relaatiomalli usein epäonnistuu sen rakenteiden ilmaisussa. [19]

GML on relaatiomallia joustavampi ja luonnollisempi tapa ilmaista paikkatietoa. Siitä on nopeasti tullut avoin standardi tiedon mallintamisessa, siirtämisessä ja tallentamisessa. GML koodaa paikkatiedon ja ei-paikkatiedon XML-muotoon. GML-muotoa käyttävät paikkatietosovellukset lieventävät järjestelmien yhteensopivuusongelmaa. Tiedon editointi suoraan GML dokumenttiin on helpompaa kuin tiedon vienti relaatiotietokantaan. Internetin ollessa suosittu media informaation siirrossa, on myös paikkatiedon siirto internetin yli tullut tärkeämmäksi. OGC:n Web Processing Service (WPS) määrittelee standardoidun rajapinnan, joka tekee helpommaksi paikkatietoprosessien julkaisemisen. XML- ja GML-teknologioiden käyttäminen tietokantateknologian sijaan paikkatietojärjestelmien ytimessä mahdollistaa paremman tuen standardeille. [19]

Yleiset XML-käsittelijät eivät prosessoivat suuria dokumentteja tehokkaasti ja on yksi yleisten XML-käsittelijöiden heikkous. GML:n määrittelyssä ei huomioida suorituskykyyn liittyviä kysymyksiä. Tämä saattaisi viitata siihen, että GML ei ole tarkoitettu hyvää suorituskykyä vaativiin sovelluksiin, eikä tällöin paras valinta verkkosovelluksiin. [19]

### 3.4.5 Web Map Service (WMS)

Karttakuvapalvelu eli Web Map Service on OGC:n kehittämä palvelurajapinnan määrittely, joka on myös saatavilla virallisena ISO-standardina (ISO 19128:2005). WMS määrittelee Web Services -palvelun, joka tarjoaa pääsyä paikkatietoon visualisoidussa karttamuodossa. WMS-palvelu tarjoaa perustoimintona palvelun metadatan, toimittaa karttoja ja lisätoimintona tarjoaa lisätietoa kartalla klikattavasta kohteesta. WMS määrittelee joukon parametreja kyselyitä varten sekä määrittelee merkityso- pin palvelun ja asiakasohjelmiston tiedonsiirron välille. WMS-palvelut tyypillisesti toimittavat kartat rasterimuodossa. [4]

### 3.4.6 Web Feature Service (WFS)

Kohdepalvelu eli Web Feature Service on myös OGC-lähtöinen palvelurajapinnan määrittely paikkatiedon toimittamiseen. WFS-palvelu tarjoaa pääsyn palvelun meta- dataan, kohdetyyppien skeemakuvauskuviin ja kohdemuotoiltuun esitykseen itse datasta. Lisäksi WFS-rajapintamäärittely sisältää toiminnot datan päivittämiseen rajapinnan kautta. Rajapinnassa on myös ominaisuuksia tiedon lukitukseen, jotta päivitysten yh- tenäisyys säilyisi. Oletusarvoisesti kohdetyyppien skeemakuvauskysely palauttaa osan XML-skeemaa. Varsinaisen datan kysely taas palauttaa GML-koodattua dataa. WFS on myös esitelty ISO-standardina ISO 19142:2010. [4] Tässä työssä WFS-palvelulla on suuri merkitys paikkatiedon siirrossa taustajärjestelmistä paikkatietosovellukseen.

## 3.5 Yhteenveto paikkatiedosta

Paikkatiedon käyttö ja paikkatietojärjestelmät ovat kehittyneet viime vuosikym- menien aikana järeistä keskitetyistä järjestelmistä joka paikassa käytettäväksi tek- nologiaksi. Tietoliikenneyrityksissä paikkatiedon hyödyntämisellä on pitkä histo- ria. Paikkatietoa on hyödynnetty suunnittelussa, markkinoinnissa ja asiakaspal- velussa. Paikkatieto voi olla rasteri- tai vektorimuotoista. XML-pohjainen GML on suosittu paikkatiedon tallennus- ja siirtomuoto, jota hyödynnetään mm. WFS- palvelurajapinnassa.



## 4 Hajautettujen järjestelmien arkkitehtuurimallit

Tässä luvussa tutkitaan erilaisia ohjelmiston arkkitehtuurimalleja, jotka on erityisesti tarkoitettu hajautettuihin järjestelmiin. Tarkasteluun joutuvat palvelukeskeinen arkkitehtuuri, HTTP-protokollaan perustuva REST sekä Mashup.

### 4.1 Arkkitehtuuri

Ohjelmistoarkkitehtuuri on kuvaus järjestelmän pääkomponenteista, niiden relaatioista sekä niiden välillä välitettävästä informaatiosta. Arkkitehtuuri on suunnitelma järjestelmiä varten, jotka toteuttavat hyvin määritellyt vaatimukset nyt ja tulevaisuudessa. Ohjelmistoarkkitehtuurin perustavoitteena on helpottaa monimutkaisten ohjelmistojärjestelmien hallintaa erityisesti liiketoiminnan, organisaation ja teknisen ympäristön muutostilanteissa. [20]

Ohjelmistoarkkitehtuuri pitää sisällään ohjelmistojärjestelmän rakenteen. Ohjelmistoarkkitehtuurissa otetaan kantaa järjestelmän rakenneosien valintaan, niiden keskinäisiin rajapintoihin sekä yhteistoimintaan. Arkkitehtuuri määrää miten näistä rakenneosista muodostuu suurempia alijärjestelmiä. Ohjelmistoarkkitehtuuri määrää tyylin, joka ohjaa ohjelmiston rakennetta, rakenneosia, rajapintoja, sekä näiden yhteistoimintaa ja kokonaisuutta. Ohjelmistoarkkitehtuuri ei käsittele ainoastaan rakennetta ja toimintaa, vaan myös ohjelmiston käyttöä, toiminnallisuutta, suorituskykyä, vikasietoisuutta, uudelleenkäyttöä, ymmärrettävyyttä, taloudellisia ja teknisiä rajoitteita, kompromisseja sekä esteettisyyttä. [21]

Useimmat ohjelmistoarkkitehtuurin määritelmät ovat yksimielisiä siitä, että arkkitehtuuri kuvaa järjestelmän koostumuksen. Määritelmien eroavaisuuksia syntyy taas erilaisista näkökulmista määritellä järjestelmä tai kokoonpano. Perinteinen ohjelmistoarkkitehtuuri on keskittynyt sovellusten rakentamiseen, kun taas palvelukeskeinen arkkitehtuuri keskittyy laajuudeltaan yritystasoisiin tai organisaatorajat ylittäviin ratkaisuihin. Tyypillinen ohjelmistoarkkitehtuuri voi kuvata luokkarakenteen tai komponentit ja niiden yhteydet, sekä niiden tuottaman arvon ohjelmistotuotteelle, mutta palvelukeskeinen arkkitehtuuri kuvaa palveluiden rakenteen, niiden yhteydet, sekä niiden tuottaman arvon yritystason prosesseille ja ratkaisuille. [20]

Tuotteilla, sovelluksilla ja yrityksillä on jokaisella oma ainutkertainen arkkitehtuurinsa, vaikka niiden välillä voikin olla paljon samankaltaisuuksia. Arkkitehtuurinen tyyli on käsitteiden ja yhteyksien sanasto, sekä rajoitteiden joukko, jotka yhdistettyinä tuovat ratkaisun korkeamman tason tavoitteisiin muodostamalla erityisen arkkitehtuurin. Arkkitehtuurinen tyyli on samojen periaatteiden ja ominaisuuksien jakavien arkkitehtuurien joukko. Arkkitehtuurinen tyyli yrityssovelluksille valitaan asetettujen vaatimusten pohjalta. Valitulla arkkitehtuurisella tyylillä voidaan esimerkiksi vaikuttaa kuinka hajautettu, skaalautuva, laiteriippumaton, uudelleen käytettävä ratkaisu on. [20]

Arkkitehtuurin tärkein peruseriaate on haasteiden eriyttäminen. Haasteiden eriyttämisellä pidetään arkkitehtuurin komponentit erillisinä. Tämän etuna nähdään se, että muutokset yhdessä osassa järjestelmää eivät vaikuta haitallisesti muihin osiin. Arkkitehtuurinäkömät taas tarjoavat toisen tärkeän tavan haasteiden eriyttämi-

seen. Arkkitehtuurinäkymään voidaan sisällyttää tai jättää pois tietyt yksityiskohdat riippuen mille sidosryhmän jäsenelle tietoa esitetään. Tyypilliset yrityksen arkkitehtuurinäkymät ovat liiketoiminnallinen näkymä, informaationäkymä, sovellusnäkymä, teknologianäkymä ja toteutuksellinen näkymä. Arkkitehtuurin pitää tarjota riittävää joustavuutta, jotta tulevat muuttuvat tarpeet voidaan helposti tyydyttää. Joustava arkkitehtuuri tunnistaa tulevat muutostarpeet sekä mahdolliset muuttuvat osa-alueet. Abstraktio on tärkeä arkkitehtuurinen työkalu, jolla erotellaan haasteita ja mukautetaan muutokseen. Abstraktiokerros kahden kerroksen välissä erottaa kerrokset suorasta vuorovaikutuksesta, joka puolestaan lisää joustavuutta. Abstraktion avulla päästään myös korkeamman tason vuorovaikutukseen. Yksi arkkitehtuurin tavoite on yhdenmukaisuuden ja uudelleenkäytettävyyden lisääminen. [20]

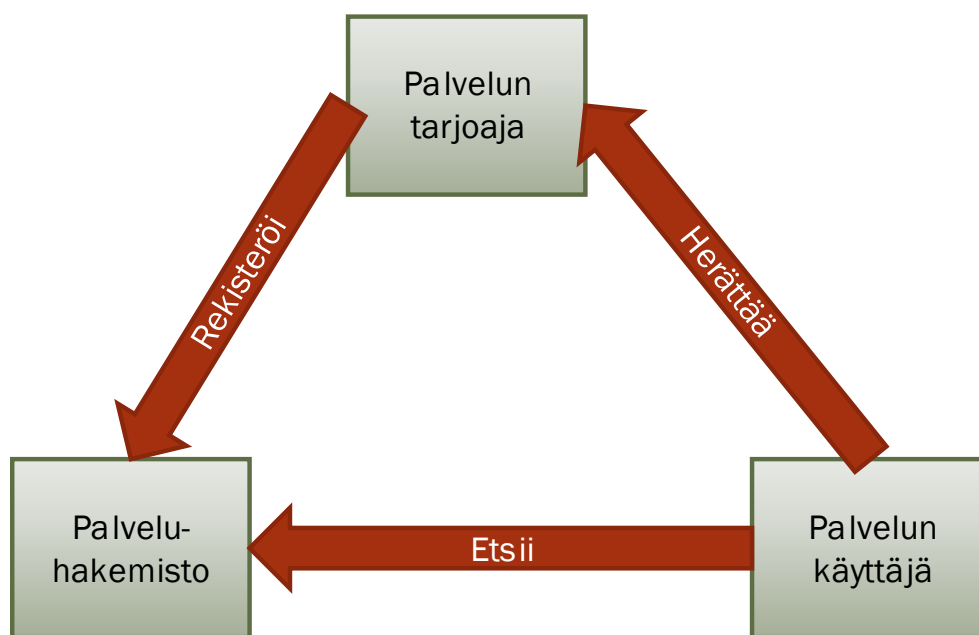
Arkkitehtuurin mahdollisesti tärkein lähtökohta on sen liiketoiminnallinen periytyminen. Arkkitehtuurin olemassaolon tarkoitus on tukea yrityksen liiketoimintaa. Arkkitehtuuri pitää sisällään malleja. Mallit taas kuvaavat ratkaisun liittyen tiettyyn joukkoon erilaisia vaatimuksia. Vaatimukset ovat sellaisia, joihin törmätään jatkuvasti. Arkkitehtuurin tarkoituksena on myös helpottaa ratkaisujen toteuttamista. Se ei siis vain kuvaa mitä järjestelmä tekee, vaan myös tarjoaa keinot miten järjestelmä ja sen komponentit tulisi rakentaa. Arkkitehtuuri toimii myös viestinnän pohjana. Se tarjoaa mekanismin ihmisille kommunikointiin sekä lisää yhteistä ymmärrystä tietojärjestelmistä. [20]

## 4.2 Palvelukeskeinen arkkitehtuuri

Palvelukeskeisen arkkitehtuuri (SOA) voidaan määritellä arkkitehtuuriseksi tyyliksi, jossa liiketoimintakeskeinen palvelu on ratkaisun oleellinen peruselementti [20]. Useimmat palvelukeskeisen arkkitehtuurin määritelmät keskittyvät sovelluksien toiminnallisuuksiin, jotka ovat löydettävissä ja uudelleen käytettävissä löyhästi kytkettyjen standardien rajapintojen kautta [20, 22, 23]. Yksi fokusoitu määritelmä on peräisin CBDI Forumilta: Palvelukeskeinen arkkitehtuuri sisältää kaikki ne toimintatavat, käytännöt ja viitekehykset, jotka mahdollistavat sovelluksen toiminnallisuuden ja ovat tarjottavissa ja hyödynnettävissä joukkona palveluita. Nämä palvelut on julkaistu siinä mittakaavassa, jotka ovat relevantteja palveluiden kuluttajille. Palveluiden kuluttajat voivat kutsua, julkaista ja löytää nämä palvelut sellaisella abstraktiotasolla yhden standardimuotoisen rajapinnan kautta, joka piilottaa palvelun teknisen toteuksen. [24] Palvelukeskeisen arkkitehtuurin kuvasi ensimmäisenä Gartner vuonna 1996 [25].

Palvelukeskeisen arkkitehtuurin perusrakenne on esitetty kuvassa 5. Ensin standardoitu palvelu rekisteröidään palveluhakemistoon, kuten UDDI:hin. Kun palveluun on päästy kiinni, niin palvelun kuluttaja tekee liipaisun palvelun tuottajalle. [22]

Palvelukeskeinen arkkitehtuuri on toimiva ratkaisu eri sovellusten integroimiseen, jotka on toteutettu erilaisilla teknologioilla [22]. Palvelukeskeinen arkkitehtuuri siis sopii hyvin, kun komponentit suoritetaan erilaisilla alustoilla ja toimittajan tuotteilla, sovellus pitää julkaista verkossa ja se on mahdollista paketoita Web Services -palveluiksi. [26] Yhtenä vaatimuksena on se, että sovelluksissa julkaistut rajapinnat ovat standardimuotoisia. [22].



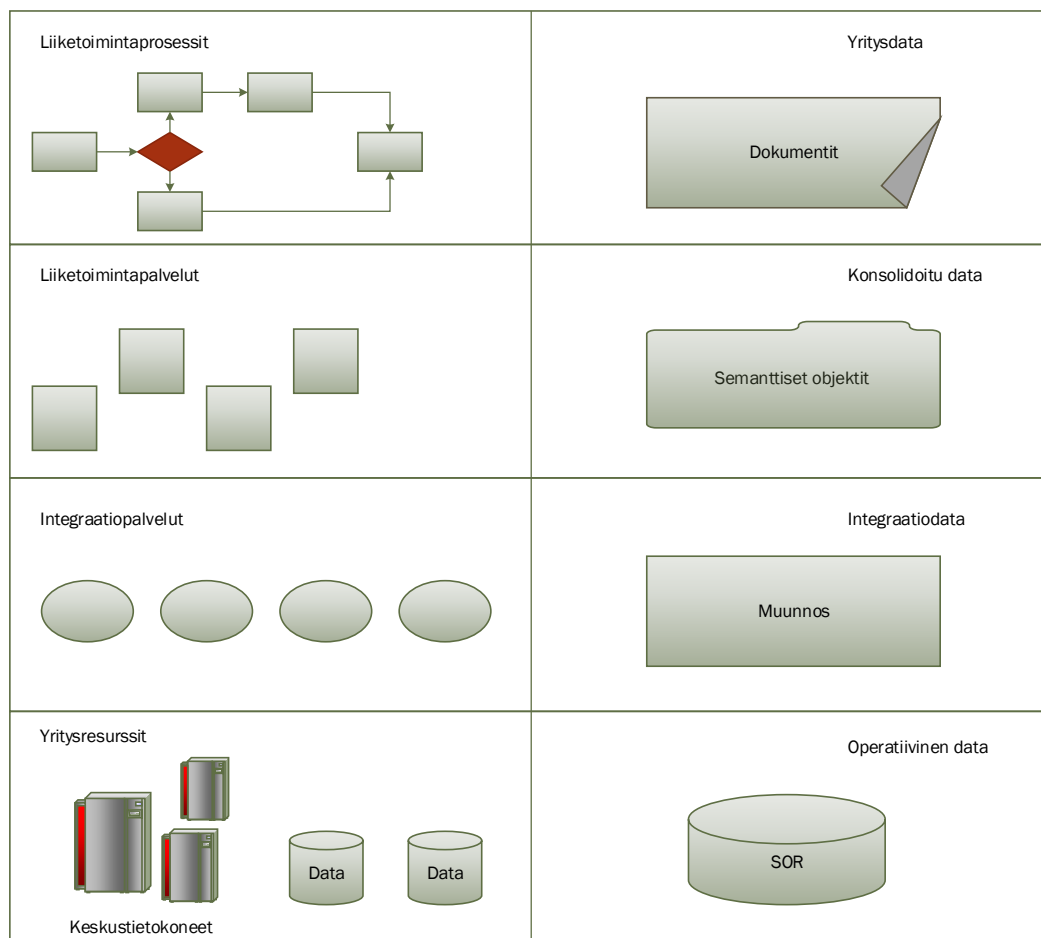
Kuva 5: SOA-malli. [22]

Kuva 6 esittää kerrostetun palvelukeskeisen arkkitehtuurin sisältäen kaksi eri käsitettä kerrosta kohden. Vasemmalla on toiminnalliset käsitteet, joita käytetään järjestelmien ja prosessien rakentamisessa. Oikealla puolella ovat tietokäsitteet, joilla välitetään, kuvaillaan tai muokataan dataa kullakin toiminnallisella tasolla. [20]

Kuvan 6 alin taso sisältää yrityksen olemassa olevat sovellukset tietojärjestelmät. Näissä sovelluksissa käsitellään yrityksen yksittäiset liiketoiminnalliset tapahtumat, jotka vaikuttavat yhden tai useamman tietueen lukemiseen, kirjoittamiseen tai muuttamiseen System of Record (SOR) -informaatiovarastossa. [20] Termi SOR viittaa sellaiseen tiedon tallennuspaikkaan, jossa jokaisesta tietoelementistä on olemassa ajantasaisin ja ainoaksi totuudeksi luokiteltava versio. [27]

Integraatiopalvelut tarjoavat järjestelmien väliset integraatiot ja pääsyn olemassa oleviin tietojärjestelmiin. Integraatio- ja liiketoimintapalveluiden eriyttäminen on tärkeää joustavan ympäristön ylläpitämiseksi. Tämä kerros yleensä muuntaa tiedon ja tarjoaa funktioita liiketoimintapalvelukerroksen tarpeisiin. [20]

Liiketoimintapalvelut tarjoavat korkean tason toiminnallisuuden koko yrityksen tasolla. Tämä taso tarjoaa palvelurajapinta-abstraktion, joka piilottaa sen alla olevat integraatiot. Tämä kerros myös rikkoo prosessien ja tietojärjestelmien riippuvuuden toisistaan. [20] Liiketoimintapalvelu voi esimerkiksi olla "TV-lähetysasemien hallinta" ja sillä erilaisia operaatioita: aseman haku nimellä, aseman haku kunnan perusteella, huollossa olevat asemat tai uuden aseman lisääminen. Kaikki operaatiot eivät välttämättä käsittele tietoa samassa tietojärjestelmässä, vaan operaatiot voivat kohdistua eri tietojärjestelmiin. Toisin sanoen, yhdellä operaatiolla voidaan hallita tietoa



Kuva 6: Palvelukeskeisen arkkitehtuurin elementit. [20]

useassa eri tietojärjestelmässä. Liiketoimintapalvelut operoivat semanttisia dataobjekteja, jotka ovat virtuaalista, eri palveluiden välillä jaettavaa ja välitettävää dataa. Semanttinen dataobjekti usein koostetaan eri olemassa olevien tietojärjestelmien tiedon pohjalta. [20]

Liiketoimintaprosessi koostuu sarjasta operaatioita, joita suoritetaan halutussa järjestyksessä. Liiketoimintaprosessi kuvataan usein Business Process Model (BPM)-mallilla, joka perustuu notaatioon Business Process Modeling Notation (BPMN). Prosessi suoritetaan sille tarkoitetulla ohjelmistolla (Business Process Management System, BPMS). Operaatioiden järjestämistä, valintaa ja suorittamista kutsutaan orkestroinniksi. [20]

Arkkitehtuuri pyrkii edistämään palvelukeskeisyyttä ratkaisun suunnittelussa, toteuttamisessa ja koostamisessa. Tarkemmin sanottuna, palvelukeskeinen arkkitehtuuri käsittelee toisistaan riippumattomia liiketoimintakeskeisiä palveluja, joita yhdistelemällä voidaan luoda korkeamman tason liiketoimintaprosesseja ja ratkaisuja yrityksen tarpeeseen. Palvelujen luonti ei ole palvelukeskeisen arkkitehtuurin haaste, vaan varsinainen arvo syntyy kun uudelleen käytettävät palvelut yhdistellään

ketteriksi ja joustaviksi liiketoimintaprosesseiksi. [20]

Arkkitehtuurista on käytävä ilmi mitkä ovat ratkaisun tärkeät osat, mitkä ovat näiden osien väliset suhteet ja miten ne yhdistetään ylemmälle tasolle arvoa tuottavaksi kokonaisuudeksi. Palvelukeskeisessä arkkitehtuurissa tärkeät osat ovat prosessit, palvelut, integraatiot, olemassa olevat järjestelmäsovellukset, dokumentit, semantiikka, tietomuunnokset ja tietoliikenne. [20]

IBM developerWorks esittää useita erilaisia näkökulmia palvelukeskeiseen arkkitehtuuriin. Ensimmäiseksi, palvelujen luonnissa kehitetään sovelluksia, jotka tarjoavat halutun liiketoimintalogiikan palveluna. Sovellusten luonnissa myös uudelleen käytetään olemassa olevaa liiketoimintalogiikkaa, joka on julkaistu palveluna. Toiseksi, palveluiden on oltava vuorovaikutuksissa toisiinsa ja toimittava yhdessä. Kolmas näkökulma on palvelukeskeisen arkkitehtuurin hallinta (SOA Governance), jolla hallitaan palveluiden kehittämiseen ja käyttämiseen liittyviä prosesseja. Sillä määritellään palveluiden kehittämisprosessit, -työkalut, palveluiden omistajat sekä kuka, miten ja milloin palvelua voi käyttää. [28]

Toinen näkökulma on moduulien yhdistäminen. Se kuvaa prosessia, jossa pienistä ja erillisistä moduuleista koostetaan suurempi ja tarkoituksenmukaisempi moduuli. Palvelukeskeisessä arkkitehtuurissa on kolme pääteknologiaa, jolla moduulien yhdistäminen on toteutettavissa: Service Component Architecture (SCA), Business Process Execution Language (BPEL) ja Message Driven (MD). SCA-moduulin komponentit on toteutettu eri ohjelmointikielillä erilaisille alustoille ja ne voidaan ottaa käyttöön tietoverkon eri solmuissa. BPEL on prosessi-integroitimenetelmä, joka keskittyy kokoamaan erilaiset palvelut yhdeksi palveluketjuksi. MD käyttää taas sanomanvälitysmallia ja sanomanvälitykseen tarvittavaa väliohjelmistoa löyhästi yhteen kytkettyjen ympäristöjen rakentamiseen ylempää ohjelmistokerrosta varten. [6] Tässä työssä palveluiden yhdistämistä toteutettiin pääasiassa MD-pohjaisella lähestymistavalla.

Palvelukeskeinen arkkitehtuuri mahdollistaa eri organisaatioiden kehittävän toisistaan riippumattomasti palveluja, jotka voidaan edelleen yhdistellä liiketoimintaprosesseiksi ja järjestelmiksi. Tällöin vaaditaan, että palvelut ovat samankaltaisia kooltaan, muodoltaan ja toiminnallisuuksiltaan. Palveluiden on myös noudatettava yhteisiä standardeja. Niiden on myös kommunikoitava keskenään sekä teknisellä, että semanttisella tasolla. Palveluiden vastuualueissa ei saa olla puutteita tai palveluiden toiminnallisuudet eivät saa olla päällekkäisiä. [20]

Teknologia itsessään ei ratkaise liiketoiminnan ongelmia ja parhaimmillaankin teknologia voi toimia vain mahdollistajana. Teknologian, liiketoiminnan ja organisaation toimiva yhdistelmä tarvitaan aina liiketoiminnallisten tavoitteiden saavuttamiseksi. Tämä sama pätee myös palvelukeskeiseen arkkitehtuuriin. SOA voi tarjota ketterän tavan korkean tason prosessien tuottamiseen perusliiketoimintapalveluja yhdistelemällä. Se ei kuitenkaan riitä, vaan rinnalla oltava liiketoimintamalli, jota ilman arvon tuottaminen ei ole mahdollista. Palvelukeskeistä arkkitehtuuria on mahdollista tuottaa ilman liiketoimintamallia, mutta tuloksena voi olla joukko yhteensopimattomia palveluja. Mitä enemmän liiketoiminnan tavoitteita, liiketoimintastrategiaa ja liiketoimintaprosesseja tunnetaan, sitä paremmin voidaan tuottaa niitä tukevia palveluita. Ilman tätä ymmärrystä on riski, että tehdään turhaa työtä, palveluiden

uudelleentoteutuksia tai tarpeetonta palveluiden replikointia. [20]

Tämän tyyppinen arkkitehtuuri on välinpitämätön millaista laitteistoa, ohjelmistoa, ohjelmointikieltä tai käyttöjärjestelmää käytetään. Sen vuoksi palvelukeskeisillä arkkitehtuureilla on mahdollista ratkaista hajautettujen järjestelmien ja ohjelmistojen integraatioiden toteuttamiseen liittyviä haasteita. Heterogeenisten järjestelmien tieto ja palvelut on mahdollista integroida standardien rajapintojen kautta. Organization for the Advancement of Structured Information Standards (OASIS) esittää yksityiskohtaisen referenssimallin palvelukeskeiseen arkkitehtuuriin. [29]

Yksi ohjelmistoarkkitehtuurin ja palvelukeskeisen arkkitehtuurin eroavaisuus on niiden laajuus. Palvelukeskeinen arkkitehtuuri huolehtii yhdenmukaisen palvelun tarjoamisesta yritystasolla siten, että palvelu on eri ryhmiin kuuluvien sovellusten käytössä. Palvelukeskeisen arkkitehtuurin tehtävänä on edistää liiketoiminnan kyvykkyyksiä tavalla, jossa ne ovat usean liiketoimintaprosessin käytössä. [20] Palvelukeskeisessä arkkitehtuurissa painotus on ohjelmistoarkkitehtuurin taustajärjestelmässä. SOA-palveluiden toiminnalliset vaatimukset liittyvät toimintotasoon (business logic), liiketoimintaresursseihin ja liiketoiminnallisen datan käsittelyyn. [22]

SOA muuttaa ohjelmistonkehitystä tuotekeskeisestä valmistuksesta kuluttajakeskeiseen palveluiden koostamiseen. SOA-palveluiden koostamisteknologiat, kuten BPEL ja WSCI, uudet liiketoimintaprosessit, -sovellukset ja -ratkaisut voidaan rakentaa suhteellisen nopeasti ja kustannustehokkaasti koostamalla hajautettuja palveluita heterogeenisessä ympäristössä. [20]

Nykyinen palveluiden koostamisen ohjelmointimalli ja työkalut ovat pääasiallisesti tarkoitettu ammattimaisille SOA-kehittäjille. Vaikka käytössä olevat tekniikat ovatkin tehokkaita osoittamaan yrityksen SOA-haasteet, niin kolme selkeää ongelmaa on silti olemassa. Ensimmäiseksi, kehittäjän on hallittava tyypillisesti useita eri SOA-teknologioita ja -työkaluja. Lisäksi nämä työkalut vaativat usein merkittäviä investointeja sekä laitteisto-, että sovellusinfrastruktuuriin. Toiseksi, nämä työkalut eivät mahdollista lennossa tehtävien palvelukokoonpanoon tehtäviä kustomointeja. Palvelun suunnittelu, kehitys ja testaus on tehtävä ensin kehitysympäristössä ennen kuin käyttöönotto suoritussympäristöön voidaan tehdä. Kolmanneksi, nämä teknologiat eivät tue palveluiden koostamista hyvin silloin, kun Web Services -rajapintoja ei ole tarjolla. Nämä asiat ovat tulleet ongelmaksi Web 2.0 paradigman yhteydessä. [30]

Tämän hetkisen näkemyksen mukaan palvelukeskeinen arkkitehtuuri ei hävitä semanttisen heterogeenisyyden ongelmaa. Bishr [31] määrittelee semanttisen heterogeenisuuden reaali maailman asioiden erilaisina kuvauksina tietokannassa. Jotta on mahdollista vastata käyttäjien odotuksiin tietopyynnöissä, on palveluiden tarjoajien ja käyttäjien semanttiset tulkinnat koordinoitava tehokkaasti. Vasta kun koordinointi on tehty, tiedon manuaalinen tai automaattinen käsittely voi johtaa tarkoituksenmukaisiin tuloksiin [32].

#### 4.2.1 Palvelu

Palvelukeskeisen arkkitehtuurin peruskäsite on palvelu. Palvelu voidaan määrittellä erilliseksi yksiköksi osana liiketoiminnan toiminnallisuutta, joka tarjotaan saataville palvelusopimuksen kautta. Palvelusopimus määrittelee kaiken vuorovaikutuksen

palvelun tuottajan ja kuluttajan välillä. Sopimus sisältää palvelun rajapinnan, rajapintadokumentaation, palvelun toimintaperiaatteen, palvelun laadun (QoS) ja suorituskyvyn. [20]

Yksi tärkein ero palvelun ja muun ohjelmistorakennelman välillä on se, että palvelu on selkeästi hallittu. Palvelun laatu ja suorituskyky hallitaan palvelutasosopimuksella (SLA). Lisäksi koko palvelun elinkaari kuuluu hallinnan piiriin: suunnittelu, käyttöönotto, parannukset ja ylläpito. [20]

Palvelua voidaan tarkastella kahdesta eri näkökulmasta. Palvelulla on sen rajapinta sekä toteutus. Palvelussa on tyypillisesti erotettu nämä kaksi asiaa selvästi toisistaan. Palvelun rajapinta määrittelee palvelun toiminnot, eli mitä palvelu todellisuudessa tekee. Toiminnot sisältävät palvelun sisään ja ulos välittämät parametrit sekä protollat miten toimintoja käytetään ja tarjotaan. Palvelun toteutus taas voi perustua käytössä oleviin tietojärjestelmiin, muiden palveluiden yhdistelyyn ja orkestrointiin, varta vasten kirjoitettuun ohjelmakoodiin tai kaikkiin edellä mainittuihin yhtä aikaa. Palvelun kuluttaja näkee palvelun rajapinnan, ei koskaan miten palvelu on toteutettu. Palvelun tarjoaja taas on vapaa muuttamaan palvelun toteutusta, kunhan palvelun rajapinta pysyy muuttumattomana. [20]

#### 4.2.2 Soveltaminen paikkatietojärjestelmissä

Usein käytännön tilanteessa paikkatieto yksin ei ole riittävää, vaan käyttäjä haluaa yhdistää paikkatietoa paikkatiedottomaan dataan. Palvelukeskeiseen arkkitehtuuriin perustuva tietojärjestelmä tai ohjelmistotyökalu on hyvä malli tietojen integroimiseen keskenään. [12] Palvelukeskeistä arkkitehtuuria on pidetty erittäin lupaavana arkkitehtuurina seuraavan sukupolven paikkatietojärjestelmien käytännön toteutuksiin [5].

Alkuaikojen käytännöllisten ja joustavien paikkatietojärjestelmien suunnittelussa ja toteuttamisessa kohdattiin haasteita. Paikkatietomäärän lisääntyessä eri tietolähteitä piti integroida järjestelmään entistä enemmän. Perinteisillä paikkatietojärjestelmillä oli omat sisäiset datamuotonsa, jolloin datamuoto piti muuntaa oikeaksi jo ennen tiedon siirtämistä järjestelmään. Samasta datasta tehtiin siis lukuisia eri kopioita eri paikkatietojärjestelmiin. Toiseksi, Web Services -palvelujen nopean kehityksen vuoksi yhä useampi paikkatietojärjestelmä toteutettiin internetiin. Palvelujen ja datan laatu vaihtelivat eri toimijoiden välillä, samoin myös palveluiden rajapinnat. Näistä syistä johtuen internetin paikkatietojärjestelmät olivat alustavia, yksinkertaisia ja pienehköjä, jotka eivät vastanneet käyttäjien tarpeisiin. Haasteena oli yhdenmukaistaa palveluiden rajapinnat ja koostaa pienistä palveluista kokonaisia ja käytännöllisiä paikkatietojärjestelmiä. Kolmanneksi, internetissä olevan paikkatiedon ja paikkatietopalveluiden pohjalta syntyneet paikkatietojärjestelmät olivat paljon monimutkaisempia kuin aiemmin. Monimutkaisen järjestelmän piti samalla kuitenkin olla julkinen ja helposti käytettävä ratkaisu. [6]

Palvelukeskeisen arkkitehtuurin yleistymisen on tehnyt hajautetut paikkatietojärjestelmät entistä palvelukeskeisimmiksi. Palvelukeskeinen arkkitehtuuri yhdistelee eri paikkatietojärjestelmäpalveluita ja esittää eri lähteistä peräisin olevaa erilaista dataa. [22]

Paikkatietojärjestelmien tullessa avoimiksi ja verkottuneiksi, on moduulit ja data sijoitettu eri solmuihin hajautetussa ympäristössä. Samanaikaisesti palvelukeskeinen arkkitehtuuri on hajautettujen ohjelmistojärjestelmien rakentamiseen tähtäävä metodologia. Paikkatietojärjestelmien ja palvelukeskeisen arkkitehtuurin yhdistäminen saattaa tuoda uusia ajatuksia ja malleja. [6]

Palvelukeskeisen arkkitehtuurin vaikutuksia paikkatietojärjestelmiin voidaan tarkastella kahdesta näkökulmasta. Ensimmäinen niistä on moduulien kapselointi. Kapseloinnilla tarkoitetaan eri paikkatietojärjestelmien moduulien vuorovaikutusta, vaikka niitä suoritetaan eri alustoilla ja käyttävät erilaisia datamuotoja. Kapselointi tarjoaa oikeamuotoisen rajapinnan, joka toimii moduulien uudelleenkäytön perustana. Kun rajapintana käytetään Web Services -teknologiaa, kuvaa WSDL palvelun rajapinnan ominaisuudet. Kapseloinnilla siis saavutetaan moduulien uudelleenkäytettävyys sekä maantieteellisen datan jakaminen. [6] Web Services ja WSDL on kuvattu tarkemmin luvussa 5.

Erilaiset moduulien yhdistämistavat tarjoavat erilaisia menetelmiä paikkatietojärjestelmien rakentamiseen. Paikkatietojärjestelmän toiminnot usein koostuvat useammasta muusta toiminnosta, joilla on sidonta toisiinsa. Esimerkiksi kartan zoomaus ja liikuttaminen ovat itsenäisiä toimintoja, mutta niillä on maantieteellinen sidonta keskenään. Tällaisissa tapauksissa SCA:n avulla voidaan koota erilaiset moduulit yhteen ja tarjota yhtenäinen käyttöliittymä. [6]

Maantieteelliset prosessit ovat yleisiä paikkatietojärjestelmässä. Esimerkiksi reitin haku koostuu lähtöpisteen ja päätepisteen koordinaattien selvittämisestä, reitin laskemisesta pisteiden välillä sekä reitin piirtämisestä kartalle. Maantieteellinen prosessi on sarja riippumattomia palveluita tai moduuleita. Näissä tapauksissa BPEL on sopiva työkalu koostamaan erilaiset palvelut tai moduulit tarkoituksenmukaiseksi prosessiksi. [6]

Monissa paikkatietojärjestelmissä jokin maantieteellinen tapahtuma laukaisee jonkin toiminnon tai prosessin käyntiin. Tapahtuman tuottaja on aina riippumaton tapahtuman vastaanottajasta, mutta silti perinteisessä paikkatietojärjestelmässä osapuolet ovat tiukasti yhteen kytkettyjä. MD muuttaa kytkennän löyhäksi, jolloin eri moduuleja voidaan kehittää toisistaan irrallisina. [6]

SOA on kehitetty hajautettujen yritystason sovellusten rakentamiseen, joten on olemassa useita paikkatietojärjestelmäkohtaisia haasteita, joihin SOA ei tarjoa ratkaisua. Ensimmäiseksi, SOA auttaa tiedon jakamisessa ja kapseloinnissa, mutta joissakin tilanteissa pitää siirtää suuret määrät paikkatietoa verkon yli. Näissä tapauksissa SOA ei ole tehokas vaihtoehto. Paikkatietoanalyysi ja erilaiset laskennat tuottavat helposti suuren määrän paikkatietoa. Toiseksi, joissakin tilanteissa halutaan saada tilatietoa tai välituloksia maantieteellisten prosessien suorituksen ajalta. SOA-palvelut ovat aina tilattomia, joten tilaongelma pitää huomioida palveluiden suunnittelussa erityisellä tavalla. Kolmanneksi, SOA ei ole sopiva suorituskykykriittisiin tai tietoturvakriittisiin ympäristöihin. [6]

Palvelukeskeisen arkkitehtuurin ja paikkatietojärjestelmien yhdistämisessä on kaksi tutkimuksellista näkökulmaa. Ensimmäinen niistä keskittyy tutkimaan miten Web Services -teknologiaa käytetään paikkatietojärjestelmissä. Tämän tyyppinen tutkimus käyttää aina Web Services -rakennetta, joka koostuu palvelun tuottajas-



ta, palvelurekisteristä ja palvelun käyttäjästä. SOA ei ole kuitenkaan sama asia kuin Web Services, joten tämän tyyppinen tutkimus ei oikeastaan yritäkään soveltaa palvelukeskeistä arkkitehtuuria paikkatietojärjestelmiin. Toinen näkökulma pyrkii keskittymään suurten paikkatietojärjestelmien suunnitteluun. Tämän kaltainen tutkimus yrittää hyödyntää palvelukeskeisen arkkitehtuurin ydinmalleja, kuten palveluiden luokittelua tai palveluväylää makroskooppisen arkkitehtuurin suunnittelussa koko paikkatietojärjestelmälle. Tutkimuksessa ei oteta kantaa yksityiskohtiin tai toteutukseen, vaan pysytellään kuvailevalla ja käsitteellisellä tasolla. [6]

Akateeminen maailma ja teollisuus ovat vuosia väitelleet onko paikkatietojärjestelmillä tietomalleja, jotka poikkeavat muista tietojärjestelmistä [33]. Tämä on aiheuttanut hankaluuksia paikkatietojärjestelmien yhteensopivuudessa. OGC on keskittynyt edistämään paikkatietojärjestelmien yhteensopivuutta jo syntyaajoistaan lähtien. Sen ehdotuksesta on syntynyt palvelustandardit Web Map Service (WMS) ja Web Feature Service (WFS). [13] Lisäksi Web Processing Service (WPS) määrittelee standardin rajapinnan paikkatietoprosessien julkaisemiseen ja löytämiseen [34]. Kriittisesti tarkasteltuna nämä OGC-palvelut eivät ole välttämättä täysin linjassa SOAP- tai WSDL-standardien kanssa. Tämä voidaan nähdä yhtenä syynä, miksi keskustelu paikkatietojärjestelmien yhteensopivuudesta rajoittuu vain paikkatietoon. Tämä keskustelu on toki perusteltua, koska jatkuva maantieteellisen datan, tiedon ja palveluiden integraatio on tavoiteltavaa. Palvelukeskeisen arkkitehtuurin tuomat edut eivät kuitenkaan rajoitu vain paikkatietoon, vaan se mahdollistaa palveluiden vuorovaikutuksen myös paikkatietokentän ulkopuolelle. [13]

### 4.3 Representational State Transfer (REST)

REST on HTTP-protokollaan perustuva arkkitehtuurimalli, joka määrittelee World Wide Webin laadulliset ominaispiirteet. WWW:tä pidetään avoimena, löyhästi kytkettynä ja erittäin hajautettuna hypermediajärjestelmänä. REST on kokenut viime vuosina suosion kasvua, sillä sitä on pidetty laajasti yksinkertaisempänä ja web-tyyppisempänä tapana julkaista palvelurajapintoja. Vertailukohtana on ollut erityisesti monimutkaisemmat ja raskaammat SOAP ja Web Services sekä RPC-lähtöiset eli proseduurien etäkutsuihin keskittyneet protokollat. REST-arkkitehtuurimalli edistää korkeaa suorituskykyä ja skaalautumista helpottamalla palvelinpäässä ja välikerroksissa tehtävää replikointia. [35]

REST käsitteenä tuli käyttöön 2000-luvun alussa, kun WWW käsitteellistettiin löyhästi kytketyksi hajautetuksi hypermediajärjestelmäksi. Muutama vuosi tämän jälkeen tuli selväksi, että REST olisi varteenotettava kilpailija mallille, jossa Web Services pohjautuisivat SOAP:iin. REST-malliin pohjautuvia Web Serviceihin viitataan termillä RESTful Web Services. [36]

Yksinkertaistettuna REST on joukko hypermediajärjestelmän suunnittelun rajoituksia. Näistä rajoituksista syntyy arkkitehtuuri, joka mahdollistaa hyvän skaalautuvuuden, mashup-kyvykkyyden, käytettävyyden ja saatavuuden. REST on erityisesti hyväksytty käyttöön tilanteissa, joissa tietojärjestelmän nykyisten ja tulevien komponenttien suunnittelun keskitetty hallinta puuttuu. REST siis johtaa ratkaisuihin, jotka ovat vähemmän tiukasti kytkettyjä, kuin vakiintuneimmissa arkkitehtuureissa.

[36]

REST:in ytimenä voidaan pitää seuraavia asetettuja vaatimuksia: [36]

- Kaikki sovelluksen ja sen tilan kannalta tärkeät resurssit pitää olla tunnistettavissa yksilöllisesti ja pysyvästi. Resurssina voidaan pitää mitä tahansa sovelluksen käsittelemää informaatiota, esimerkiksi transaktiodokumenttia eli vaikkapa tilausta.
- Kaikki vuorovaikutus tapahtuu yhtenäisen rajapinnan kautta, joka tukee vuorovaikutusta tarjoamalla toiminnallisuudeltaan riittävän joukon metodeita. RPC ei tarjoa yhtenäistä rajapintaa, vaan joukon kutsuttavia metodeita. REST:issä taas julkaistaan resurssi, johon tapahtuu vuorovaikutus aina samanlaisen rajapinnan kautta.
- Vuorovaikutukseen käytettävien sanomien pitää olla itsensä selvästi kuvaavia. Resurssin ilmentymän pitää olla sellainen, että osapuolet ymmärtävät resurssin ja sen tilan. Toisin sanoen, resurssin ilmentymää pitää pystyä muuttamaan ilman, että resurssista pitäisi tietää jotain mikä ei tule itse resurssin ilmentymästä esille.
- Välitettävät ilmentyvät voidaan linkittää toisiin ilmentymiin siten, että niiden semantiikka on määritelty ilmentymässä. Linkityksen kautta on mahdollista päästä kiinni muihin sovelluksen resursseihin. Toisin sanoen, asiakassovelluksen ei tarvitse tietää verkkosovelluksesta mitään etukäteen.
- Interaktiot asiakas- ja palvelinohjelmiston välillä ovat tilattomia eli kaikki interaktiot ovat toisistaan riippumattomia. Palvelinpäässä ei tallenneta tietoa asiakassovelluksen tilasta eli asiakas- ja palvelinohjelmiston välille ei synny sessiota. Tämä tärkeä rajoite mahdollistaa palvelinpään skaalautuvuuden.

## 4.4 Mashup

Mashup on interaktiivisten web-sovellusten lajityyppi, jossa sovellus käyttää hyväkseen erilaisia ulkoisia tietolähteitä uusien innovatiivisten palveluiden luonnissa. Mashuppeja voidaan pitää toisen sukupolven web-sovellusten tunnusmerkkeinä. Tätä myös kutsutaan epävirallisella nimellä Web 2.0. Erilaisia sovellustyyppejä voivat olla kartta-, video- ja valokuva-, haku- ja ostos- sekä uutis-mashupit. Yksi suurimpia mashuppien edistäjiä karttapuolella oli Googlen esittelemä Google Maps API, joka mahdollisti web-kehittäjille kaiken paikkatiedon esittämisen kartalla. Pian tämän jälkeen vastaavanlaisen API:n tarjosi Microsoft, Yahoo ja AOL. [37]

Arkkitehtuurisesti mashup koostuu kolmesta eri osapuolesta, jotka ovat loogisesti ja fyysisesti erillisiä. Ohjelmointirajapinta tai muu sisällön tarjoaja, mashup-sivusto ja WWW-selain. Datan haun helpottamiseksi datan tarjoajat julkistavat sisältöään web-protokollilla, kuten Web Services, REST tai RSS/Atom. Kuitenkaan kaikkea kiinnostavaa sisältöä ei ole tällä tavalla julkaistu ohjelmointirajapinnan kautta. Tässä tapauksessa on käytetty web scraping -menetelmää, jossa tietoa on kaivettu suoraan sisällön tarjoajan sivustolta. [37]

Mashup-sivusto pitää sisällään mashup-logiikan, mutta sen suoritus voi tapahtua joko palvelinpäässä tai selainpäässä. Mashupit voidaan toteuttaa perinteisten web-sovellusten tapaan käyttämällä palvelinpään dynaamisia sisällöngenerointitekno- logioita kuten Java servlets, CGI, PHP tai ASP. Vaihtoehtoinen tapa on generoida sisältö selainpäässä JavaScriptin tai sovelmien avulla. Tällä lähestymistavalla on mahdollista toteuttaa rikkaita internet-sovelluksia (Rich Internet Application, RIA), joiden avulla päästään lähemmäksi interaktiivista käyttäjäkokemusta. [37]

Osittain palvelukeskeisen arkkitehtuurin haasteista ja sopimattomuudesta Web 2.0 paradigmaan on syntynyt uusia teknologioita. Webistä on tullut sekä ohjelmis- tojen suunnittelu-, että ajoympäristö. Muutoksia voidaan tehdä lennossa, jolloin palvelukokoonpanoihin tehtävät muutokset reagoivat nopeammin käyttäjän tarpeissa tapahtuviin muutoksiin. Lisäksi on ollut tarve päästä kiinni nykyisiin verkossa oleviin sovelluksiin ja dataan. [30]

Liu ym. ovatkin sitä mieltä, että mashup on konseptina yksinkertaisempi, kustan- nustehokkaampi ja tarjoaa itsepalvelutyypin lähestymistavan palvelujen koostami- selle. Näillä päästään eroon SOA:n monimutkaisuudesta ja rajoituksista palvelujen koostamisessa. [30]

## 4.5 Yhteenveto hajautettujen järjestelmien arkkitehtuuris- ta

Ohjelmistoarkkitehtuuri kuvaa järjestelmän pääkomponentit, komponenttien suhteet toisiinsa ja niiden välillä välitettävän informaation. Palvelukeskeinen arkkitehtuuri (SOA) on yksi ohjelmistoarkkitehtuurin malli, jossa peruselementtinä on palvelu. SOA on kehitetty erityisesti hajautettujen järjestelmien arkkitehtuurimalliksi ja se soveltuu hyvin myös paikkatietojärjestelmiin. Palvelukeskeisen arkkitehtuurin kom- ponentit ovat palvelun tuottaja, palvelun kuluttaja ja palvelurekisteri. Palveluja tuottamalla ja yhdistelemällä voidaan toteuttaa integraatioita ja tarjota hyötyjä liiketoimintaprosessitasolle. Palvelukeskeisen arkkitehtuurin rinnalle ovat tulleet mm. REST-arkkitehtuurimalli ja mashup-sovellukset, jotka pyrkivät myös tuottamaan vastaavanlaisia hajautettuja järjestelmiä. Yritysovelluksissa palvelukeskeinen arkki- tehtuuri antaa hyvän mallin järjestelmien kehittämiselle ja integraatioille standar- deilla. Keskitetyn hallinnan puuttuessa, kuten internetissä, REST on osoittautunut toimivaksi arkkitehtuurimalliksi hajautetuille järjestelmille.

## 5 Web Services -teknologiat

Tässä luvussa esitellään standardeja Web Services -teknologioita, joita sovelletaan palvelukeskeisen arkkitehtuurin mukaisissa hajautetuissa järjestelmissä. Tässä luvussa syvennytään myös lisää W3C:n XML-standardeihin, sekä XML:ään pohjautuviin tiedonvälitysprotokolliin. Luvun viimeisessä osassa esitellään palvelukeskeisen arkkitehtuurin perusrakenteeseen kuuluva UDDI-palvelurekisteri.

### 5.1 XML-standardit

Tässä luvussa esitellään W3C:n XML-standardit XML Schema, XPath ja XQuery, jotka ovat olleet merkittävässä roolissa tässä työssä. XML-skeema kuvaa XML-dokumentin rakenteen ja sitä hyödynnetään skeemamuunnoksissa kuvaamaan tiedon lähtö- ja kohdemuoto. Lähtö- ja kohdemuotojen välille tehdään skeemakuvaus, jonka toteuttamiseen voidaan käyttää XPath- ja XQuery-standardeja. Niiden avulla tietoa poimitaan lähteenä olevasta XML-dokumentista ja muodostetaan skeeman mukainen kohdedokumentti.

#### 5.1.1 XML Schema Definition Language (XSD)

XML Schema Definition Language (XSD) on W3C:n kehittämä mallinnuskieli, joka määrittelee XML-sanaston. Uusien XML-pohjaisten kielten luonti voidaan toteuttaa rakentamalla uusi XML-skeema. XML-skeema sisältää runsaan joukon mallinnuselementtejä sekä rakenteen, että tietotyyppien käyttöön XML-dokumentissa. Myös XML-skeemadokumentit on koodattu XML-muotoon [4, 38]. Koska XML Schema on myös itsessään XML-dokumentti, on sitä mahdollista parsia ja manipuloida, kuten mitä tahansa XML:ää [39]. XML-skeemaan voidaan viitata XML-dokumentti-instanssista, jolloin XML-jäsenin voi dokumenttia lukiessaan tehdä myös validointitarkastuksen [4].

XSD on kieli, joka tarjoaa välineet kuvaamaan XML-dokumentin rakenteen ja sen rajoitukset. Skeeman tarkoituksena on määrittellä ja kuvata XML-dokumentti käyttäen skeeman komponentteja. Niillä määritellään ja dokumentoidaan XML:n olennaisten osien tarkoitus, käyttö ja keskinäisen suhteet. Osilla tarkoitetaan tietotyyppisiä, elementtejä ja niiden sisältöjä sekä attribuutteja ja niiden arvoja. Skeema voi myös määrittellä dokumentin lisäominaisuuksia, kuten elementtien ja attribuuttien arvojen normalisointia tai oletusarvoja. [38]

XML Schema -kieltä käytetään XML-dokumentin sallitun rakenteen, sisällön ja rajoitusten kuvaamiseen. Ajatus XML Schemasta on hyvin samanlainen, kuin Data Definition Language (DDL) relaatiotietokannoissa. Relaatiotietokannassa DDL:llä luodaan tietokantataulu ja määrätään sen säännöt ja rajoitukset. Skeemakieli tarjoaa parannetut, kokonaisvaltaisemmat ja voimakkaammat ominaisuudet kuin Document Type Definition (DTD), joka on ollut perinteinen mekanismi XML-dokumenttien rakenteen ja sisällön kuvaamiseen. [39]

XML Schema tarjoaa rikkaan datatyyppityksen tyyppillisten ohjelmointikielten tapaan. Valmiit datatyyppit löytyvät merkkijonolle, kokonaisluvulle, päivämäärälle ja

niin edelleen, mutta kehittäjä voi myös luoda omia datatyyppejä. XML Schema tukee periytymistä. Uusia skeemoja voidaan luoda ja ne voivat periä ominaisuuksia olemassa olevilta skeemoilta. Skeema voidaan pilkkoa erillisiin komponentteihin. Skeemoja kirjoittaessa on tällöin mahdollista viitata näihin valmiisiin komponentteihin. XML Schema -kieli on myös tiukasti integroitu XML-nimiavaruuksien kanssa tekemällä elementtien ja attribuuttien luomisen nimiavaruuteen helpoksi. Nimiavaruuden avulla voidaan erottaa toisistaan samannimisiä elementtejä ja attribuutteja. [39]

XML-skeeman käyttö ohjelmistoissa tekee XML-dokumenttien validoinnin helpoksi. Skeema voi mallintaa monimutkaista informaatiota tehokkaasti. Sen mallintamistapa on myös elegantti, joka tekee skeemoista ihmisluettavan. [39]

Riippumattomat tietolähteet ovat usein heterogeenisiä, vaikka niissä käsiteltäisiinkin samaan aihealueen liittyvää tietoa. Huolimatta haasteista heterogeenisen tiedon käsittelyssä, tarve integroida heterogeenista tietoa monimuotoisista tietolähteistä on kasvanut. Integrointia tarvitaan silloin kun kaksi tai useampi organisaatio yhdistyvät. Viime aikoina verkkosivut ovat linkittyneet erilaisiin tietolähteisiin, jolloin integraatiota näihin tietolähteisiin on vaadittu. Integraatiojärjestelmien tehtävänä on tarjota yhtenäinen rajapinta tietolähteisiin. Integraation yhtenä etuna on vapauttaa käyttäjä etsimästä ja olemasta vuorovaikutuksessa kaikkien tietolähteiden kanssa. [40]

Skeemat ja skeemakuvaukset ovat kaksi peruskomponenttia integraatioissa. Skeemat kuvaavat rakenteen ja jossain määrin myös tiedon semantiikan tietolähteessä. Skeemakuvauksia taas voidaan käyttää tiedon muuntamiseen kahden skeeman välillä tai kyselyn muuntamisessa skeemasta toiseen. Mäppäyksiä voidaan erityisesti käyttää välitysskeeman muuntamisessa tietolähteiden paikallisiin skeemoihin. Skeemakuvaukset ovat usein määriteltä korkeatasoisesti ja muodollisesti ja ne kuvaavat eri skeemojen vastaavuudet loogisella tasolla. Integraatioiden avainkysymys on nimenomaan skeemojen integraatio. [40]

Skeemaintegraatio sisältää globaalin ja paikallisen skeeman luomisen ja näiden välille tehtävän kuvauksen. Skeemaintegraatiossa itse tietoon ei kosketa, vaan eri tietolähteiden skeemoja yhdistetään toisiinsa. Paikallinen skeema luodaan tietojärjestelmän relaatiomallin mukaiseksi. Globaalin skeeman luomisessa taas on huomioitava paikallisten skeemojen ominaisuudet. Globaalissa skeemassa ratkaistaan semanttiset ristiriidat eri paikallisten skeemojen välillä. [40]

Globaali skeema tarjoaa yhtenäisen ja virtuaalisen näkymän kysellä tietoa heterogeenisista tietolähteistä. Vastatakseen käyttäjien kyselyihin, järjestelmä käyttää semanttisia kuvauksia yhtenäisen rajapinnan ja tietolähteiden paikallisen skeeman välillä. Semanttiset kuvaukset voivat sisältyä kuvaustaulukon tyyppiseen tietorakenteeseen. Kuvaustaulu pohjalta tehdään käyttäjältä tulevien kyselyiden ja paikallisten tietolähteiden välillä tapahtuvat skeemamuunnokset. [40] Luvussa 6 kerrotaan tarkemmin integraatioista, joissa kaikissa skeemamuunnokset näyttävät suurta roolia.

### 5.1.2 XPath

XPath eli XML Path Language on kieli, jolla voidaan käsitellä XQuery- ja XPath-datamallin mukaisia arvoja. Datamalli tarjoaa puumaisen esityksen XML-dokumenteista

ja niiden atomaarisista arvoista, kuten kokonaisluvuista ja merkkijonoista. XPath-lausekkeen tuloksena voidaan saada syötteenä olleesta XML-dokumentista valikoima solmuja tai atomaarinen arvo. Yleisemmin, XPath-lausekkeen tuloksena voi olla mikä tahansa datamallin mahdollistama sekvenssi. Kielen nimi periytyy sen merkityksellisimmistä ominaisuudesta, polkulausekkeesta, joka tarjoaa mahdollisuuden osoittaa hierarkkisesti XML-puussa olevia XML-solmuja. World Wide Web Consortiumin (W3C) tuorein suositus on XPath 3.1, joka on hyvin yhteensopiva aiempien versioiden kanssa. [41]

### 5.1.3 XQuery

Kasvava määrä tietoa varastoidaan, välitetään ja esitetään XML-muodossa. Tällöin syntyy myös kasvava tarve tehdä kyselyitä näihin XML-tietolähteisiin. Yksi XML:n vahvuuksista on sen joustavuus esittää monimuotoisista tietolähteistä peräisin olevaa informaatiota. Jotta tätä XML:n joustavuutta päästään hyödyntämään, pitää XML-kyselykielessä olla ominaisuuksia informaation hakemiseen ja tulkitsemiseen näistä monimuotoisista tietolähteistä. [42]

XQuery on suunniteltu kieleksi, jossa kyselyt ovat tiiviitä ja ymmärrettäviä. Se on myös tarpeeksi joustava kyselemään erilaisista XML-tietolähteistä, mukaan luettuna sekä tietokannat, että dokumentit. [42]

XML:n käsittely on suuressa roolissa monissa internet-sovelluksissa. XML:ää käsitellään tavanomaisten ohjelmointikielien XML-ohjelmointirajapinnoilla. Välikerroksen web-sovelluksissa on paljon ohjelmakoodia tiedon yhteensovittamista varten eri järjestelmien ja tietolähteiden välillä. Järjestelmissä voivat olla erilaiset ohjelmointirajapinnat ja XML-kyselykielet. Natiivina XML-kielenä XQuery yksinkertaistaa XML:n käsittelyä. Väliohjelmistoissa, joissa alun perin ei-XML-muotoista dataa esitetään XML:nä, helpottaa se tiedon integrointia, koska se vapauttaa ohjelmoijan jokaisen tietolähteen yksityiskohdista. Samalla tavalla kuin SQL tekee kyselyjä tietokantaan ja palauttaa vastauksina taulukoita, XQuery tekee kyselyjä XML:ään ja palauttaa vastauksena XML:ää. Tällä hetkellä XQuery on tuettu useissa eri ohjelmointikielissä, integraatiokehitysympäristöissä ja -sovelluksissa sekä jopa tietokannoissa. [43]

XML:n rakenne on ohjannut XQueryn suunnittelua. XPath- ja XML-datamalli esittää XML:n rakenteen. Malli tukee niin tyypittämätöntä XML-dataa, että XML Scheman alkeistietotyyppejä. [43]

XML-rakenteiden etsimisessä XQuery käyttää XPath-lausekkeitä, joka tukee rakenteellisten riippuvuussuhteiden käyttöä ja vertailua solmujen välillä. XQuery-konstruktorit esittelevät XML-rakenteen muotosäännöt. XQuery siis luo elementit, attribuutit ja kommentit XML:n muotosääntöjen mukaisesti. Dynaaminen sisältö tuodaan XML:ään XQuery-lausekkeiden avulla. FLWOR (for, let, where, order by, return)-lausekkeiden avulla voidaan yhdistellä dataa useasta eri lähteestä ja tuottaa niiden pohjalta uusia tuloksia. XQuery sisältää valmiita funktioita, mutta käyttäjä voi myös luoda omia funktioita ja kirjastoja. [43]

W3C:n XML Query -työryhmä julkaisi monen vuoden kehityksen jälkeen suosituksen XQuery 1.0 vuonna 2007 [43]. Tuorein suositus XQuerysta on versio 3.1.

[42]

## 5.2 Protokollat

Tässä luvussa esitellään palvelukeskeisen arkkitehtuurin tyypillisesti hyödyntämää teknologiaa. Web Services -rajapinnat tarjoavat pääsyn hajautettuihin palveluihin. SOAP-protokolla vastaa sanomien välittämisestä eri palveluiden välillä ja siten sen avulla voidaan koostaa palveluita suuremmiksi kokonaisuuksiksi. WSDL-kielen avulla voidaan palvelu kuvata standardilla tavalla. UDDI on palvelurekisteri, josta palvelun käyttäjä voi hakea tarvitsemiaan palveluita.

### 5.2.1 Web Services (WS)

Perinteinen kolmikerrosarkkitehtuuri koostuu resurssienhallintakerroksesta (palvelin), sovelluslogiikkakerroksesta (väliohjelmisto) ja esityskerroksesta (asiakas). Kolmikerrosarkkitehtuuri kohtaa haasteita, kun integroitavien järjestelmien lukumäärä kasvaa - aina yritysrajojen ulkopuolellekin. Vaikka sovellukset ovat hajautettuja, niin väliohjelmisto säilyy keskitettynä. Tällöin on haastavaa soveltaa standardoituja väliohjelmistoprotokollia läpi koko yrityksen toimitusketjun. [13] Alonso ym. mainitsee, että nämä kolmikerrosarkkitehtuurin rajoitukset johtivat panostuksiin Web Services -teknologiaan, kuin myös palvelukeskeisen paradigman yleistymiseen sovelluskehityksessä [44].

Web Services julkaisee tietojärjestelmän toiminnallisuuden ulospäin ja tuo sen saataville avoimen standardin rajapinnan kautta. Kommunikointi on toteutettu käyttäen standardeja protokollia, kuten Hypertext Transfer Protocol (HTTP) ja Simple Object Access Protocol (SOAP). Rajapinnat ovat määritelty XML:ään pohjautuvalla kuvauskielillä Web Service Description Language (WSDL). [33]

Web Services -palveluiden käyttö edistää uudelleenkäyttöä, koska sovelluskomponentit tarjotaan palveluina. Lisäksi palvelukeskeinen arkkitehtuuri mahdollistaa ohjelmistokomponenttien löyhän vuorovaikutuksen. Vastakkainen malli tälle on staattinen ohjelman kääntövaiheessa tehty sidonta. [13]

Web Services on palvelukeskeisen arkkitehtuurin palveluinfrastruktuuriin soveltuva teknologia [20]. Web Services -teknologia on mahdollistanut palvelukeskeisen arkkitehtuurin siirtämisen käytäntöön [45]. Hajautettujen järjestelmien luonne määrää, että palvelut on määriteltävä korkeammalla tasolla. Liiketoimintapalvelut, jotka välittävät liiketoimintadokumentteja XML-muodossa ottavat tämän toki huomioon, mutta samalla aiheuttaa uusia haasteita liiketoimintapalveluiden koostamisessa ja dokumenttien käsittelyssä. BPM tarjoaa kehitys- ja suoritussympäristön, jossa liiketoimintapalveluja yhdistetään suuremmiksi yritystason prosesseiksi. Business Process Execution Language (BPEL) on nimenomaan suunniteltu toimimaan Web Services -palveluiden kanssa ja tarjoamaan Web Services -palveluiden koordinoinnin ja integroinnin korkeamman tason liiketoimintapalveluiksi. [20]

Web Services -termin käytöllä voidaan viitata RPC-malliin, REST-malliin tai yleisesti mihin tahansa palvelun toimittamiseen verkon ylitse. Pitää huomata, että RPC:n ja REST:in erot eivät välttämättä liity teknologiaan, vaan kyse on arkki-

tehtuurisista eroavaisuuksista. Nykyään Web Services ei viittaa enää pelkästään SOAP:iin, kuten sen alkuaikoina, vaan ennemminkin joukkoon WS-alkuisia standardeja. Näitä usein nimitetään yhteisnimellä "WS-\*. Näillä WS-standardeilla on pyritty lisäämään yksinkertaisen SOAP:in ilmaisuvoimaa. [36]

### 5.2.2 Simple Object Access Protocol (SOAP)

1990-luvun lopulla syntyi ajatus käyttää WWW-teknologioita muuhunkin kuin web-sivustojen toimittamiseen HTTP-asiakasohjelmiston ja -palvelinohjelmiston välillä. WWW ja sen tekninen perusta sisältäen URI, HTTP ja HTML tulivat erittäin laajasti käytetyiksi tiedon jakelu- ja palvelualueiksi. [36]

Tämän ajatuksen pohjalta syntyi Simple Object Access Protocol (SOAP), joka hyödynsi XML:ää. SOAP oli paketoitu muoto etäkutsuun tarkoitetuille RPC-mekanismeille, joita käytettiin hajautettujen järjestelmien toteutuksessa ja näiden paketoinnissa XML- ja HTTP-teknologioilla. Aiemmin useimmat RPC-mekanismit olivat toteutettu omilla pakkaus- ja marshallointimuodoilla sekä omilla siirtoprotokollilla. WWW-teknologioiden uudelleenkäyttö RPC-mekanismeissa oli siis täysin perusteltua, jolla saatiin vähennettyä omisteisten teknologien määrää. Lisäksi, SOAP-protokollan tunnelointi HTTP:n läpi saavutti yrityksissä suurta hyötyä, kun yhteydet palomuurien läpi olivat usein oletusarvoisesti avattu. [36]

SOAP:in käyttö RPC:n web-toteutuksena saavutti paljon suosiota ja siitä tuli hetken aikaa synonyymi Web Serviceille. SOAP:in käyttö RPC-kutsujen välittäjänä ei tosiasiallisesti toteuttanut aitoa Web Services -mallia, joka perustui webin arkkitehtuurimalliin. SOAP:in avulla yritettiin toteuttaa todellisia Web Servicejä sen sijaan, että toteutettiin pelkkää RPC:tä webin yli. Tämä johti SOAP:in ja muiden siihen liittyvien teknologioiden standardointiin 2000-luvun alussa. [36]

Internetin luontaisesta hajautetusta ja heterogeenisyydestä johtuen, viestintämekanismien on oltava alustariippumattomia, kansainvälisiä, turvallisia ja mahdollisimman kevytrakenteisia. XML:stä on tullut informaation ja datan koodauksen yleiskieli, joka on samalla myös alustariippumaton ja kansainvälinen. Viestintäprotokollien rakentaminen XML:n varaan on ollut luonnollinen vaihtoehto Web Serviceille. [46]

SOAP on alunperin Microsoftin luoma protokolla, jota sen jälkeen kehitettiin laajemmassa yhteistyössä. SOAP on XML-pohjainen protokolla sanomaliikenteelle ja proseduurien etäkutsuille (RPC). SOAP toimii olemassa olevien siirtoprotokollien päällä, kuten HTTP, SMTP tai MQSeries. [46]

SOAP on kevytrakenteinen protokolla, joka on tarkoitettu rakenteellisen tiedon välittämiseen hajautetuissa järjestelmissä. SOAP määrittelee avorakenteisen sanomaliikennemallin, jossa se hyödyntää XML-teknologioita. SOAP tarjoaa sanomarakenteen, jota voidaan välittää vaihtelevien protokollien päällä. Sanomaliikennemalli on suunniteltu täysin riippumattomaksi mistään ohjelmointimallista tai muusta toteutukseen liittyvästä semantiikasta. [47]

Suunnittelun pohjana SOAP:ssa on ollut yksinkertaisuus ja avorakenteisuus. SOAP pyrkii tavoittelemaan tätä jättämällä sanomavälitysmallista pois ominaisuuksia, jotka ovat aiemmin olleet mukana hajautetuissa järjestelmissä. Monet näistä ominaisuuksista on jätetty SOAP:n laajennuksiksi ja tehty erillisinä standardeina.



[48]

SOAP on pohjimmiltaan tilaton ja yksisuuntainen sanomanvälitysparadigma, mutta sovellukset voivat luoda monimutkaisempia vuorovaikutusmalleja. SOAP ei välitä sanoman sovelluskohtaisesta semantiikasta, kuten ei myöskään sanomien reitityksestä, datan siirron luotettavuudesta, palomuurien läpäisystä ja niin edelleen. SOAP tarjoaa puitteet, jolla sovelluskohtainen informaatio voidaan välittää avoimella tavalla. [48]

SOAP-sanoma koostuu SOAP-kuoren sisällä olevasta kahdesta elementistä, otsikosta ja rungosta. Otsikko ei ole sanomassa pakollinen. Otsikon tarkoituksena on voida välittää tietoa SOAP-sanomassa, joka ei liity sanoman varsinaiseen hyötykuormaan. Tämä mahdollistaa SOAP-sanomien laajentamisen sovelluskohtaisilla tavoilla. SOAP-otsikon suunnittelussa on pyritty ennakoimaan SOAP:in monia erilaisia käyttötapoja. Monet näistä liittyy lähetys- ja vastaanottosolmujen välillä oleviin välittäjäsolmuihin. Välittäjäsolmut voivat tarjota lisäarvoa tuottavia palveluita. SOAP-solmu voi tutkia, lisätä, poistaa tai välittää otsikkotietoja. SOAP-määritelmä ei käsittele otsikkoelementtien tietosisältöä tai miten SOAP-sanomat reititetään solmujen välillä. SOAP-runko on pakollinen elementti kuoren sisällä. Runkoelementin sisällä välitetään SOAP-sanoman ensisijainen päästä päähän siirrettävä informaatio. [48]

SOAP-sanoma on pohjimmiltaan yksisuuntainen siirto SOAP-solmujen välillä. Sovellukset voivat kuitenkin yksisuuntaisia siirtoja yhdistellä, jolloin on mahdollista toteuttaa keskustelunomaisia pyyntö-vastausmallin mukaista sanomanvaihtoa. SOAP:n alkuvaiheessa korostettiin tällä tavalla toteutettuja sovellusten etäkutsuja (RPC). On tärkeää huomata, että kaikkea SOAP:n pyyntö-vastausmallin välitystä ei voi, eikä tarvitse mallintaa etäkutsuina. Sellaisia pyyntö-vastaus malliin perustuvia käyttötapauksia on paljon enemmän, joissa tarkoituksena on vain yksinkertaisesti siirtää XML-pohjaista sisältöä SOAP-sanomissa ja muodostaa edestakainen keskusteluyhteys. [48]

### 5.2.3 Web Service Description Language (WSDL)

SOAP tarjoaa protokollana mahdollisuuden viestimiseen, mutta se ei kerro millaisia sanomia pitäisi vaihtaa, jotta palvelun kanssa voisi olla onnistuneesti vuorovaikutuksessa. Tätä varten on olemassa Web Service Description Language (WSDL), joka on IBM:n ja Microsoftin luoma XML:ään perustuva kieli. Toisin sanoen, WSDL-dokumentti kuvaa Web Services -rajapinnan ja tarjoaa palvelun kuluttajalle kontaktipisteen. [46]

Täydellinen WSDL-kuvaus tarjoaa kahdenlaista informaatiota: sovellustason palvelukuvauksen eli abstraktin rajapinnan sekä täsmälliset protokollakohtaiset yksityiskohdat, joita palvelun käyttäjältä vaaditaan. Tämä rajapinnan eriyttäminen kahdeksi kokonaisuudeksi selittyy sillä, että samanlainen sovellustason palvelu voi olla saatavilla useammassa eri päätepisteessä hieman erilaisilla pääsynhallintaprotokollan parametreilla. Palvelukuvauksen eriyttäminen kahdeksi erilaiseksi näkökulmaksi auttaa WSDL:ää kuvaamaan yleisen toiminnallisuuden näennäisesti erilaisten palvelun päätepisteiden välillä. [46]

WSDL:n abstraktissa palvelukuvauksessa on kolme pääkomponenttia: sanasto,

sanoma ja vuorovaikutus. Yksimielisyys käytetystä sanastosta on perusta kaikelle viestinnälle. WSDL käyttää ulkoista tyyppijärjestelmää tietotyyppien määrittelyyn. WSDL tukee useita järjestelmiä, mutta useimmat palvelut käyttävät XML Scheman mukaisia tietotyyppisiä. WSDL:n sidontaelementit tarjoavat tiedon mitä viestintäprotokollaa käytetään ja miten tällä kyseisellä protokollalla tehdään interaktio palvelussa. [46]

#### 5.2.4 Universal Description Discovery and Integration (UDDI)

Universal Description Discovery and Integration (UDDI) tarjoaa käyttäjille yhtenäisen ja systemaattisen tavan palveluntarjoajien löytämiseksi keskitetystä palvelurekisteristä [46]. UDDI:n määrittelyä on johtanut OASIS, joka ratifioi vuonna 2005 UDDI Version 3 OASIS-standardiksi [49].

UDDI tarjoaa kaksi perusmäärittystä, jotka määrittelevät palvelurekisterin rakenteen ja toiminnan. Toinen määrittelee informaation, mikä jokaisesta palvelusta on tarjottava ja miten sitä koodataan. Toinen taas määrittelee ohjelmointirajapinnan rekisterille, jonka kautta tietoa rekisterissä voidaan kysellä tai päivittää. Pääsy rekisteriin tapahtuu standardin SOAP-rajapinnan kautta. [46]

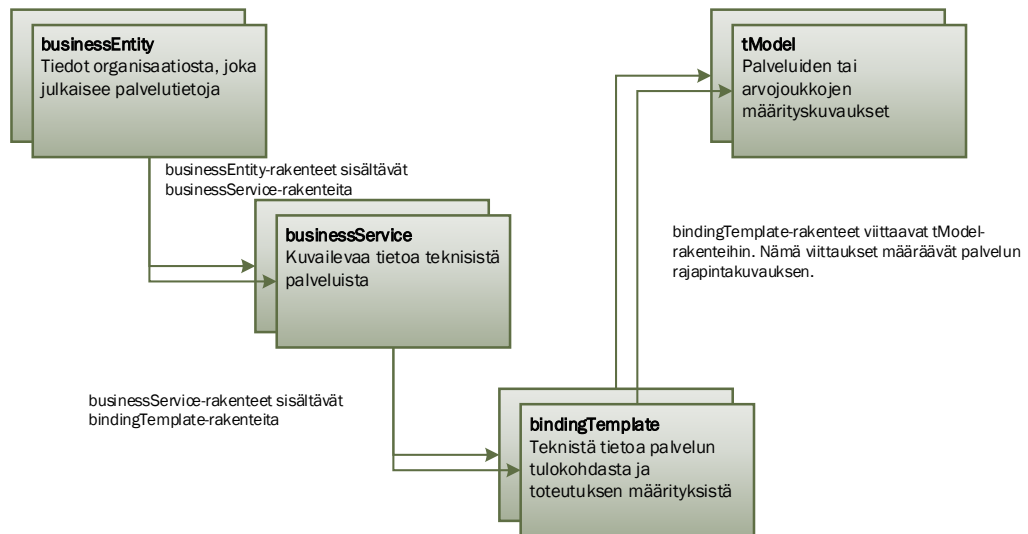
UDDI koodaa kolmen tyyppistä tietoa Web Serviceistä. "Valkoisten sivujen" informaatio sisältää tiedot nimistä ja yhteistiedoista, "keltaisten sivujen" informaatio tarjoaa kategorisoinnin liiketoiminta- ja palvelutyypeittäin ja "vihreiden sivujen" informaatio sisältää teknistä tietoa palveluista. [46]

UDDI:n tietomalli muodostuu pysyvien tietorakenteiden instansseista, joita kutsutaan entiteeteiksi. Entiteetit ovat XML-muotoisia, jotka ovat pysyvästi tallennettu UDDI-solmuihin. UDDI:n tietomalli on kokoelma ennalta määrättyjen entiteettityyppien instansseja. Entiteettityypit kuvaavat tietoja palvelutuottajaorganisaatiosta, tarjottavista palveluista sekä niiden teknisistä yksityiskohdista, jotka ovat välttämättömiä palveluiden käyttämisessä. UDDI:n tietomalli muodostuu neljästä pääentiteetistä, jotka muodostavat yhdessä muiden tietorakennetyyppien kanssa UDDI API Scheman. Neljä päätietorakennetyyppiä eli entiteettiä ja niiden suhteet toisiinsa on esitetty kuvassa 7. [49]

### 5.3 Yhteenveto Web Services -teknologioista

Hajautetut järjestelmät ja palvelukeskeinen arkkitehtuuri asettavat tiettyjä vaatimuksia teknologialle. Web Services -teknologiat mahdollistavat tällaisten järjestelmien rakentamisen. Web Services -teknologioiden taustalla on lukuisia XML-standardeja. XSD määrittelee XML-dokumenttien rakenteen ja antaa pohjan skeemapohjaisille sisältömuunnoksille, joita on tämänkin työn hajautetussa järjestelmässä hyödynnetty runsaasti. XQuery ja XPath tarjoavat hyviä työkaluja kyselyjen tekemiseen XML-dokumentteista ja siten myös toimivat olennaisena osana skeemamuunnoksissa.

Web Services-teknologialla tarjotaan palvelulle rajapinta, jonka käyttäminen verkon yli on mahdollista. WSDL on standardi kuvauskieli, jolla kuvataan Web Service -palvelun rajapinta ja sen käyttötapa. SOAP taas on yksinkertainen protokolla, jonka avulla voidaan toteuttaa sovellusten etäkutsuja tai XML-pohjaisen tiedon



Kuva 7: UDDI-tietorakenne.

siirtämistä järjestelmien välillä. Web Services -teknologiat ja XML-standardit luovat pohjan useille erilaisille järjestelmäintegraatiomalleille, joita on esitelty tarkemmin seuraavassa luvussa.

## 6 Järjestelmäintegraatiot

Sovelluskehityksen kasvavana tarpeena on ollut tiedon saatavuus ja siihen pääsy. Erilliset tietojärjestelmät eivät enää vastaa näihin tarpeisiin ja yrityksen sisäisiä ja yritysten välisiä integraatioita tarvitaan yhä enemmän. Yrityksissä on edelleen vanhoja perinnejärjestelmiä, mutta yhä enemmän uusia ratkaisuja, jotka perustuvat nykyaikaiseen arkkitehtuuriin. Tiedon saatavuuden nimissä kaikkien erilaisten järjestelmien on pystyttävä integroitumaan keskenään. Sovellusintegraatio voidaan nähdä haastavana tehtävänä, jopa suurimpana ongelmana, joka kohdataan yritysten tietojärjestelmien kehittämisessä.[50]

Tässä luvussa kuvataan yritysten tietojärjestelmien integraatioon merkittävästi vaikuttaneet suuntaukset: Enterprise Application Integration, väliohjelmistot ja palveluväylä (Enterprise Service Bus). Tässä työssä toteutetut integraatiot pohjautuvat vahvasti näihin integraatiomalleihin.

### 6.1 Enterprise Application Integration (EAI)

Uudet yritysten liiketoimintamallit, sisältäen modulaariset yhtiöt, ulkoistamisen ja yritysten verkottumisen, ovat tuoneet yritystoimintaan joustavuutta, nopeampaa reagointikykyä markkinatilanteen muutokseen, parantunutta lyhyen tähtäimen kilpailuetua ja riskittömämpiä innovaatioita. [51]

Tietojärjestelmien mukautumisen tähän muutokseen on mahdollistanut Enterprise Application Integration (EAI). EAI mahdollistaa monimuotoisten tietojärjestelmien kytkeytymisen toisiinsa ketterällä tavalla. EAI toimii tiedon, kommunikaation ja prosessien jakajana vähentäen samalla liiketoimintoja haittaavia informaatioosiloja. Uusien järjestelmien käyttöönotto yritys- tai yritysostotilanteessa ilman pitkiä ohjelmointipanoksia on selkeä etu. EAI-ratkaisut tarjoavat tavan yhdistää yhteistyötahojen ja kumppaneiden järjestelmät niin pitkään kuin on tarve ja irrottaa järjestelmät toisistaan kunnes liitto päättyy. Gartner määritteli EAI:n vuonna 2001 rajoittamattomaksi yrityksen tiedon ja liiketoimintaprosessien jakamiseksi kaikkien yhteen liitettyjen sovellusten ja tietolähteiden välillä. [51]

EAI:n hyödyt eivät rajoitu vain tiedon välittämiseen, vaan myös toteutustapaan. Aiemmin luotiin ohjelmakoodia, joka mahdollisti tietokoneen kommunikoimisen toisen kanssa. Jokainen lähdesovellus vaati kustomoidun yhteyden kohteeseen. EAI sitä vastoin käyttää väliohjelmistoa, johon sovellukset voivat kytkeytyä. EAI-palvelin toimii keskusjärjestelmänä, joka muuntaa tietoa sovelluksesta toiseen ja koordinoi prosesseja sovellusten välillä. Sovelluksia voi lisätä väliohjelmistoon tai poistaa sieltä ilman monimutkaista ohjelmointia. Standardinmukaiset Web Services -työkalut ovat mahdollistaneet tiedon kokoamisen yhteen internetin yli, jolloin EAI-projektien fokus on siirtynyt yritysten sisäisten järjestelmien integroinnista yhteistyöhön muiden tahojen kanssa. [51]

Tämän hetken tarpeena on jakaa informaatiota ja liiketoimintaprosesseja ilman laajoja muutoksia olemassa oleviin sovelluksiin ja tietorakenteisiin. Tämän tavoitteen saavuttamiseksi EAI on näyttänyt lupaavalta lähestymistavalta. EAI:n käyttöä organisaatioissa on edistänyt kolme tekijää. Ensimmäiseksi, yritykset ovat pyrkineet

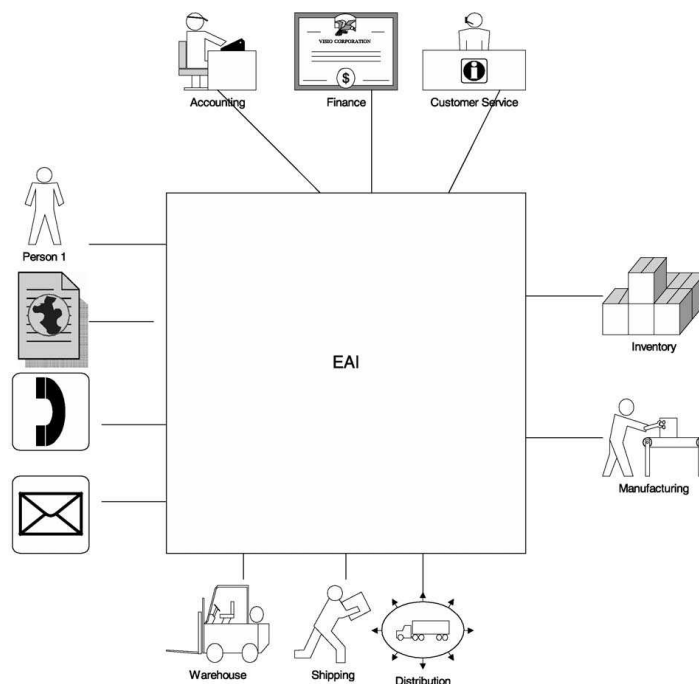
mukaan verkkokauppaan minimoimalla riskit ja maksimoimalla nykyisten tietojärjestelmäinvestointien hyödyntämisen. EAI tarjoaa tähän ratkaisun integroimalla useat liiketoimintaprosessit kumppaneiden ja asiakkaiden kanssa. Toiseksi, yritysfuusioissa ja yritysostoissa laajennetaan liiketoimintaa ja lasketaan kustannuksia konsolidoimalla toimintoja ja eliminoimalla päällekkäisyyksiä. Yrityssovellusten konsolidointi ei kuitenkaan yleensä onnistu matalin kustannuksin, sillä ne tyypillisesti ovat erilaisia sovelluksia eri alustoilla. EAI ratkaisee näitä haasteita. Kolmanneksi, paketoituissa toiminnanohjauksen järjestelmissä tieto integroituu tehokkaasti järjestelmän osien välillä, mutta toiminnanohjausjärjestelmien integrointi muiden paketoitujen järjestelmien kanssa on haastavaa, varsinkin jos sovelluksia ei ole kustomoitu integraatiota varten. [52]

EAI-arkkitehtuurin voi jakaa neljään palvelukerrokseen: siirtokerros, jakelukerros, muunnoskerros ja prosessinhallintakerros. Siirtokerros vastaa nimensä mukaisesti tiedon siirrosta järjestelmien välillä. Jakelukerros vastaa informaation jakelusta eli sanomien reitityksestä. Muunnoskerroksessa muunnetaan tieto vastaanottajalle soveltuvaan muotoon. Prosessinhallintakerroksessa sovelletaan dataan liiketoimintasääntöjä, jotka ovat osa liiketoimintaprosesseja. Liiketoimintasäännöt myös ohjaavat alemman kerroksen tietomuunnoksia. [52]

Erasala ym. [52] ja Venugopal ym. [53] tekevät vahvan eron EAI:n ja väliohjelmiston kanssa. Perinteinen lähestymistapa integraatioihin on ollut muuttaa järjestelmien ohjelmakoodia, jotta ne ovat voineet kommunikoida keskenään. Väliohjelmistoon liittyvä lähestymistapa on sallinut liitettävyyden vain järjestelmästä toiseen, jolloin se on ollut tietämätön muista samankaltaisista tapahtumista samassa IT-ympäristössä. [52] Väliohjelmisto on teknologia- ja sovellusriippuvainen, eikä tarjoa juurikaan näkyvyyttä liiketoimintaprosesseihin [53]. Uusien järjestelmien tuominen väliohjelmistointegraatioiden piiriin aiheuttaa monimutkaisen vyyhdin erilaisia linkkejä ilman keskitettyä hallintaa. EAI ei ole järjestelmästä toiseen tehtävää integraatiota, vaan enemmänkin monesta järjestelmästä moneen tehtävää integraatiota. EAI tarjoaa sekä tiedolle, että prosesseille yleisen tavan yli sovellusrajojen [53]. Tätä EAI:n mallia kutsutaan Hub-and-Spoke malliksi, jossa integraatiojärjestelmä toimii keskitettynä pisteenä. Käyttäjän tehtäväksi jää määrittellä integraatio EAI-ohjelmiston ja sovellusten välille. Kuvassa 8 on esimerkki Hub-and-Spoke-integraatiosta. Tilaus voi tulla verkkokaupan kautta puhelimitse tai sähköpostitse. EAI-järjestelmässä tietoa haetaan ja jaellaan eri sovelluksille. [52]

XML:n on huomattu sopivan hyvin myös EAI-ratkaisuihin, koska se tarjoaa yleisen välitysmuodon tiedolle ja on tekstipohjainen, jolloin useimmat alustat pystyvät sitä käsittelemään. XML on tietokantaneutraali, jolloin sen käyttäminen heterogeenisten tietokantojen liittämässä yhteen voi tarjota selkeää etua. XML:n tieto on rakenteista, jolloin kukin tietojärjestelmä voi poimia siitä vain tarvittavan tiedon. XML on sopiva muoto web-sovelluksissa, koska XML:n ja HTML:n yhdistäminen on helppoa. XML-dokumentti kuvaa myös eri XML-elementtien suhteet, jolloin ei välttämättä tarvita etukäteen tietoa sitä lähettävästä tietojärjestelmästä. XML mahdollistaa asiakasohjelmapäässä tehtävän manipulaation, jolloin tietoa ei tarvitse hakea joka kerta palvelinohjelmistosta erikseen. [52]

Standardien XML-protokollien käyttö on tehnyt Web Services -teknologiasta



Kuva 8: Hub-and-Spoke järjestelmäintegraatio.

alusta-, kieli-, ja toimittajariippumattoman, jolloin Web Services sopii hyvin myös EAI-ratkaisuihin. Web Services onkin toiminut katalysaattorina EAI:lle tarjoamalla luotettavan ja turvallisen infrastruktuurin. Vaikka EAI:n adaptereita ja konnektoreita tarvitaan yhä perinnejärjestelmien tietolähteiden kanssa, uusimpien järjestelmien sisäänrakennetut Web Services -ominaisuudet vähentävät kolmannen osapuolen adapterien ja konnektorien käyttöä. Perinteisiin EAI-teknologioihin verrattuna Web Services -teknologiat ovat yksinkertaisempia, edullisempia, standardipohjaisia, tehokkaampia ja dynaamisempia. [52]

## 6.2 Väliohjelmistot

Väliohjelmisto on uudelleenkäytettävä sovelluskerros, joka tarjoaa standardirajapintoja ja protokollia usein vastaan tuleviin ongelmiin, kuten heterogeenisuuteen, yhteentoimivuuteen, skaalautuvuuteen, vikasietoisuuteen, turvallisuuteen ja resursien jakamiseen. Väliohjelmisto pyrkii ensisijaisesti suojaamaan taustalla olevan hajautetun ympäristön monimutkaisuuden pitämällä sovellukset erillään protokollien käsittelystä, datareplikoinnista, verkkotason virheistä ja rinnakkaisuudesta. Lisäksi väliohjelmisto piilottaa järjestelmäarkkitehtuurin heterogeenisuuden, käyttöjärjestelmät, ohjelmointikielät ja verkkoprotokollat tehdäkseen sovelluksien kehittämisen ja hallinnan helpommaksi. Nämä kiinnostavat ominaisuudet ovat tuoneet väliohjelmiston vahvaksi työkaluksi järjestelmäkehityksen saralla. [54]

Väliohjelmiston eduksi voidaan laskea se, että se piilottaa taakseen lähde- ja

kohdejärjestelmien monimutkaisuuden. Sen tarjoamat liitettävyyden mahdollisuudet tekevät siitä ihanteellisen ratkaisun tiedon siirtämiseen järjestelmien välillä. Väliohjelmistoissakin on eroja ja niitä voidaan luokitella käyttötarkoituksiensa mukaan. Haluttuja ominaisuuksia voivat olla esimerkiksi turvallinen ja luotettava sanomanvälitys, tietomuunnokset, yleislähetys ja kommunikaatio. Väliohjelmisto voi esimerkiksi helposti täyttää lisääntyneen verkkokaupankäynnin tarpeet. Kuormanjakajilla voidaan väliohjelmiston kuormaa jakaa vaihtoehtoisille laitteille. [52]

Remote Procedure Call (RPC) on protokolla, jonka avulla komponentit voivat kutsua prosedureja etäjärjestelmissä ilman, että vuorovaikutus pitää ohjelmoida erikseen yksityiskohtaisella tasolla. RPC-väliohjelmisto laajentaa proseduurikutsun rajapinnan kaikille ohjelmistokehittäjille tunnetuksi tarjoamalla abstraktiomekanismit proseduurien kutsumiseen verkon kautta. RPC-väliohjelmistot ovat yleensä synkronisia ja siten eivät ole mahdollisia rinnakkaisuuteen ilman monisäikeisyyttä. Lisäksi niillä on tyypillisesti rajoitetut ominaisuudet virheidenkäsittelyssä. [54]

Message-Oriented Middleware (MOM) tukee tiedonvälitystä hajautettujen komponenttien välillä sanomanvälityksellä. Asiakassovellus lähettää pyynnön sisältävän sanoman parametreineen palvelulle verkon yli. Palvelinohjelmisto voi vastata vastaussanomalla, joka sisältää palvelun palauttaman tuloksen. [54]

MOM voidaan edelleen luokitella kahteen eri kategoriaan sanomanvalintamekanismin mukaan jonopohjaiseen väliohjelmistoon ja publish/subscribe-väliohjelmistoon. MOM tarjoaa toiminnallisuudet sanomien julkaisemiseen, tilaamiseen, toimittamiseen siten, että säilyvyys, replikointi, reaaliaikainen suorituskyky, skaalautuvuus ja turvallisuus ovat taattuja. MOM tukee asynkronista tiedonvälitystä luonnollisella tavalla, jonka seurauksena asiakasohjelmisto ja palvelinohjelmisto ovat voitu eriyttää toisistaan. [54]

Tapahtumakeskeisessä väliohjelmistossa (Transaction-Oriented Middleware, TOM) tapahtumat ovat sitoumuksia, jotka takaavat yhtenäiset tilasiirtymät järjestelmässä. Tätä käytetään pääasiassa arkkitehtuureissa, joiden komponentit ovat tietokantasovelluksia. TOM tukee tapahtumia sisältäviä komponentteja, joita suoritetaan hajautetulla alustalla. [54]

Hajautettu oliopohjainen väliohjelmisto tukee viestinvälitystä hajautettujen olioiden välillä. Asiakasolio voi kutsua palvelinoliota eri alustalla, kuten ne olisivat samassa muistiavaruudessa. Oliopohjainen väliohjelmisto tarjoaa samat oliopohjaisen ohjelmistotekniikan edut, kuten kapseloinnin, periytymisen ja monimuotoisuuden myös hajautetussa ympäristössä. [54]

Service-Oriented Middleware (SOM) eli palvelukeskeinen väliohjelmisto on väliohjelmistojen luokka, joka pohjautuu palvelukeskeiseen arkkitehtuuriin. Palvelun tuottaja, palvelun kuluttaja ja palvelurekisteri luovat SOA:n tunnusomaisen tyylin. SOM on tässä kontekstissa vastuussa palveluiden käyttöönotosta ja koordinoinnista. Tarkemmin sanottuna, SOM tarjoaa palvelun tuottajille perustavaa tukea palvelujen käyttöönotossa ja ilmoittamalla niiden olemassaolosta palvelurekisteriin. Palvelun kuluttajille SOM taas tarjoaa ominaisuudet palveluiden löytämiseen ja käyttämiseen. [54]

### 6.3 Palveluväylä

Taloudellinen globalisaatio ja tietoverkkojen kehittyminen ovat kasvattaneet yritysten tarvetta jakaa informaatiota ja sitä kautta lisännyt liiketoimintojen integraation tarvetta. Yritysten sovellusten integraatiot ovat kohdanneet uusia haasteita. Jotkin yritykset laajentavat liiketoimintaansa fuusioiden ja yritysostojen kautta. Toisaalta alati muuttuva markkinatilanne vaatii yritysten olevan joustavia ja mukautuvia ja siten mukautuvan uuteen markkinatilanteeseen samalla suojaten ja edistävän yrityksen kilpailuetuja. Vanhoja tietojärjestelmiä on pystyttävä hyödyntämään kattavasti ja historiallista informaatiota tarvitaan todentamaan kasvu. Nämä kaikki uudet haasteet vaativat entistä parempaa yrityksen sovellusten integraatiota. Sen ei pidä vain olla avoimempi, hajautetumpi ja löyhemmin rakennettu, vaan myös taata, että integroitava kohde säilyy riippumattomana. [55]

Yrityksen tietojärjestelmien lisääntynyt monimutkaisuus on suosinut palvelukeskeisen arkkitehtuurin periaatteiden omaksumista. Laaja joukko yrityksen verkossa olevia toiminnallisuuksia on paketoitu palveluiksi pisteissä, joissa toiminnallisuudet on toteutettu. Kun toiminnallisuudet ovat saatavilla palveluina, ovat ne uudelleen käytettävissä muissa yrityksen sovelluksissa. Palvelukeskeinen arkkitehtuuri suosii sovellusten rakentamista riippumattomista rakennuspalikoista eli palveluista, joita voidaan yhdistellä eri tavoin luotaessa uusia sovelluksia. Jotta voidaan taata maksimaalinen joustavuus palveluiden yhdistelyssä, pitää palveluiden olla tilattomia. Perättäisten palvelukutsujen pitää siis olla toisistaan täysin riippumattomia. Tällaisen arkkitehtuurin ytimessä on palveluväylä (ESB). Se on mahdollistamassa sovellusten toiminnan tekemällä palveluista kutsuttavia ja kommunikoidalla palveluiden kanssa paikasta riippumattomalla tavalla. Palveluväylä voi olla yhteydessä palveluihin, jotka käyttävät toisistaan poikkeavia protokollia ja datamuotoja käyttämällä tähän tarkoitettuja sovitusfunktioita. Palveluväylän toteutuksen ytimessä on väliohjelmisto, joka määrittelee palveluiden käyttämän tiedonvaihtoon tarkoitettun yleiskielen, joka ei riipu käytettävien palveluiden sijainneista tai niiden hyödyntämisen vaatimuksista. Väliohjelmisto voi käyttää lukuisia mekanismeja, kuten sanomajonoja, tuottaja-tilaaja-sanomanvälitystä tai Web Servicejä. Web Services tarjoaa hyvin määritellyn standardin, joka tekee sen ihanteelliseksi yhteentoimivuudessa. Kuitenkin hallinnoidussa ympäristössä, kuten yrityksessä, käytännöllisin ratkaisu on usein käyttää jotain omisteista sanomanvälityksen muotoa. Tämä johtuu usein suorituskykyeduista, joita yksinkertaisemmat protokollat tarjoavat sekä tarpeesta tukea perinteistä järjestelmäinfrastruktuuria. [56]

Väliohjelmisto yrityksessä on perinteisesti ollut raskastekoinen komponentti, koska pääasiassa sen roolina on ollut sanomiin sisältyvän tiedon siirto erittäin luotettavalla tavalla. Tämän tasoisen varmuuden hintana on protokollan yleisrasite ja levymuistin käyttö siirtämättömän tiedon tallennukseen. [56]

Yrityksissä siirryttiin manuaalisten integraatioiden kautta EAI:hin, mutta viimeisinä vuosina huomio on ollut palvelukeskeisessä arkkitehtuurissa. Jotta SOA:n edut voidaan tehokkaasti saavuttaa, yritykset ovat lähestyneet sitä palveluväylän (ESB) avulla. ESB on väliohjelmisto, joka kokoaa palvelukeskeisen arkkitehtuurin ja mahdollistaa kommunikoinnin web-pohjaisten yrityssovellusten välillä. Palveluväylä



tarjoaa liitettävyyden palvelun kuluttajan ja tarjoajan välille. Palveluväylän etuna aikaisempiin integraatoratkaisuihin on helppous, käytön joustavuus ja skaalautuvuus suurempiin ympäristöihin. [57]

Ensimmäiset SOA-toteutukset olivat konfiguroitu EAI:n Hub-and-Spoke-malliin perustuen. Järjestelmän älykkyys oli sijoitettu keskitetylle välittäjälle. Organisaatioiden kasvaessa sovellusten ja palveluiden määrä lisääntyy, joka saattaa olla liikaa yksittäiselle välittäjälle. Tämä johti ESB:n väylämäiseen arkkitehtuuriin. Palveluväylän toiminnassa on tiettyjä yhtäläisyyksiä PC:n perinteiseen laitteistoväylään, joka siirtää tietoa pitkin yhteistä väylää eri sovellusten välillä. Integraatiomootorit ovat jaeltu tämän arkkitehtuurin mukaan palvelimille, joilla pyörii palveluväyläohjelmisto. Se ei välttämättä ole EAI:n mukainen keskusvälittäjä. [57]

Palveluväylä hallitsee datamuodon muutokset, reitityksen sekä sanomiin liittyen niiden hyväksynnät, reititykset ja käsittelyn. Välityspalvelu käsittelee funktiot, kuten sovelluskuvaukset ja älykkään reitityksen. Palvelut mahdollistavat informaation siirron sovellusten välillä ottamalla vastaan tiedon lähettävän järjestelmän tarjoamassa muodossa ja muuntamalla se yhteensopivaksi vastaanottavaan sovellukseen. Tämä tapahtuu perusadapterien avulla, jotka toimivat standardeissa ympäristöissä, kuten Web Services tai Java Connector Architecture (JCA). Sanomanvälitysprotokollat mahdollistavat viestinnän palvelun kuluttajan ja tarjoajan välillä. Palveluväylä voi tukea satoja avoimia ja omisteisia protokollia, kuten Web Services, SMTP ja FTP. [57]

Palveluväylän konfiguroimat palvelut ovat uudelleen käytettäviä. SOA-järjestelmät voidaan integroida kerran, mutta tämän jälkeen integraatiopalvelut ovat käytössä aina kun niitä tarvitaan. Tämä auttaa liiketoimintaprosessien kehittäjiä, sillä palveluita ei tarvitse toteuttaa joka kerta kun integraatio otetaan käyttöön. EAI:n kanssa tilanne on ollut heikompi. Arkkitehtuurin hajauttaminen EAI:hin verrattuna auttaa uusien sovellusten liittämistä palveluun vaiheittaisesti. Palveluväylä myös mukautuu paremmin organisaatioiden toimintojen lisäykseen kuin EAI, koska sen sanomanvälityspalvelut ovat hajautettu ympäri verkkoa. [57]

## 6.4 Yhteenveto järjestelmäintegraatioista

Tässä luvussa esitettiin järjestelmäintegraatioihin liittyvät käsitteet EAI, väliohjelmistot ja palveluväylä. Perinteinen tapa tehdä järjestelmäintegraatioita oli tehdä muutokset suoraan keskusteleviin järjestelmiin. EAI oli ensimmäinen arkkitehtuurimalli, joka pyrki jakamaan yrityksen tietoa rajoittamattomasti yrityksen liiketoimintaprosessien ja sovellusten välillä. EAI:n kantavana ajatuksena oli tuoda integraatiot keskitetysti EAI-järjestelmään, jolloin integroitavien järjestelmien ei tarvinnut integroitua kuin EAI-järjestelmään, josta taas oli rajapinnat kaikkiin muihin järjestelmiin. Väliohjelmiston ajatus integraatiossa on ollut täysin EAI:n kaltainen, vaikka tutkimuksissa pyrittiinkin tekemään eroa EAI:n ja väliohjelmistoon perustuvien integraatioiden välille. On mahdollista, että ensimmäiset yritysten integraatioihin tarkoitettut väliohjelmistot olivat kehitysasteeltaan EAI-järjestelmiä perässä, mutta tätä väitettä pitäisi tutkia enemmän.

Palveluväylä (ESB) on tutkituista käsitteistä tuorein ja se luokitellaan sanoma-

tai palvelukeskeiseksi väliohjelmistoksi. Palveluväylä on ajan saatossa omaksunut EAI:n piirteet, mutta kehittynyt järjestelmänä edelleen toteuttamaan palvelukeskeisen arkkitehtuurin mallia, jolloin esimerkiksi palveluiden uudelleenkäyttö on esillä vahvemmin kuin EAI:ssa.

Web Services- ja XML-tekniologioiden yleistymisen nopeutti tietojärjestelmäintegraatioiden käyttöönottoa yrityksissä ja korvasi EAI:n sovellusadapterit standardeilla rajapinnoilla. Palveluväylässä lähtökohtana on ollut alusta saakka web- ja XML-tekniologia, joiden sopivuus palvelukeskeiseen arkkitehtuuriin on hyvä. Integraatioarkkitehtuurissa on siirrytty keskitetysti hallituista integraatioista yhä hajautetumpaan suuntaan. EAI:ssa integraatio oli keskitetty täysin EAI-ohjelmistoon ja kaikki muutokset integraatioissa piti huomioida EAI:ssa. Palveluväylämallissa integraatiotoiminnallisuutta voi yhä keskittää palveluväylälle, mutta trendinä on yhä enemmän viedä sitä palveluväylän käyttämiin palveluihin.

## 7 Karttapalvelusovelluksen vaatimusmäärittely

Tämän luvun tarkoituksena on kuvata yksityiskohtaisesti Digitan karttapalvelusovellukselle asetetut vaatimukset ja tuottaa vaatimusmäärittely. Tämä vaatimusmäärittely tarjoaa toiminnalliset vaatimukset karttapalvelusovellukselle, mutta myös asettaa vaatimuksia taustajärjestelmäratkaisuille ja sen rajapinnoille. Vaatimusmäärittely kuvaa myös olemassa olevien järjestelmien ja tehtyjen teknologiavalintojen asettamat rajoitukset.

### 7.1 Johdanto

Digitassa oli jo ennen tätä työtä tunnistettu tarve tarjota kuluttajille web-pohjaista karttasovellusta, jonka avulla esitettäisiin televisio- ja radiopalveluiden peittoalueita. Vaikka kuluttajat eivät olekaan suoraan Digitan asiakkaita, on Digita verkkooperaattorina luonnollisin taho tiedottamaan kootusti antennitelevisioverkon ja FM-radioverkon palveluiden kattavuudesta ja häiriöistä Suomessa. Digitalla on myös verkkotoimilupien myötä velvoite häiriötiedottamiseen [58]. Mediayhtiöt toisinaan tiedottavat omissa karttasovelluksissaan tai kotisivuillaan omien kanaviensa peittoalueita ja muuta verkkoon liittyvää ajankohtaista tietoa. Kuluttajan kannalta on kuitenkin yksinkertaisempaa nähdä peittoaluekartat ja muut palveluita koskevat tiedot samasta keskitetystä sovelluksesta kaikkien kanavien osalta.

Tarvemäärityksessä haluttiin selvittää kuluttajakäyttäjien lisäksi myös muut mahdolliset käyttäjäryhmät, joita Digitan karttapalvelu voisi palvella. Tarvemäärityksessä tunnistettiin käyttötapauksia kuluttajille, Digitan asiakkaille, Digitan antenniasentajista koostuvalle kumppaniverkostolle ja sisäiseen käyttöön.

Viestintäviraston määräys 66/2014 edellyttää, että teleyrityksen on tiedotettava käyttäjille vähintään sellaisista häiriöistä, jotka kestävät yhtäjaksoisesti yli 60 minuuttia ja vaikuttavat vähintään 250 käyttäjään. Nämä tiedottamiskynnyksen ylittävät häiriöt on tiedotettava internetissä ja puhelimitse. Internetissä tapahtuvan tiedottamisen on tapahduttava myös karttaesityksenä, kun vika tai häiriö koskettaa mm. maanpäällistä digitaalista joukkoviestintäverkkoa. Lisäksi karttaesityksen pitää olla viestintäpalvelun käyttäjämäärään, luonteeseen ja merkittävyyteen nähden tarkoituksenmukaista ja ymmärrettävää. [58]

Joukkoviestintäverkoissa 250 käyttäjän raja rikkoutuu häiriötilanteissa lähes poikkeuksetta, mutta yli 60 minuutin katkokset taas ovat verrattain harvinaisia. Viestintäviraston määräys on kuitenkin maanpäällisen digitaalisen televisioverkon osalta selkeä. Häiriötilanteiden esittäminen karttaesityksenä on yksi teleyrityksille määrätty velvoite, johon Digitan on sitouduttava televisio- ja radiopalveluiden osalta.

### 7.2 Yleiskuvaus

Digita halusi tuoda karttapalvelusovelluksen osaksi julkisia WWW-sivujaan, jolloin palvelu olisi helposti kuluttajien saatavissa. Sisäiseen käyttöön tarjottava karttapalveluun piti olla pääsy vain Digitan sisäverkosta ja mahdollisesti asiakkaille tarjottava

sovellus piti olla mahdollista upottaa heidän omille WWW-sivuille tai asiakkaan sisäverkkoon, esimerkiksi intranettiin. Vaikka karttapalvelusovellusta piti tarjota erinäköisenä eri käyttäjäryhmille, Digita piti tärkeänä, että ylläpidettävänä olisi vain yksi järjestelmä. Eri käyttäjäryhmien karttapalvelut olisivat vain erilaisia parametroituja käyttöliittymänäkymiä samasta sovelluksesta. Tällä nähtiin olevan suuri etu sovelluksen ylläpidossa ja jatkokehityksessä.

Uuden karttapalvelusovelluksen ei haluttu lisäävän ylläpitoon kuluva työmäärää, joten sovellus haluttiin toteuttaa täysin ilman pääkäyttäjille osoitettua hallintaliittymää. Hallintaliittymää voitiin harkita siinä tapauksessa vain, jos sitä kautta ylläpidettäisiin vain erittäin harvoin muuttuvaa tietoa tai järjestelmän parametreja. Hallintakäyttöliittymää ei nähty siis varsinaisena vaatimustekijänä, mutta sen toteuttamista voitiin harkita, mikäli projektin aikana siihen nähtäisiin tarvetta. Karttapalvelusovelluksen sisällön ylläpito haluttiin rakentaa pelkästään integraatioiden varaan. Sovelluksessa tarjottava sisältö haluttiin rajata sellaiseen sisältöön, jota oli jo valmiiksi Digitan tietojärjestelmissä. Mikäli sovelluksessa haluttaisiin nyt tai tulevaisuudessa esittää sellaista sisältöä tai paikkatietoa, jota Digitan tietojärjestelmissä ei ole tarjolla, nähtiin ensisijaisena vaihtoehtona muutokset Digitan taustajärjestelmissä ja niiden rajapinnoissa, kuin erillisen hallintakäyttöliittymän rakentamisen näiden tietojen ylläpitämiseen. Manuaalisen työn vähentämisen lisäksi, integraatio taustajärjestelmiin mahdollistaa huomattavasti reaaliaikaisemman tietosisällön esittämisen sovelluksessa. Tässä tietenkin oletuksena on se, että tieto Digitan taustajärjestelmissä on saatavilla ja ajantasaista. Taustajärjestelmissä oleva tieto kuitenkin palvelee myös muita käyttäjiä, järjestelmiä ja liiketoimintaprosesseja, jolloin jo olemassa olevat toimintatavat varmistavat tiedon ajantasaisuuden. Lisäksi on huomattu, että käyttäjien motivaatio tiedon ylläpitoon on silloin suurempi, kun samaa tietoa hyödynnetään monessa eri liiketoimintaprosessissa.

Sovelluksen käyttöliittymän haluttiin olevan WWW-pohjainen, jonka voisi upottaa käytännössä osaksi mitä tahansa internet-sivustoa. Sovelluksen piti toimia yleisimmissä selaimissa ilman tarvetta asentaa ylimääräisiä selainlaajennuksia tai muita ohjelmistoja. Sovellus ei saanut olla Java-sovelma tai se ei saanut sisältää Java-sovelmia, joiden kanssa oli havaittu aiemmin yhteensopivuusongelmia, johtuen erilaisista käyttöjärjestelmä-, selain- ja Java-ajoympäristökombinaatioista.

Digita ei itse halunnut tarjota palvelinympäristöä karttapalvelusovellukselle, vaan halusi sijoittaa sovelluksen ulkopuolisen palveluntarjoajan palvelimille. Mahdollisina vaihtoehtoina nähtiin koko ohjelmiston hankkimista palveluna (SaaS), sovelluksen asentamista ulkoiselle palvelualustalle (PaaS) tai pelkästään sijoittamalla sovellus ja sen alustapalvelut ulkopuoliseen palvelinsaliin (IaaS). Tällä valinnalla haluttiin mahdollistaa suorituskyvyn skaalautuminen karttapalvelusovelluksen käytön lisääntyessä. Järjestelmän suorituskyvyn piti vastata myös haasteeseen, jossa samanaikaisten käyttäjien määrä hetkessä satakertaistuu. Tällaisia piikki-ilanteita järjestelmän kuormitukseen voivat aiheuttaa esimerkiksi häiriöt televisio- ja radioverkoissa tai sovelluksen saama julkisuus valtakunnallisessa mediassa.

## 7.3 Tiedot ja tietokanta

Karttapalvelussa esitettävät tiedot tuodaan järjestelmäintegraation kautta Digitan taustajärjestelmistä ja muista lähteistä. Taustajärjestelmissä olevat tiedot sijaitsivat pääasissa relaatiotietokannoissa. Poikkeuksen teki peittoalueinformaatio, joka oli MapInfo Professional kartta- ja analyysisovelluksella tuotettua tiedostomuotoista paikkatietoa [59]. Lisäksi haluttiin tuoda kartalle tietoa teleoperaattoreiden 800 MHz:n tukiasemien sijainneista, jonne saatiin ajantasaista tietoa sähköpostitse Microsoft Exchange -sähköpostipalvelimella olevaan postilaatikkoon. Karttapalvelun taustakarttoja ja hakutoimintojen vaatimia tietoja Digitalla ei ollut, joten vaatimukseksi asetettiin näiden tietojen tuominen karttapalvelusovellukseen jostain ulkoisesta tietolähteestä.

Karttapalvelusovellukseen tulevien HTTP-pyyntöjen ei haluttu kuormittavan taustajärjestelmien tietokantoja tai muita alustoja. Tällöin on riskinä, että karttapalveluun kohdistuva suuri kuormitus kuormittaisi myös Digitan liiketoimintakriittisiä järjestelmiä. Karttapalvelusovellukseen kohdistuvan kuormituksen ei haluttu vaikuttavan taustajärjestelmiin millään tavalla.

Karttapalvelun tietokannat haluttiin jakaa selvästi kolmeksi eri kokonaisuudeksi. Ensimmäiseksi, Digitan taustajärjestelmien tietokannat pysyivät Digitan omassa verkossa eikä niihin haluttu kohdistuvan kuormitusta juurikaan nykyistä enemmän. Toiseksi, taustajärjestelmien heterogeenisista tietokannoista haluttiin tietojen replikoituvan yhteiseen tietokantaan, johon kohdistuisi myös kaikki karttapalveluaineiston tietopyynnöt. Kolmanneksi, karttapalvelusovelluksen oma tietokanta haluttiin sijoittaa palveluntarjoajan palvelimelle, jolloin tietyt tietoturvaan ja suorituskykyyn liittyvät riskit saatiin ulkoistettua. Digitalla ei myöskään ollut valmiina riittävän skaalautuvaa ja turvallista ajoympäristöä karttapalvelusovellukselle, eikä kokemusta vastaavanlaisista kuluttajille tarjottavista palveluista.

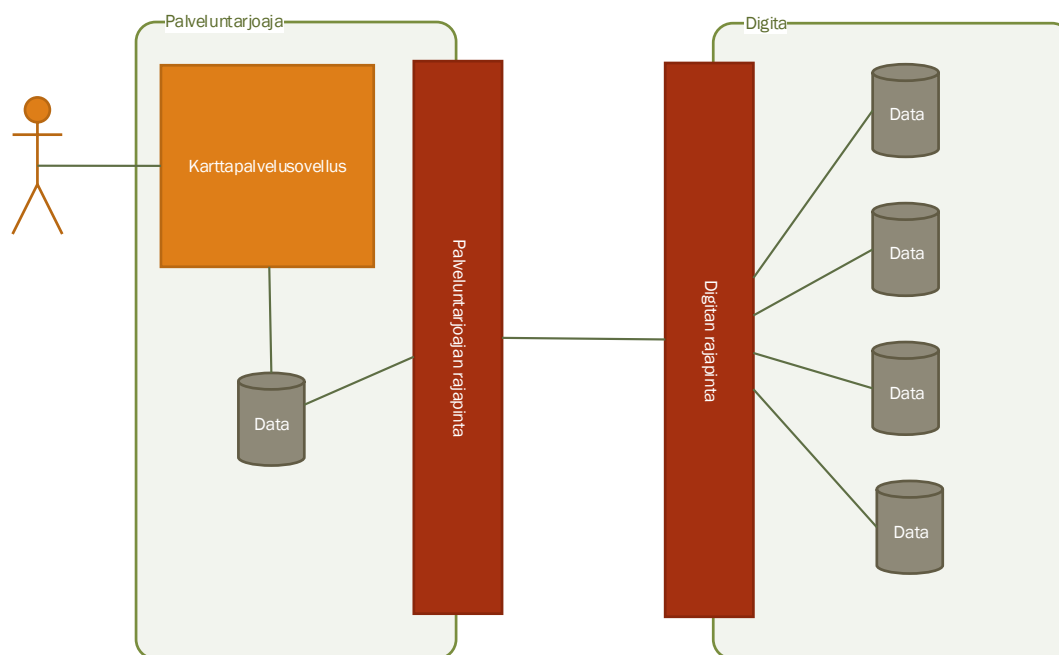
## 7.4 Toiminnot

Karttapalvelusovelluksen vaatimusmäärittelyä lähdettiin tuottamaan käyttötapausten avulla. Vaatimusmäärittelyn yhtenä tavoitteena oli saada aikaiseksi mahdollisimman tarkat käyttötapaukset koskien kuluttajia, asiakkaita, Digitan sisäisiä käyttäjiä ja muita sidosryhmiä. Käyttötapauksia tarkasteltiin erikseen FM-radiopalveluiden ja televisiopalveluiden näkökulmasta. Kuvissa 10 ja 11 on esitetty sovelluksen käyttötapaukset UML-käyttötapauskavioina.

### 7.4.1 Sijainnissa vastaanotettavissa olevat palvelut

Karttapalvelun suunnittelun lähtökohtana oli toteuttaa heille suunnattu palvelu, joka tarjoaisi käyttäjille tiedon, mitä televisio- ja radiopalveluita tietyssä sijainnissa olisi saatavilla. Sijainti tarkoittaa sellaista pistettä Suomessa, jolle voidaan osoittaa maantieteelliset koordinaatit.

Tarvemäärittelyssä havaittiin, että käyttäjällä on tarve määrittää sijaintinsa kolmella eri tavalla. Useimmiten käyttäjä haluaa etsiä sovelluksesta esimerkiksi kotinsa

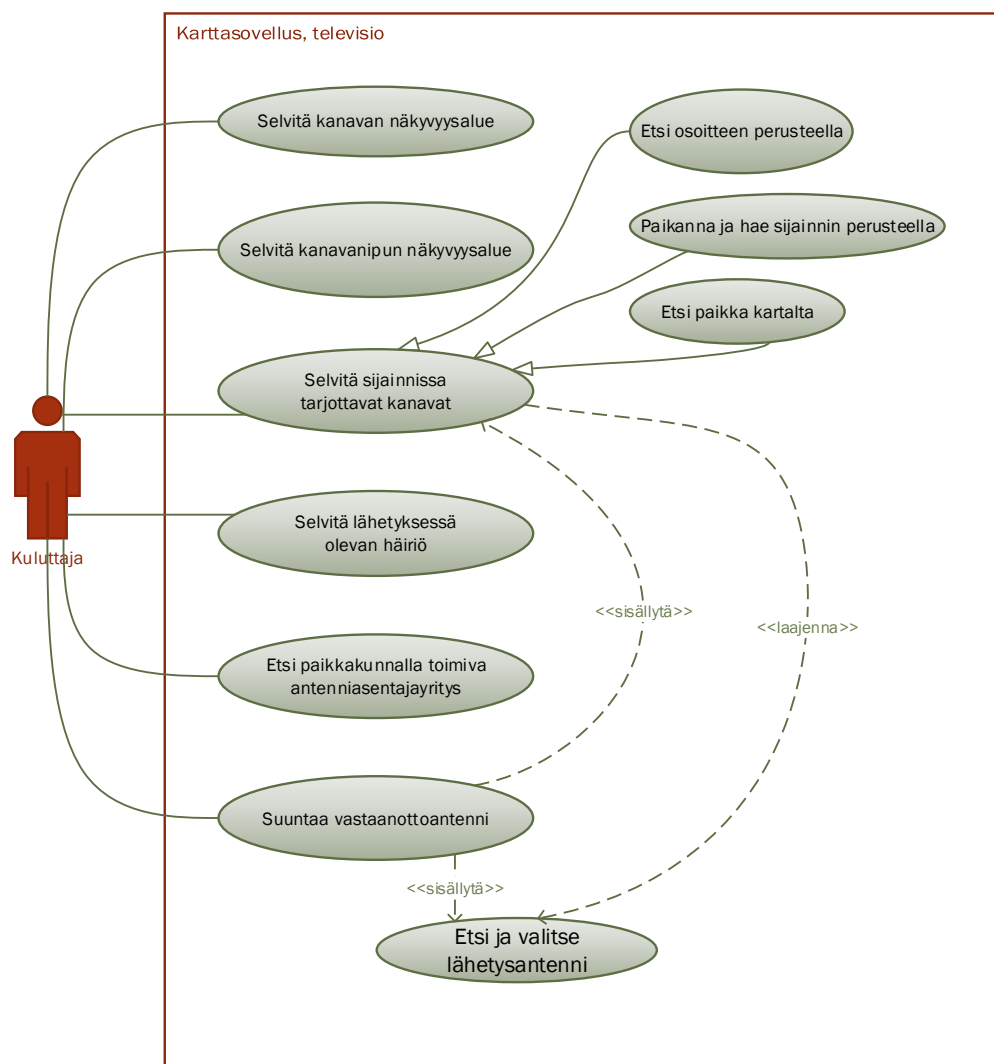


Kuva 9: Vaatimusmäärittelyssä esitetty karkea arkkitehtuurimalli.

tai kesämökkkinsä sijainnin, jotka useimmiten on haettavissa postiosoitteen perusteella. Tällöin karttapalveluun haluttiin toteuttaa osoitehaku, jolla on mahdollista hakea kaikki osoitteet Suomessa. Toiseksi, käyttäjä on useimmiten jo siinä samaisessa sijainnissa, johon hän haluaa myös karttapalvelussa navigoida. Tämän vuoksi yhdeksi vaatimukseksi asetettiin paikannustoiminto, jossa käyttäjän omaa sijaintia voitiin hyödyntää sovelluksessa. Kolmanneksi, hakutoiminto voi palauttaa osoitehauulla sijainnin, joka ei täsmälleen vastaa haluttua sijaintia tai käyttäjä etsii sellaisella paikannimellä, jota hakutoiminto ei tunnista. Tällöin haluttiin antaa käyttäjälle vielä mahdollisuus asettaa tarkasteltava sijainti kartalle manuaalisesti. Tämän vuoksi karttapohjasta haluttiin selkeyden lisäksi myös sellainen, joka sisälsi runsaasti paikannimiä ja kadut olivat nimettyjä.

Hakutavasta riippumatta tulos esitetään karttapalvelussa samaan tyyliin. Haettu sijainti etsitään karttapohjalta, kartta keskitetään sijaintiin ja zoomataan siitä sopivalle etäisyydelle. Sopiva etäisyys määritettiin siten, että ainakin yksi lähetysasema pitää mahtua hakutuloksen näkymään yhdessä haetun sijainnin kanssa. Ei kuitenkaan haluttu, että zoomausetäisyys määritetään dynaamisesti, vaan valitaan sopiva staattinen zoomausetäisyys, joka lähes poikkeuksetta tuo halutuloksen sopivalle zoomausetäisyydelle.

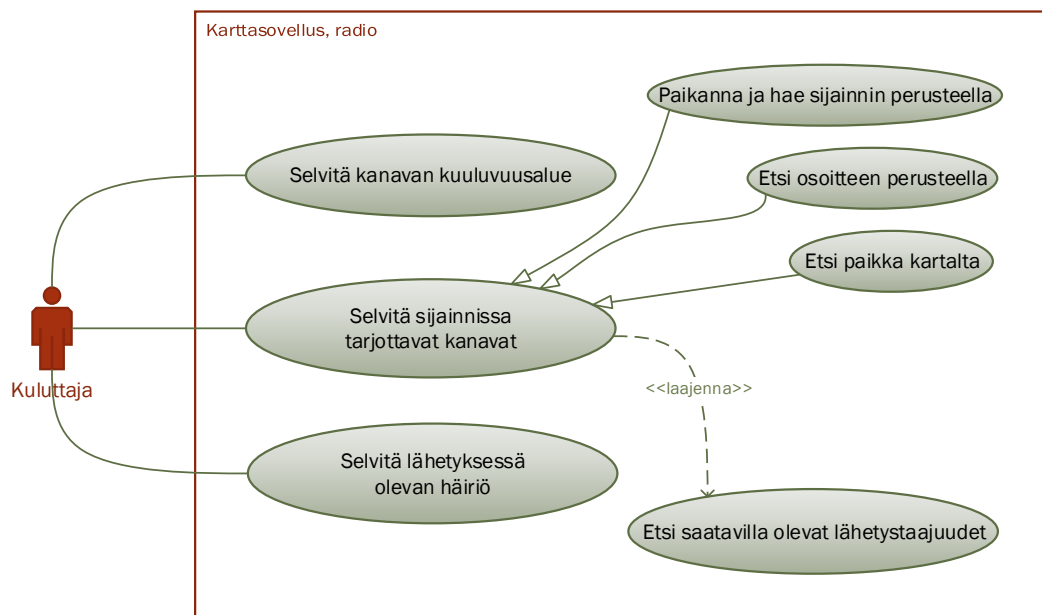
Saatavilla olevat televisiopalvelut vaihtelevat Suomessa paikkakunnittain. Toki vaihtelua aiheutuu jo siitäkin, kuuluuko talous kaapeli- ja antennitelevisioverkkoon. Antenniverkossa tarjottavat televisiokanavatkin vaihtelevat paikkakunnittain. Kaikilta lähetysasemilta ei lähetetä kaikki kanavanippuja ja joidenkin kanavanippujen



Kuva 10: Televisiokarttapalvelun käyttötapaukset.

sisältökin voi vaihdella lähetyksittäin. Suomessa on sijainteja, joissa on vastaanotettavissa yhden, useamman tai ei minkään lähetyksen televisiosignaalia. Karttasovelluksen tavoitteeksi asetettiin näiden kolmen eri tilanteen esittäminen selkeällä tavalla siten, että kuluttaja tyypillisesti osaisi valita itselleen parhaimman vastaanotettavan signaalin. Kuluttajasta riippuu arvostaako hän häiriötöntä ja vahvaa signaalia, mahdollisimman suurta televisiokanavien lukumäärää tai jonkin tietyn televisiokanavan vastaanottamista.

FM-radiotoiminnassa paikallisuus näkyy televisiotoimintaakin vahvempana. Ympäri Suomea levittyneet kaupalliset paikallisradiot kuuluvat tietyillä paikkakunnilla. Valtakunnallinen yleisradiotoiminta taas puolestaan kattaa lähes koko maan. Näiden lisäksi on olemassa kaupallisia radioasemia, joiden kuuluvuus on paikkakunta- tai maakuntatasoa suurempi, mutta ne eivät kuitenkaan pääse väestöpeitossa Yleisradion



Kuva 11: Radiokarttapalvelun käyttötapaukset.

palveluiden tasolle. Karkeasti voidaankin sanoa, että Suomessa ei ole kahta kuntaa, joissa radiopalveluiden tarjonta olisi identtinen. Toisaalta myös kuntien sisällä radiotarjonta vaihtelee. Aikaisemmin Digitassa on radiokanavien kuuluvuuksia listattu kunnittain, mutta on kovin subjektiivista vetää raja siihen, milloin radiosignaali on koko kunnan alueella niin hyvä, että voidaan julkisesti väittää radiokanavan kuuluvan kyseisessä kunnassa. Näihin haasteisiin karttasovellus tarjoaa erinomaisen työkalun. Sen avulla voidaan tarjota tietoa radiokanavien kuuluvuudesta jokaisessa osoitteessa, tai täsmällisemmin ilmaistuna, jokaisessa pisteessä Suomessa. Tässä tapauksessa karttapalvelu tarjoaa kuluttajalle vastauksen siitä, mitä radiokanavia hänen pitäisi pystyä vastaanottamaan ja millä lähetystaajuudella.

Valtakunnallisten radiokanavien lähetyverkko peittää Suomen jokseenkin kokonaisvaltaisesti. Suuressa osassa maata on mahdollista vastaanottaa samaa radiokanavaa kahdelta tai useammalta eri lähetyksasemalta. Kuluttajalle tämä tilanne näkyy siten, että jokin tietty radioasema on vastaanotettavissa kahdella eri taajuudella. Haluttiin, että karttapalvelu ei vain kerro tietyssä sijainnissa tarjottavaa radiokanavaa, vaan sen kaikki lähetystaajuudet, joiden vastaanottaminen on mahdollista. Lisäksi haluttiin, että karttapalvelu pystyy luettelemaan lähetystaajuudet siten, että kuluttaja saa tiedon mikä on kulloinkin vahvin vastaanotettavissa oleva radiosignaali.

#### 7.4.2 Suuntaa vastaanottoantenni

Televisiokanavien vastaanotto tapahtuu suunta-antennilla, joka suunnataan kohti signaalia tarjoavaa lähetyksasemaa. Radiovastaanotossa antennin suuntausta ei välttämättä tarvita. Televisioantennin suuntaus kotitalouksissa ja mökeillä on ollut



kohtuullisen haastavaa, koska useimmiten näköyhteys lähetysantenniin on puuttunut. Karttapalveluun haluttiin tarjota kuluttajille palvelu, joka tarjoaa tiedon parhaasta ja toiseksi parhaasta lähetysasemasta kussakin sijainnissa ja kompassisuunnan näille asemille. Lisäksi haluttiin antaa kuluttajalle mahdollisuus valita sovelluksesta muitakin asemia, mikäli sovelluksen ehdottamat kaksi parasta lähetysasemaa eivät olleet hänelle riittäviä. Tällainen tilanne voi tulla vastaan esimerkiksi siinä tilanteessa, jos näiltä kahdelta asemalta ei lähetetä sellaista kanavanippua tai televisiokanavaa, jota kuluttaja haluaa ehdottomasti vastaanottaa.

### 7.4.3 Etsi paikkakunnalla toimiva antenniasentajayritys

Lähes kaikilla Suomen paikkakunnilla toimii Digitan kouluttamia ammattitaitoisia DigitaPRO Antenniasentajia. He tekevät erilaisia antennien ja digisovittimien asennus-, kunnossapito- ja opastustöitä. Tiedettiin, että erilaisissa ongelmatilanteissa kuluttajat hakeutuvat Digitan WWW-sivuille ja karttapalveluun. Haluttiin, että samassa tilanteessa missä kuluttaja on etsinyt oman osoitteensa tai sijaintinsa kartalta, hänelle lueteltaisiin tarjottavien palveluiden lisäksi myös hänen paikkakunnalla toimivat DigitaPRO Antenniasentajat. Sovelluksessa haluttiin tarjota antenniasentajien ajantasaiset yhteystiedot, jotta kuluttaja pystyy ongelmatilanteessa ottamaan heihin vaivatta yhteyttä.

### 7.4.4 Selvitä lähetyksessä oleva häiriö

Karttapalvelusovellus nähtiin hyvänä välineenä häiriötiedotuksessa, jolla joukkoviestintäverkkojen häiriöiden vaikutusalueet pystyttiin kuvaamaan visuaalisesti. Toiseksi, Viestintäviraston määräys [58] myös velvoitti tiedottamaan verkossa olevista häiriöistä karttapohjalla. Kuluttajien kannalta tässä nähtiin myös aito käyttötapaus, jossa karttapalvelulla on merkittävä rooli omatoimisessa vianselvityksessä. Kuluttaja pyrkii selvittämään televisio- tai radiolähetyksessä olevaa häiriötä viimeistään siinä vaiheessa, kun häiriön kesto on ylittänyt hänen sietokykynsä. Häiriön selvittämistä varten kuluttaja tarvitsee tiedon onko häiriö mahdollisesti hänen vastaanottimessaan, antennijärjestelmässään, lähetysverkossa tai muussa järjestelmän osassa. Kuluttaja itse on vastuussa vastaanottimestaan ja mahdollisesti myös antennijärjestelmästä, jolloin hän haluaa sulkea pois vian mahdollisuuden lähetyksessä tai lähetysverkossa. Tällöin Digitan karttapalvelun pitää pystyä näyttämään mahdolliset verkossa tai lähetyksessä olevat häiriöt, niiden maantieteellisen vaikutusalueen, mihin palveluihin häiriö vaikuttaa ja häiriön alkamisajan. Näiden tietojen perusteella kuluttajan on mahdollista arvioida onko hänen kokema häiriö palvelussa lähetyksestä ja verkosta peräisin, vai onko häiriö sittenkin vastaanottopäässä. Tässä yhteydessä sovelluksen pitää pystyä tarjoamaan myös yhteystiedot antenniasentajayrityksistä, mikäli kuluttaja tarvitsee häiriön selvitykseen ammattilaisen tukea.

### 7.4.5 Palveluiden peittoalueet

Palvelun peittoalueella tarkoitetaan joko televisiokanavan näkyvyysaluetta tai radio-kanavan kuuluvuusaluetta Suomessa. Kuluttajat halusivat nähdä karttapalvelussa

myös tällaisen kokonaisnäkyvän kanavan maantieteellisestä peitosta. Karttapalveluun haluttiin lista kaikista Digitan välittämistä televisio- ja radiokanavista. Valitsemalla listasta yhden kanavan, haluttiin karttasovelluksen piirtävän alueen, jossa palvelu on saatavilla. Tällä tavalla kuluttajille tarjottiin mahdollisuus nähdä mitä palveluita Suomessa on saatavilla ja sitä kautta lisätä kysyntää palveluille tulevaisuudessa. Mediayhtiöt Digitan asiakkaina päättävät millä alueilla he haluavat palveluitaan tarjota ja tässä päätöksenteossa mediayhtiöt kuuntelevat myös kuluttajien tarpeita ja toiveita. Peittoalueen haluttiin olevan kuuluvuutta tai näkyvyyttä edustava monikulmio, joka on väritetty karttapohjasta erottuvalla tavalla ja on puoliläpinäkyvä. Puoliläpinäkyvyyden haluttiin olevan sellainen, joka tuo karttamerkinnät selvästi näkyville peittoaluekuvan läpi.

#### 7.4.6 Digitan kuluttajapalvelu

Digitan sisäisessä käytössä karttapalvelua määriteltiin ensisijaisesti kuluttajapalvelu Digita Infon tarpeiden mukaan. Digita Info palvelee kuluttajia arkisin puhelimen ja sähköpostin välityksellä. Kuluttajapalvelun ensisijaiset tehtävät on tarjota perustietoa Digitan palveluista, erityisesti antennitelevisiosta ja radiosta. Kuluttajapalvelun asiantuntemus ja tekninen tuki kattaa erityisesti kysymykset liittyen Digitan lähetyserkkoihin, antennijärjestelmiin, näkyvyys- ja kuuluvuusalueisiin, vastaanottoongelmiin, radiokeleihin ja eri päätelaitteiden yhteensopivuuteen Digitan verkkojen palveluiden kanssa.

Digitan sisäiset käyttötapaukset eivät poikenneet merkittävästi kuluttajien käyttötapauksista. Digita Infolla oli samalla tavalla tarve selvittää sijainnissa vastaanotettavat palvelut, avustaa vastaanottoantennin suuntaamisessa, informoida kuluttajaa paikkakunnalla toimivista antenniasentajista, nähdä verkossa olevat häiriöt ja tiedottaa niistä kuluttajaa, sekä luonnollisesti nähdä Digitan verkkojen palveluiden näkyvyys- ja kuuluvuusalueet.

Digita Info myöntää kuluttajille oikeuden hankkia satelliittipaketti yhteiskunnan tukemana, mikäli tv-vastaanotto muista toimenpiteistä huolimatta ei ole mahdollista. Vastaanottohäiriöiden selvittämiseksi Digita Infolle on tärkeää nähdä näiden myönnettyjen satelliittipakettien sijainnit ja sitä kautta nähdä antennitelevisioverkon katvealueet. Tämä tieto helpottaa kuluttajapalvelua häiriötilanteiden ja syiden analysoinnissa.

Sisäiseen karttapalveluun haluttiin toteuttaa lisätoimintoja, jotka laajensivat kuluttajille määriteltyjä käyttötapauksia. Sovelluksen hakutoimintoa haluttiin laajentaa kattamaan paikkakunta- ja osoitehaun lisäksi myös haku maantieteellisillä koordinaateilla tai Digitan asematunnuksella. Hakutuloksena oli sama etsityn paikan keskittäminen ja kartan zoomaaminen kuten kuluttajien karttapalvelussa. Lisäksi Digita Info halusi poimia karttasovelluksesta minkä tahansa sijainnin koordinaatit EUREF-FIN-koordinaatistossa, yhtenäiskoordinaatistossa (YKJ) tai WGS84-koordinaattijärjestelmässä. Koordinaattien piti olla mahdollista kopioida muihin karttasovelluksiin mahdollista jatkoanalyysia varten.

Sisäisessä käytössä karttapohjalle asetettiin erilaisia vaatimuksia kuin kuluttajien karttapalvelulle. Jos sisäisessä käytössä taustakartan selkeys ja rauhallinen väri-

tys olivat tärkeimpinä vaatimuksina, niin Digita Infon piti karttapohjasta nähdä asuinrakennuksien sijainnit, katujen nimet, korkeuskäyrät sekä maastomerkinnyt. Mäillä, rakennuksilla ja puustolla on myös vaikutus televisio- ja radiopalveluiden vastaanottoon.

Televisiokanavien näkyvyysalueet haluttiin näyttää sisäisessä karttapalvelussa ydinalueena sekä sen ympärillä olevana lievealueena. Ydinalueella signaalin vastaanotto onnistuu paikasta riippumatta 99 % todennäköisyydellä ja lievealueella todennäköisyys on 80 % [59]. Ydin- ja lievealueiden erottelua ei haluttu näyttää kuluttajien karttapalvelussa, koska kuluttajilla ei ole pääsääntöisesti tietoa eri alueiden eroista. Toisaalta myös lisäinformaatio karttapohjalla tekee sen tulkitsemisesta aina hankalampaa.

#### 7.4.7 Taajuustalkoiden neuvontapalvelu

Vuoden 2014 alusta 800 MHz:n taajuusalue on vapautunut mobiililaajakaistaa palvelulle Long Term Evolution-verkolle (LTE). Uusi mobiiliverkko voi aiheuttaa antenni-TV-talouksille häiriöitä televisiovastaanotossa, koska taajuusalue on heti televisio-toiminnassa käytetyn taajuusalueen vieressä. Suomalaiset matkapuhelinoperaattorit ovat perustaneet yhteishankkeen nimeltä Taajuustalkoot, joka pyrkii selvittämään ja ratkaisemaan mobiiliverkon aiheuttamia häiriöitä televisiovastaanotossa. Taajuustalkoita varten on perustettu neuvontapalvelu, joka selvittää aiheutuuko kuluttajien havaitsemat vastaanotto-ongelmat 800 Mhz:n LTE-verkosta. Neuvontapalvelu tilaa antenniurakoitsijan selvittämään ja ratkaisemaan ongelman paikanpäälle.

Neuvontapalvelun toiminnan yhtenä mahdollistajana tarvittiin karttapalvelutyökalu, joka näyttäisi kaikkien LTE-tukiasemien sijainnit mahdollisimman reaaliaikaisesti. Kuluttajan ottaessa yhteyttä neuvontapalveluun television vastaanotto-ongelmiin liittyen, voi neuvontapalvelun asiantuntija arvioida lähellä olevien tukiasemien vaikutuksen. Neuvontapalvelussa haluttiin tukiasemasta nähdä myös muuta tietoa kuin sijainnin. Karttapalvelun kautta piti ainakin nähdä tukiaseman omistavan teleoperaattorin nimi sekä tukiaseman käynnistysajankohta. Kuluttajan vastaanotto-ongelmien alkamisen ajoittuminen tukiaseman käynnistymisajankohtaan viestii usein ongelman liittyvän uuteen LTE-tukiasemaan.

#### 7.4.8 Asiakkaan tai yhteistyökumppanin oma karttapalvelu

Kokonaisten palveluiden peittoalueiden esittäminen karttapalvelussa vastasi myös Digitan asiakkaiden tarpeeseen esittää heidän omien kanaviensa peittoalueita yhtiön omilla kotisivuilla tai muualla verkossa. Sovellus haluttiin suunnitella siten, että tarvittaessa se voitaisiin rajata esittämään vain tietyn asiakkaan kanavaa tai kanavia ja upottaa se vaikka suoraan asiakkaan omille sivuille. Toisin sanoen karttapalvelusta haluttiin tehdä myös suoraan Digitan asiakkaille tarjottava palvelu.

Asiakkaiden ja muiden yhteistyökumppanien tarpeet haluttiin nähdä myös laajemmalla mittakaavassa. Digitalle toteutettu karttapalvelu tuskin sellaisenaan ratkaisee kaikkia asiakkaiden tarpeita, joten Digitan karttapalveluratkaisuun haluttiin to-

teuttaa aineistorajapinta, jonka kautta kaikkea karttapalvelussa esitettävää sisältöä voitaisiin jakaa sitä tarvitseville.

## 7.5 Ulkoiset liittymät

Karttapalvelusovellus ei ole irrallinen järjestelmä, vaan toimii välittömässä vuorovaikutuksessa Digitan olemassa olevien järjestelmien kanssa. Tämän vuoksi vaatimusmäärittelyssä otettiin kantaa sovelluksen laitteisto-, tietoliikenne- ja ohjelmistoliittymiin. Järjestelmän eri osat haluttiin kytkeä toisiinsa mahdollisimman löyhästi, jolloin suoria laitteistoliittymiä ei tarvittu, tietoliikenneliittymissä haluttiin nojata jo luonteeltaan löyhästi kytkeytyviin web-teknologioihin. Ohjelmistotason liittymissä ei otettu kantaa käytettäviin protokolliin, vaan pyrittiin ennemminkin tunnistamaan ohjelmistot, jotka piti kytkeä osaksi karttapalveluratkaisua. Käytännössä tietoliikenneliittymien rajaaminen HTTP-protokollaan ohjasi ohjelmistoliittymien valintaa vahvasti SOAP- ja RESTful-protokollien suuntaan.

### 7.5.1 Laitteistoliittymät

Erityisille laitteistoliittymille ei nähty tarvetta tulevassa ratkaisussa, koska järjestelmään ei nähty liittyvän tietokoneiden lisäksi muita laitteita. Tietokoneiden väliset yhteydet ja integraatiot määriteltiin sovellustasolla. Järjestelmää varten toteutettaville ohjelmistokomponenteille ei asetettu mitään erityisiä laitevaatimuksia tai -rajoituksia, joten valittavissa oli laajasti eri teknologioita. Ohjelmistojen ja laitteistojen liittymissä haluttiin kuitenkin käytettävän hyväksi havaittuja teknologioita ja laitteistoja. Karttapalvelun käyttäjien näkökulmasta ratkaisu haluttiin olevan laitteisto- ja selainriippumaton.

### 7.5.2 Tietoliikenneliittymät

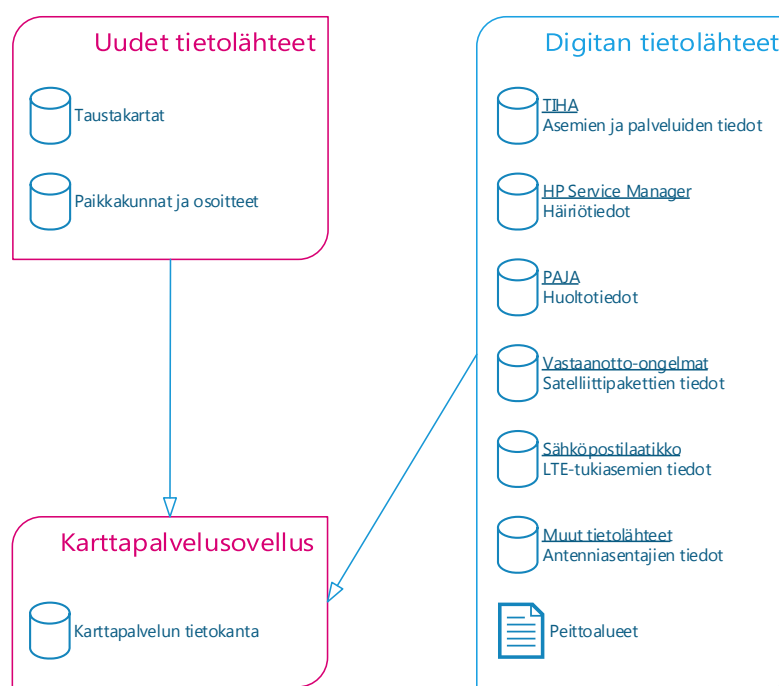
Järjestelmän ulkoiset tietoliikenneliittymät haluttiin toteuttaa ensisijaisesti salatulla HTTPS-protokollalla, mikäli liittymä toteutettiin Digitan tietoverkon ulkopuolelle internetin kautta. HTTPS:n päällä nähtiin tarve välittää lähinnä ohjelmistojen etäkutsuja ja sanomanvälitystä, jotka haluttiin ensisijaisesti toteuttaa SOAP-protokollalla. Muitakin yhteystapoja pidettiin mahdollisena, varsinkin jos etäkutsuja tarvittaisiin välittää asiakkaiden kanssa ja priorisoivat muita tietoliikenneprotokollia. Mahdollisena vaihtoehtona HTTPS:lle nähtiin SFTP ja SOAP:in lisäksi myös kaikki muut XML:ään perustuvat etäkutsuprotokollat olivat mahdollisia. Digitalla oli valmius siis tukea useampia erilaisia tietoliikenneprotokollaan perustuvia liittymätyyppejä, mutta halusi ensisijaisesti viestiä kumppaneilleen, että SOAP toteutettuna HTTPS:n päällä oli ensisijainen tietoliikenneprotokolla järjestelmien välisille integraatioille. Digitan taustajärjestelmät nähtiin kiinteänä osana karttapalveluratkaisua, vaikka niiden ensisijainen käyttötarkoitus onkin tukea muita yrityksen prosesseja. Tietoliikenneliittymät näihin järjestelmiin haluttiin nähdä kuitenkin ulkoisina liittyminä, joiden koko toteutus sisältyi karttapalveluhankkeeseen. Yhdenmukaisuudesta johtuen näissäkin liittymissä haluttiin integraatioiden ja proseduurien etäkutsun perustuvan SOAP-protokollaan. Kaikki nämä taustajärjestelmät sijaitsivat Digitan sisäverkossa, jolloin

tarvetta suojattuun HTTPS:ään ei nähty tarpeellisenä, vaan tietoliikenneliittymät haluttiin perustuvan HTTP:hen. Mikäli HTTP ei olisi mahdollinen tietoliikenneprotokolla taustajärjestelmien tai niille erikseen rakennetun rajapinnan kanssa, pidettiin erilaisia tiedostosiirtoon perustuvia ratkaisuja mahdollisina. Määrittelyvaiheessa ei haluttu rajata mihin protokoliin tai teknologioihin tiedostosiirrot perustuisivat. Tiedostosiirtojen piti olla mahdollista eri käyttöjärjestelmien ja tiedostojärjestelmien välillä.

### 7.5.3 Ohjelmistoliittymät

Ohjelmistoliittymien tarve määriteltiin suoraan järjestelmien toiminnallisten vaatimusten kautta. Kuvassa 12 on esitelty kaikki tunnistetut tietokannat ja muut aineistolähteet, joiden sisältöä karttapalvelusovelluksessa tarvitsee hyödyntää.

Karttasovellus vaatii toimiakseen luonnollisesti useita eri tarkkuuksisia karttapohjia, jotta käyttäjä voi eri tarkkuustasolla hahmottaa eri lähetysasemien ja peittoalueiden sijainnit suhteessa omaan tai haettuun sijaintiin. Lisäksi karttapalveluun haluttiin toteuttaa osoite- ja paikkakuntahaku, jolloin karttapalvelusta tarvittiin ohjelmistoliittymä tällaiseen koko Suomen osoitteiston sisältävään tietokantaan.



Kuva 12: Karttapalvelun vaatimusmäärittelyssä tunnistetut ohjelmistoliittymät.

Lähetysasemien tiedot, tiedot radio- ja televisiopalveluista sijaitsevat Digitaalisen TIHA-järjestelmän tietokannassa. Lähes kaikissa käyttötapauksissa tarvitaan tietoa lähetysasemien sijainnista ja lähetysasemilta tarjottavista televisio- ja radiokanavista. Jo määrittelyvaiheessa oli selvää, että ohjelmistorajapinta TIHA:aan tullaan

tarvitsemaan, sillä TIHA ei tarjoa valmista ohjelmistorajapintaa.

Palveluiden peittoalueiden esitys vaaditaan monen eri käyttötapauksen yhteydessä. Radion kuuluvuusalueet ja television näkyvyysalueet lasketaan Digitassa uudestaan verkon muuttuessa ja tämä tieto on saatavilla tiedostomuodossa [59]. Vastaanottoantennin suuntaamisessa haluttiin tieto parhaasta ja toiseksi parhaasta asemasta. Peittoaluekuvat eivät sisällä tätä informaatioita, joten vaatimuksena oli tuottaa sellaista peittoalueaineistoa, jossa on mukana kentänvoimakkuudet jokaisessa peittoalueen pisteessä. Valmista ratkaisua tai vaatimusta sille, miten peittoaluetiedostojen sisältämä informaatio vietäisiin karttapalveluun, ei ollut.

Tieto paikkakunnalla toimivista antenniasentajista oli Digitalla dokumentoitu, mutta tietoa ei ollut tallennettu tietokantaan. Tieto haluttiin viedä sellaiseen tietokantaan ja sovellukseen, josta tieto olisi välitettävissä ohjelmistorajapinnan kautta karttapalvelusovellukseen.

Häiriöiden reaaliaikainen esitys karttapalvelusovelluksessa vaati ohjelmistorajapinnan järjestelmään, jossa tieto Digitan verkkojen häiriöistä on ylläpidetty. Tämä HP Service Manager -järjestelmä tarjoaa valmiit SOAP-rajapinnat, joiden kautta järjestelmästä on mahdollista kysellä erilaista tietoa. Häiriöiden lisäksi myös suunnitellut huoltotyöt voivat aiheuttaa palveluihin katkoksia, joiden tieto on tallennettu Digitassa erilliseen tietokantaan. PAJA-järjestelmä pitää sisällään tietoa verkkoon tehtävistä huoltotöistä ja niiden vaikutuksista palveluun. PAJA tarjoaa toiminnallisuuden, jonka kautta PAJA voi lähettää SOAP-protokollalla tietoa ulkopuolisiin järjestelmiin silloin kun huoltotyö luodaan tai sen tietoja muutetaan.

Tieto käynnistyneestä LTE-tukiasemasta saatiin teleoperaattoreilta sähköpostilmoituksella tiettyyn Digitan sähköpostiosoitteeseen. Ilmoitus tuli lähes välittömästi kun tukiasema saatiin käyttöön. Operaattorit lähettivät koneellisesti luodun ilmoituksen, mutta sähköpostin muoto vaihteli eri operaattoreiden välillä. Vaatimuksena oli kuitenkin saada sähköpostiviesteissä olevat tukiasemat kartalle meta-tietoineen. Vaatimusmäärittelyssä ei otettu kantaa miten sähköpostissa olevat tiedot integroitaisiin osaksi järjestelmää. Operaattoreiden menettelytapaan toimittaa tukiasematiedot sähköpostitse ei voinut vaikuttaa.

Karttapalvelun sisäisessä käytössä tunnistettiin käyttötapaus, jossa kuluttajille myönnetyt satelliittipaketit haluttiin nähdä kartalla. Tämä tieto tallennettiin itse tehdyn sovelluksen kautta Microsoft Access-tietokantaan. Vaatimusmäärittelyssä haluttiin siis toteuttaa jonkinlainen rajapinta Access-tietokantaan, josta satelliittipakettien sijainnit ja myöntämiseen liittyvät tiedot välitettäisiin karttapalvelusovellukseen.

## 7.6 Muut ominaisuudet

Järjestelmän loppukäyttäjälle näkyvä suorituskyky ja käytettävyys nähtiin erittäin tärkeänä vaatimuksena. Palvelun piti vastata erittäin nopeasti käyttäjän tekemiin toimenpiteisiin ja sovellus piti olla niin helppokäyttöinen, ettei erillistä ohjeistusta tarvittaisi. Suurin riski nähtiin peittoalueissa, jotka tiedostoina olivat kohtuullisen suuria, mutta karttasovelluksessa peittoalueiden hallinta piti sujua ilman suuria viiveitä.

Karttapalvelusovelluksen on kestettävä tarpeeksi suuri yhtäaikaisten käyttäjien

määrä. Tällainen tilanne voi tulla eteen silloin kun radio- tai televisiopalvelussa on sellainen häiriö, joka saa kuluttajat Digitan internet-sivuille ja karttapalveluun selvittämään häiriön alkuperää. Käyttäjämäärän nopea kasvu voi tapahtua myös silloin, jos palvelu saa poikkeuksellista julkisuutta esimerkiksi Digitan markkinointikampanjan tai valtakunnallisen uutismedian kautta. Yhtäaikaisille käyttäjille ei määritelty tavoitemäärää, mutta karttapalvelulla arvioitiin olevan 25000 käyntiä kuukaudessa, jotka koostuvat 20000 eri käyttäjästä.

Karttapalvelusovelluksen ei haluttu lisäävään ylläpitotyötä Digitassa ja se haluttiin toteuttaa täysin ilman hallintaliittymää. Kaikki karttapalvelun sisältö päivittyisi järjestelmäintegraatioiden avulla. Todella harvoin muuttuvia parametreja tai sovellukseen liittyviä muutoksia varauduttiin tekemään joko itse tai valitun kumppanin kanssa.

Karttapalvelun sisällön haluttiin olevan mahdollisimman ajantasaista. Asemiin ja palveluihin liittyvät tiedot muuttuvat verrattain harvoin, jolloin tietojen on tuoreistettava karttapalvelussa vähintään kerran vuorokaudessa. Sama pätee myös muuta harvoin päivittyvää informaatiota, kuten antenniasentajien tietoja ja satelliittipaketteihin liittyviä tietoja. Tulevat huollot tiedetään hyvissä ajoin etukäteen ja niistä viestitään asiakkaille myös kaksi viikkoa ennen varsinaista katkosta. Tällöin huoltotietojen välitys karttapalveluun voi tapahtua kerran vuorokaudessa.

Verkossa olevat häiriöt pitää karttapalvelussa tiedottaa mahdollisimman nopeasti, joten vaatimukseksi asetettiin häiriötietojen osalta viiden minuutin viive. Toisin sanoen häiriön alkamisajan jälkeen häiriön tiedot on oltava karttapalvelussa viiden minuutin sisällä. Sama koskee myös häiriön tietosisällön muuttumista ja häiriön päättymistä.

Digitan sisäiseen käyttöön tarjottava karttapalvelu pitää olla vain saatavilla tietyille henkilöille. Julkiseen tai asiakkaille tarjottavaan karttapalveluun ei haluta toteuttaa tunnistautumismekanismia, mutta Digitan sisäinen karttapalvelu pitää olla rajatun käyttäjäryhmän saatavilla. Toinen vaihtoehto on tarjota karttapalvelua vain Digitan sisäverkossa, jolloin käytön rajoitus on toteutettu verkkotasolla sovellustason sijaan.

## 7.7 Yhteenveto karttapalvelusovelluksen vaatimusmäärittelystä

Digitan WWW-sivuille tarvittiin karttapalvelusovellus, joka tarjoaisi työkaluja kuluttajille, asiakkaille ja Digitan sisäisille toiminnoille. Tässä työssä laadittiin vaatimusmäärittely, jossa tunnistettiin joukko käyttötapauksia. Kuluttajien käyttötapaukset liittyivät televisio- ja radiopalveluiden saatavuuden esittämiseen sekä häiriö- ja huoltotilanteiden tiedottamiseen. Digitan sisäisessä käytössä karttapalvelun piti tukea kuluttajapalvelutoimintoja, joka toimi kuluttajan tukena erilaisissa kysymys- ja häiriönselvitystilanteissa. Jo vaatimusmäärittelyssä tuotiin esille järjestelmän integroitumisen tiiviisti Digitan olemassa oleviin järjestelmiin. Teknisenä suosituksena nähtiin SOAP-protokollan hyödyntäminen sovellusten välisessä sanomaliikenteessä ja etäkutsuissa. Tämän vaatimusmäärittelyn pohjalta lähdettiin toteuttamaan seuraavassa luvussa kuvattua karttapalveluratkaisua.

## 8 Karttapalveluratkaisu

Tämän luvun tarkoituksena on esitellä toteutettu karttapalveluratkaisu. Tässä työssä on keskitytty toteuttamaan karttapalveluun toimiva taustajärjestelmä, joka seuraa palvelukeskeisen arkkitehtuuriin perusperiaatteita. Luku kuvaa kaikki ratkaisun kannalta merkitykselliset olemassa olevat järjestelmät, niiden rajapinnat, sekä uudet toteutetut liiketoiminta- ja integraatiopalvelut. Lopputuloksena on toimiva palvelukeskeiseen arkkitehtuuriin perustuva hajautettu järjestelmä, joka integroituu Digitan liiketoimintaprosesseihin ja tietojärjestelmiin saumattomasti.

### 8.1 Taustajärjestelmät ja niiden rajapinnat

Hajautettuun karttapalveluratkaisuun piti liittää olemassa olevia Digitan tietojärjestelmiä. TIHA, PAJA, HP Service Manager ja Microsoft Exchange Server toimivat tietolähteinä karttasovellukselle. Lisäksi tarvittiin uusi keskitetty tietokanta paikkatiedon tallentamiseen sekä GeoServer-palvelinohjelmisto paikkatiedon muunnoksiin ja jakamiseen. Tämä luku kuvaa tarkemmin nämä tietojärjestelmät ja miten niitä on teknisesti hyödynnetty tietolähteinä.

#### 8.1.1 TIHA

TIHA on Digitan itse kehittämä ja ylläpitämä tietojärjestelmä, joka koostuu useasta alijärjestelmästä ja tietokannasta. TIHA:ssa hallitaan tietoa liittyen Digitan asemiin, palveluihin, radiomastoihin, antenneihin ja radiotaajuuksiin. Karttapalvelusovellukselle TIHA toimii lähdejärjestelmänä asemien sijainneille ja tiedoille, asemilta tarjottaville televisio- ja radiopalveluille sekä palveluihin liittyvien tietojen hallintaan.

Järjestelmä tarjoaa asemien sijainnit yhtenäiskoordinaatistossa (YKJ) sekä EUREF-FIN-koordinaatistossa. Kaikilla lähetysasemilla on julkinen nimi, jota Digita käyttää viestinnässään. Lisäksi Digita jakaa lähetysasemat eri tyyppeihin. Tyyppijaon yksinkertaistamisen vuoksi ei voitu käyttää Digitan itse käyttämiään aseman tyyppiä sellaisenaan, vaan karttapalvelusovelluksen televisionäkymässä asemien tyypit haluttiin olevan "Radio- ja TV-asema" tai "Täytelähetinasema". Täytelähetinasemalta tarjottavien televisiokanavien lukumäärä on tyypillisesti pienempi ja näkyvyysalue radio- ja TV-asemaa selvästi pienempi.

TIHA sisältää siis myös tiedon yksittäisen lähetysaseman tarjoamista televisio- ja radiopalveluista. Varsinkin radion lähetysasemien palvelutarjonta on täysin asemakohtainen, mutta myös television puolella eri lähetysasemat lähettävät eri kanavanippuja. Joissain tapauksissa kanavanippujen sisältö myös saattaa olla alueellinen. Sovellukseen piti saada näistä tiedoista tuotua asemalla tarjottavien kanavanippujen nimet, kanavanipun taajuuskanavanumero sekä lähetystaajuus. Luonnollisesti asemalla tarjottavien televisiokanavien nimet ja loogiset kanavanumerot haluttiin tuoda TIHA:sta karttasovellukseen. Looginen kanavanumero on kuluttajien tuntema kanavanumero, joka ohjaa kanavien valintaa päätelaitteissa. Radiokanavista tuotiin karttapalveluun radiokanavan nimi ja lähetystaajuus.

Digitan palveluiden, verkkojen ja laitteiden valvonta tuottaa valvontatietoa, jonka pohjalta saadaan tietoa palveluiden tilasta. Palvelujen tilatieto tuodaan TIHA:n



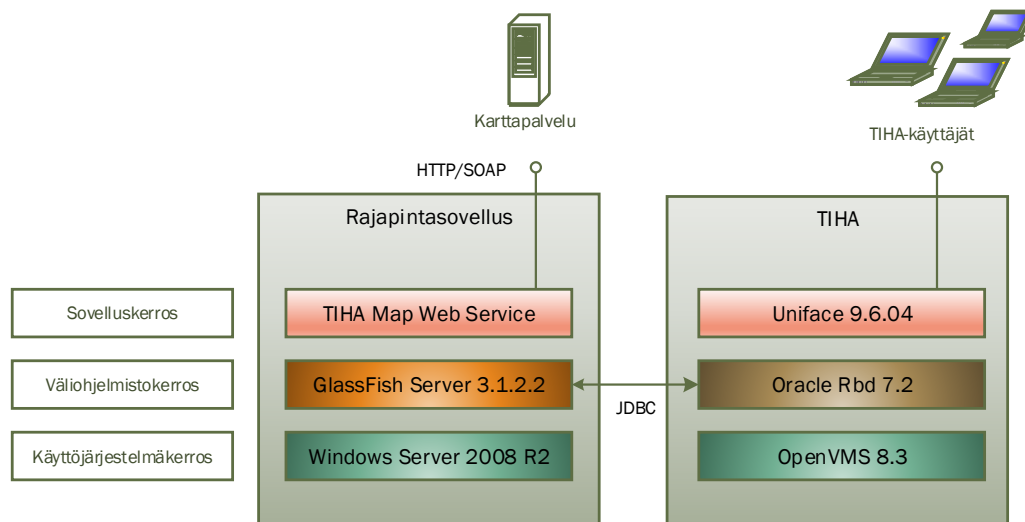
tietokantaan, jolloin tilatieto on hyödynnettävissä saman rajapinnan kautta myös karttapalvelusovelluksessa. Palvelulla on vain kaksi mahdollista tilaa: se voi olla aktiivinen tai ei-aktiivinen. Palvelun tila voi olla eri lähetysasemilla erilainen. Tätä valvontatietoon pohjautuvaa tilatietoa ei kuitenkaan haluttu tuoda karttapalvelusovellukseen sellaisenaan, koska varsinainen katkokseen liittyvä vikatiketti tehdään Digitassa aina manuaalisesti. Tilatietoa haluttiin hyödyntää vain tilanteissa, jolloin lähetysasemalla on ennakkoon suunniteltu huolto. Huollolle tyypillisesti varataan aikaa koko päivä, mutta kuluttajan kokemana alentunut palveluvaikutus kestää tyypillisesti sekunteja, minutteja tai korkeintaan tunteja. Huollon aikana halutaan viestiä kuluttajalle onko huollettava palvelu huollettavana ilman katkosta vai huollettavana ja siinä on katkos.

Itse kehitettynä sovelluksena TIHA tarjoaa mahdollisuuden tehdä järjestelmään liiketoiminnan vaatimia muutoksia ketterästi. Karttapalvelusovelluksessa haluttiin esittää jokaisessa aseman tietojen yhteydessä myös listan alueella toimivista DigitaPRO Antenniasentajista. Näitä varten järjestelmään luotiin uusi tietokantataulu. Lisäksi oli tarve hallita antenniasentajien toimialuetta jollain tavalla. Antenniasentajat itse ilmoittavat Digitalle postinnumeroalueet, joille he toimivat, mutta järjestelmään haluttiin tallentaa kunkin antenniasentajan kohdalle lista kunnista, joiden alueella he toimivat. Tämän vuoksi lisättiin järjestelmään toinen taulu, jossa antenniasentajat liitettiin eri kuntiin. Tiedon ylläpitoa varten piti myös sovellukseen luoda käyttöliittymät ja tarvittavat työkalut DigitaPRO-tietojen ylläpitämiseen. Suomen kaikki kunnat löytyivät järjestelmästä valmiina jo ennestään. Tällä relaatiolla saatiin siis pidettyä tietoa yllä Suomen kunnissa toimivista antenniasentajista. Lähetysaseman yhtenä ylläpidettynä tietona oli myös kunta, joten kunnan perusteella saatiin liitettyä jokaiseen lähetysasemaan lista alueella toimivista DigitaPRO Antenniasentajista.

TIHA ei tarjoa muille järjestelmille ohjelmointirajapintaa tai muita selkeitä liittymistapoja, joten yhtenä haasteena oli sen toteuttaminen. TIHA:n sovellusteknologia ei tarjonnut parasta mahdollista tapaa toteuttaa ensisijaisena ratkaisuna pidettyä SOAP-protokollaa. Tämän vuoksi perustettiin rajapintaa varten uusi Web Services-sovellus, jota hyödyntävä asiakassovellus olisi ensisijaisesti karttapalvelusovellus. Tietoa oli siis tarkoitus jakaa TIHA:sta XML-muodossa SOAP-protokollan avulla HTTP:n yli.

Web Services toteutettiin Java EE -teknologiaa käyttäen, jossa hyödynnettiin avoimen lähdekoodin GlassFish Server -sovelluspalvelinta. Sovelluspalvelimella oleva ohjelma hallitsee TIHA:n tietokantaan tehtäviä tietokantayhteyksiä, hakee halutun tiedon tietokannasta ja muuttaa sen XML-muotoon. Sovellus siis toteuttaa halutun SOAP-yhteyskäytännön sitä hyödyntävälle karttapalvelusovellukselle. Kuvassa 13 on esitetty TIHA-järjestelmän sekä karttapalvelua varten toteutetun rajapintasovelluksen teknologiapiinot. Sovellukset ovat toisistaan erillisiä sijaiten eri palvelimilla ja perustuen erilaiseen teknologiaan. Sovellukset kytkeytyvät toisiinsa väliohjelmistokerroksessa, jossa Javaa EE -sovelluspalvelin ohjaa sille tulleet HTTP-pyynnöt tietokantakyselyiksi JDBC-rajapinnan kautta.

Kuvassa 14 on esitelty rajapinnat, jotka toteutettiin karttapalvelusovelluksen tarpeiden näkökulmasta. Kaikki rajapinnat ovat hakutyyppejä, jotka palauttavat listan haetuista objekteista. Hakuja ei ole millään tavalla parametroitu, joten TIHA



Kuva 13: TIHA:n ja sen rinnalle toteutetun rajapinnan teknologiapiinot.

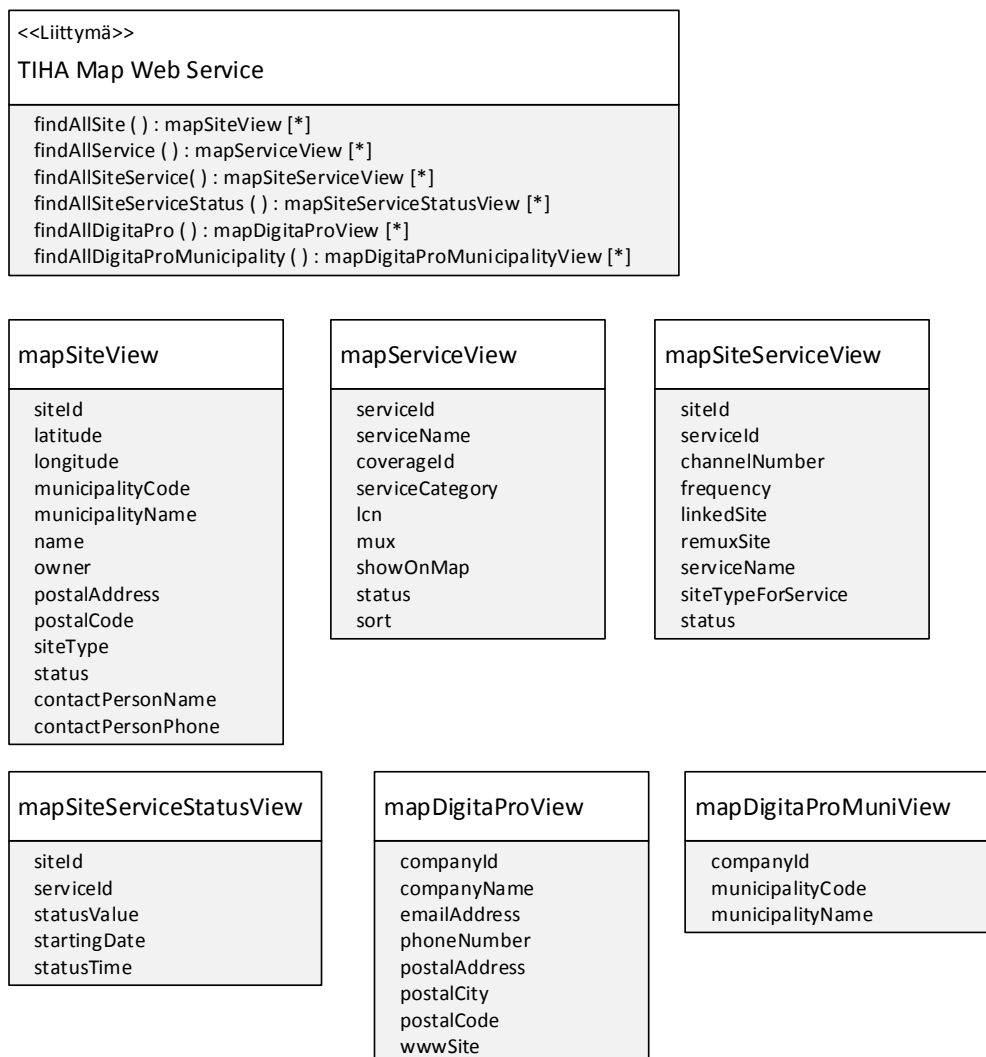
palauttaa aina kaiken haetun tiedon. Rajapintaa toteutettaessa tiedostettiin, että suunniteltu XML-skeema ei ole yhteensopiva sellaisenaan karttapalvelusovelluksen kanssa, vaan jonkinlainen XML-muunnos on tehtävä hajautetun järjestelmän josakin osassa. Näihin muunnoksiin ja integraatiokysymyksiin palataan tämän työn myöhemmässä vaiheessa.

### 8.1.2 PAJA

PAJA on Digitan työtilausten ja liikkuvien henkilöresurssien hallintaan käytetty tietojärjestelmä. Järjestelmässä hallitaan Digitan palveluihin ja verkkoihin keskittyvää ylläpito- ja rakennustoimintaa, sekä niiden aiheuttamaa palveluvaikutusten tiedottamista asiakkaille. PAJA perustuu Blå Coordinator -nimiseen ohjelmistoon.

Karttapalveluratkaisun yhtenä tarpeena oli parantaa huolloista aiheutuvien katkosten tiedottamista asiakkaille ja kuluttajille. Huoltotyöt ovat mallinnettu PAJA:ssa työtilauksina, jolloin huoltotöihin liittyvälle tiedolle PAJA toimii ensisijaisena lähdejärjestelmänä. Yhteen huoltotyöhön voi liittyä yksi tai useampi lähetykatkos tai muu palvelunlaatuun vaikuttava muutos. Yksi katkos voi taas koskettaa yhtä tai useampaa televisio- tai radiopalvelua. Lisäksi katkokselle voidaan määritellä aina vara-aika, mikäli huoltotyötä ei pystytä varsinaisena sovittuna ajankohtana tekemään. Digita viestii verkoissa olevista katkoksista asiakkailleen ja kuluttajille viimeistään kaksi viikkoa ennen huoltotyön ja katkoksen alkamisajankohtaa. Digita voi myös viestiä huoltotöistä, vaikka niillä ei olisikaan havaittavia palveluvaikutuksia.

Huoltotyö PAJA:ssa kohdistuu aina sille lähetyksasemalle, jossa huoltotyö fyysisesti tehdään. Huoltotyön aiheuttama katkos voi kuitenkin vaikuttaa laajemmalla alueella, kuin vain yhden aseman peittoalueella. Esimerkiksi televisiosignaalia releoidaan eli edelleen lähetetään pääasemilta täytelähetinasemille, jolloin pääasemalla tapahtuva

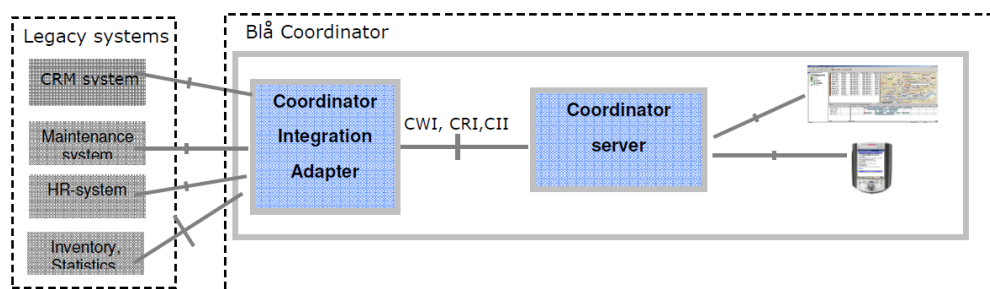


Kuva 14: TIHA-järjestelmään toteutettu Web Service -rajapinta.

katkos vaikuttaa releoinnin kautta täytelähetinasemiin. PAJA ja TIHA jakavat integraation kautta saman tiedon Digitan lähetyksasemista ja asemilta lähtetettävistä televisio- ja radiopalveluista.

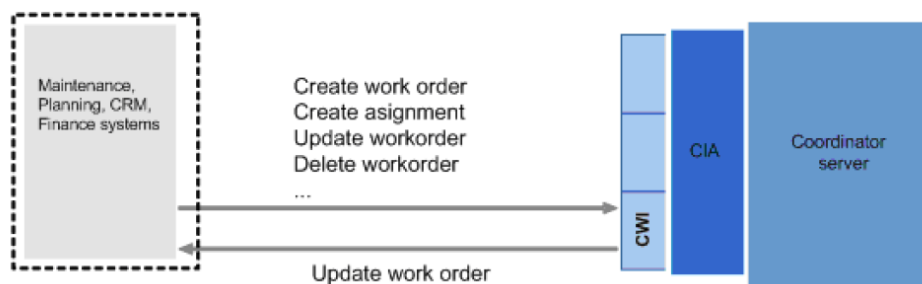
Coordinator Integration Adapter (CIA) on PAJA:n integraatiomoduli, joka sitoo PAJA:n sovelluspalvelimen yhteen taustajärjestelmien tai muiden kommunikoivien sovellusten kanssa. CIA hallitsee sanomaliikenteen kumpaankin suuntaan ja pystyy muuntaa sanomia haluttuun formaattiin. CIA SDK antaa mahdollisuuden muokata ja tehdä räätälöityjä ohjelmistomoduuleja, jotka mahdollistavat entistä monimutkaisemmat sanomamuunnokset. CIA tarjoaa kommunikaatiotavaksi HTTP:n tai JMS:n yli SOAP:ia tai muuta XML-pohjaista protokollaa. CIA sisältää myös jonojärjestelmän, joka huolehtii sanomien säilymisestä myös tilanteissa, joissa vastaanottava järjestelmä ei ole tavoitettavissa. Kuvassa 15 on esitetty miten PAJA kytkeytyy muihin sovel-

luksiin CIA-moduulin kautta. PAJA tarjoaa CIA:lle kolme ohjelmointirajapintaa: CWI, CRI ja CII. CWI on työtilausrajapinta, CRI henkilöresurssirajapinta ja CII on rajapinta järjestelmän perustietojen hallintaan, mutta myös muista järjestelmistä tulevien pyyntöjen hallintaan.



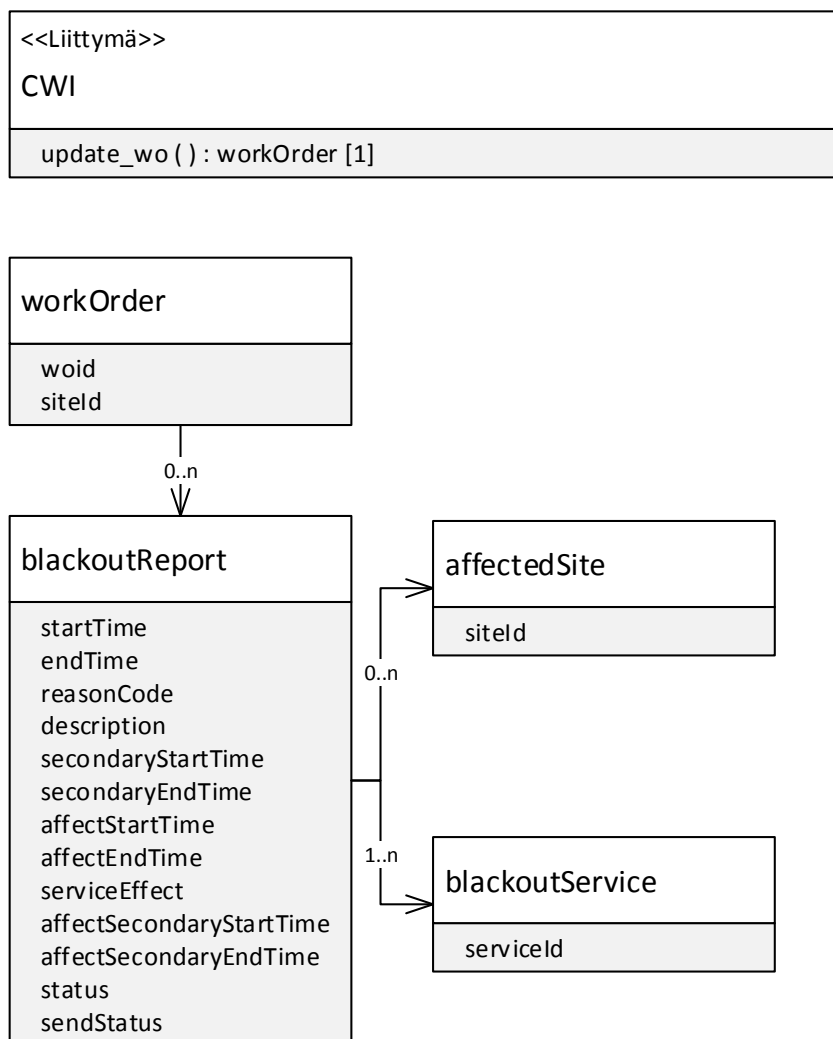
Kuva 15: PAJA-järjestelmän CIA-integraatioadapteri.

Karttapalvelusovellukseen vietävien huoltojen kannalta CWI-rajapinta osoittautui hyödyllisimmäksi. CWI ei mahdollista tietokyselyitä kuten vaikka TIHA:an toteutettu rajapinta, vaan se ottaa ulkoisista järjestelmistä vastaan vain sanomia, joilla luodaan, muutetaan tai poistetaan järjestelmässä olevia työtilauksia tai toimeksiantoja. CWI-rajapinnan käyttö ei rajoitu ainoastaan sanomien vastaanottamiseen ulkoisista järjestelmistä, vaan sen kautta voidaan lähettää sanomia myös ulospäin. Jo aiemmissa projekteissa oli toteutettu sanoman lähettäminen jokaisen PAJA:ssa tehdyn tallennuksen jälkeen. Karttapalvelusovelluksen kannalta jokainen tallennus ei ole kiinnostava, koska useimmat työtilaukset eivät sisällä karttapalvelussa esitettävää informaatiota ollenkaan tai sovelluksen kannalta oleellinen tieto muuttuu verrattain harvoin. CWI-rajapinnan perusajatus on esitetty kuvassa 16.



Kuva 16: CWI-työtilausrajapinta.

Kuvassa 17 on esitetty CWI-rajapinnan luokkamalli karttapalvelusovelluksen näkökulmasta. CWI lähettää jokaisen PAJA:ssa tehdyn tallennuksen jälkeen kaiken mahdollisen tiedon sisältävät XML-sanoman, joista kuvaan on karsittu vain karttapalvelusovelluksen kannalta oleellinen tieto. Työtilaus pitää sisällään yhden lähetyksiaseman. Jos huoltotyöstä aiheutuu katkos myös muille asemille, ilmaistaan nämä asemat `affectedSite`-luokalla. Luokka `blackoutService` ilmaisee kaikki palvelut, joihin huoltotyö vaikuttaa.



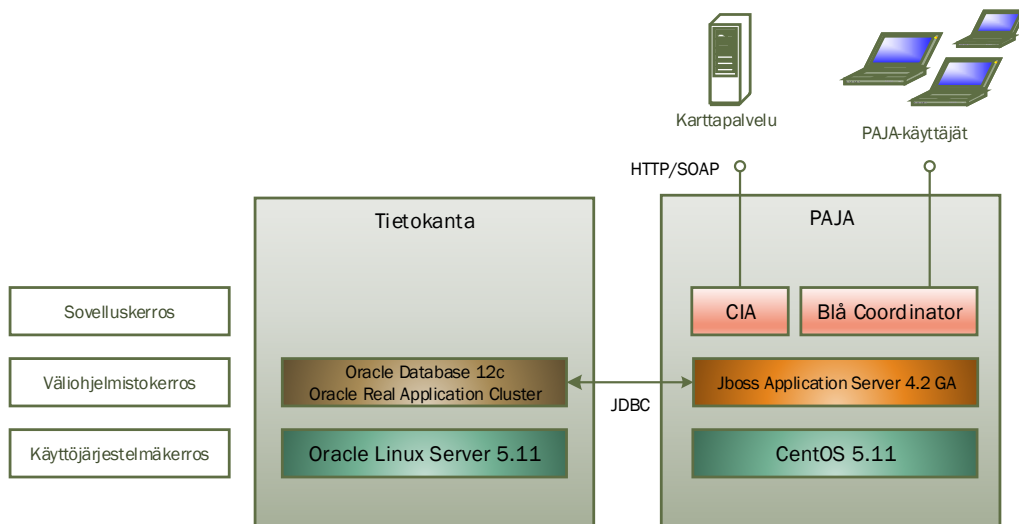
Kuva 17: PAJA-järjestelmän Web Service -rajapinnan sanomamalli (CWI).

### 8.1.3 HP Service Manager

HP Service Manager on Digitassa IT-palveluiden hallintaan ja johtamiseen käytetty järjestelmä. Järjestelmää käytetään erityisesti Digitan palvelukeskuksessa, joka toimii myös kontaktipisteenä asiakkaiden suuntaan. Järjestelmään kirjataan kaikki Digitan palveluissa olevat häiriöt, tuli häiriöilmoitus sitten valvontajärjestelmästä, asiakkaalta tai muusta lähteestä. Järjestelmä lähettää häiriötilanteissa asiakkaille sähköposti- ja tekstiviesti-ilmoituksia.

Häiriötiketti sisältää aina tiedon mitä lähetysasemaa ja palvelua häiriö koskee. Yhdelle häiriötiketille on mahdollista syöttää yksi palvelu ja useampi lähetysasema, joten laajavaikutteisimmassa häiriötilanteissa verkonvalvoja luo useita häiriötikettejä tietojärjestelmään kustakin palvelusta tai kanavanipusta erikseen.

Service Manager toimii karttapalvelusovellukselle häiriötietojen lähdejärjestel-

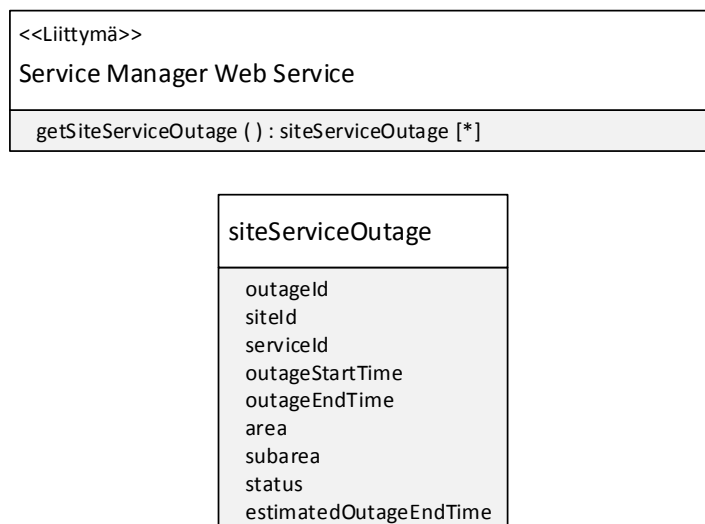


Kuva 18: PAJA-järjestelmän teknologiapino.

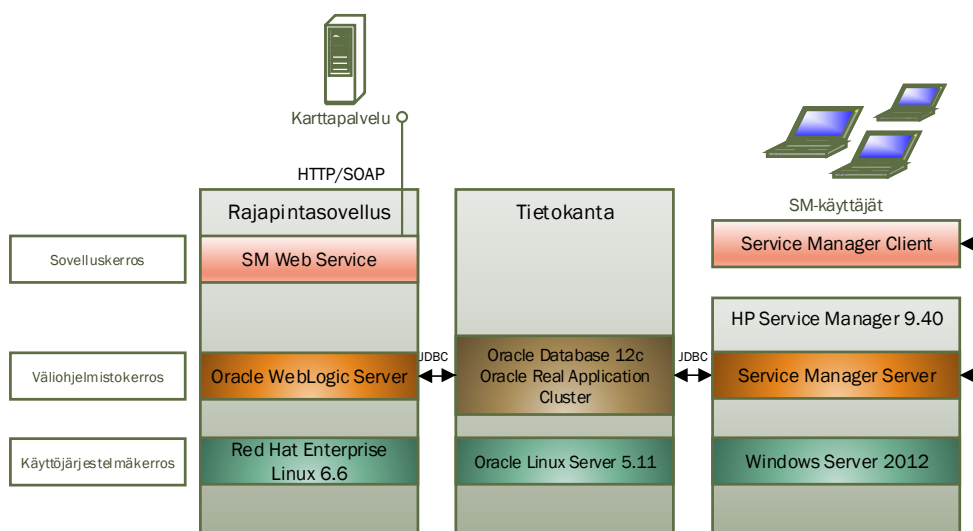
mänä. Karttapalvelusovellukseen halutaan tieto käynnissä olevista häiriöistä tai vuorokauden sisällä päättyneistä häiriöistä kuitenkin siten, että asiakkaan omista järjestelmistä johtuvat häiriöt halutaan suodattaa pois. Lisäksi, karttapalvelussa halutaan näyttää täsmälleen ne katkokset, joista asiakasta on tiedotettu sähköpostitse tai tekstiviestillä, jotta asiakasviestintä ja sovelluksen kautta tehtävä julkinen viestintä eivät olisi ristiriidassa keskenään.

Service Manager tarjoaa rajapinnan Web Service -palveluna, jota kautta on mahdollista kysellä häiriötikettien tietoja halutuilla parametreilla. Tässä tapauksessa valmiin ohjelmointirajapinnan käyttäminen ei kuitenkaan tullut kysymykseen, koska kyselyä ei ollut mahdollista rajata vain asiakasviestinnän piirissä oleviin häiriötiketteihin ainakaan yksinkertaisella tavalla.

Vaihtoehtoisena ratkaisuna toteutettiin erillinen Web Service -sovellus, joka tarjosi juuri halutun palvelun. Tämä Java J2EE -sovellus tekee tietokantakyselyn Service Managerin tietokantaan, muokkaa tulosjoukon XML:ksi ja palauttaa HTTP-pyyynnön tehneelle asiakassovellukselle. Kuvassa 20 on esitetty tapa, miten Web Service kytkeytyy Service Manageriin ja sen tietokantaan. Sovelluksen tekemät tietokantakyselyt eivät ole asiakassovelluksesta käsin mitenkään parametroitavia, vaan palautettu tietojoukko koostuu aina vain niistä häiriötiketeistä, jotka ovat kyselyn hetkellä aktiivisia, tai jotka on ratkaistu kuluneen 24 tunnin aikana. Rajapintapalvelua ei siis suunniteltu tässä kohtaa mitenkään yleiseksi, vaan palvelemaan todennäköisesti vain karttapalvelusovelluksen tarpeita. Tässäkään tapauksessa tuotettu XML-dokumentti ei ole yhteensopiva karttapalvelusovelluksen kanssa, vaan Web Servicen ja karttapalvelusovelluksen välissä on vielä tehtävä sanomamuunnos. Tämä Web Service -rajapinta on käytettävissä SOAP-protokollalla HTTP-protokollan yli. Rajapinnan luokkamalli on esitetty kuvassa 19.



Kuva 19: HP Service Manager -järjestelmään toteutettu Web Service -rajapinta.



Kuva 20: HP Service Manager -järjestelmän teknologiapino.

#### 8.1.4 Microsoft Exchange Server

Microsoft Exchange Server on Digitassa käytössä oleva sähköpostipalvelinohjelmisto. Se on asennettuna Digitan omassa palvelininfrastruktuurissa.

Matkapuhelinoperaattorit lähettävät tiettyyn sähköpostiosoitteeseen viestin jokaisesta käyttöön otetusta 800 MHz:n taajuusalueen LTE-tukiasemasta. Operaattorit lähettävän vakimuotoisen viestin, mutta jokaisella operaattorilla viestin muoto poikkeaa toisistaan. Tieto sähköpostissa on CSV- tai XML-muotoa ja sijoitettu joko

sähköpostin viestikenttään tai liitetiedostoksi.

Exchange Serverissä on Exchange Web Services -rajapinta (EWS), joka tarjoaa toiminnallisuuksia sovelluksille kommunikoida sähköpostipalvelimen kanssa. EWS tarjoaa pääsyn käytännössä samaan tietoon, joka on tarjolla esimerkiksi Microsoft Office Outlook -ohjelman kautta. SOAP tarjoaa EWS:lle sanomaliikenneympäristön sanomien välittämiseen asiakassovelluksen ja Exchange-palvelimen välillä. [60]

EWS siis tarjoaa Web Services -rajapinnan, jonka kautta voidaan etsiä ja noutaa sähköposteja tietyillä hakuehdoilla. Löydetyistä sähköposteista voidaan hakea haluttua tietoa tietystä sähköpostin kentästä. Tässä tapauksessa sähköposteista haetaan viestikentän sisältö tai liitetiedostot, jotka ovat CSV- tai XML-muotoista tietoa. Nämä tietomuodot pitää vielä parsia, jotta tieto on hyödynnettävissä karttapalveluratkaisussa.

### 8.1.5 Palveluiden peittoalueet

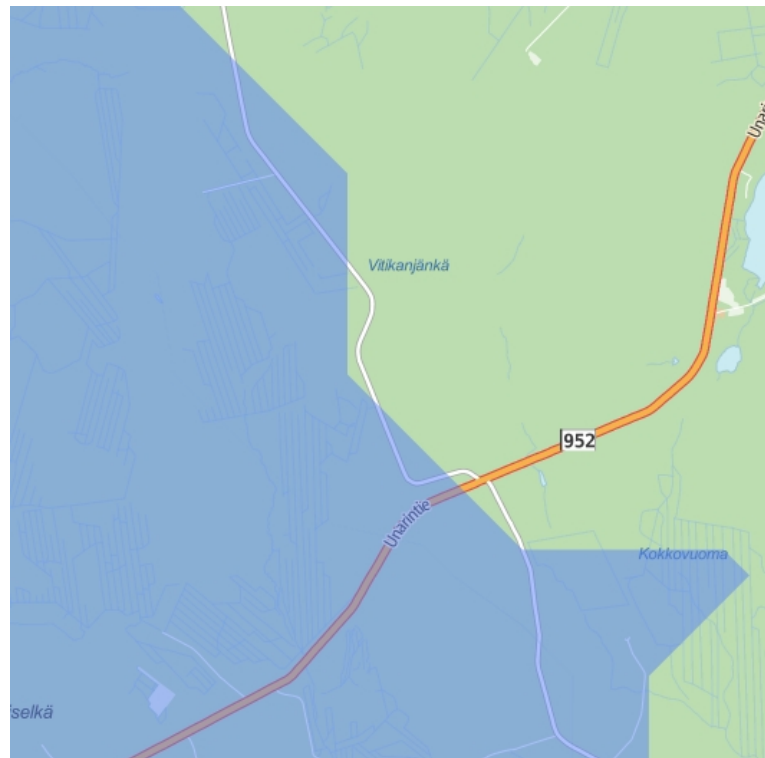
Palveluiden peittoalueiden vieminen karttapalvelusovelluksen käytettäväksi osoittautui muuta aineistoa haastavammaksi tehtäväksi. Kun muu aineisto sijaitsi standardien rajapintojen takana tai ainakin relaatiotietokannoissa, ei peittoaluetietoa ollut vastaavalla tavalla tallennettu tietokantaan tai muuallekaan, josta karttapalvelun olisi helppo sitä käsitellä.

Peittoalueet ovat laskennallisesti tuotettua aineistoa, jossa laskenta perustuu radioaaltojen etenemisen ennustamiseen. Peittoalueen laskennassa käytetään yleisesti hyväksytyjä ja tunnettuja Kansainvälisen televiestintäliiton (ITU) laskentasuosituksia. Suositus pitää sisällään useita erilaisia diffraktiomalleja. Lasketut peittoalueet pitävät sisällään varmuusmarginaaleja ja peittoaluekuvat ovat häiriörajoitteisia. Häiriörajoitteisissa peittoalueissa on huomioitu vieressä olevien peittoalueiden samantaajuiset häiriötä aiheuttavat signaalit. ITU:n laskentasuositusten pätevyys on varmistettu Digitassa kenttämittauksilla. Laskenta tuottaa rasterimuotoista aineistoa, joka laskee aseman lähettämän signaalin kentänvoimakkuuden 500 m x 500 metriä olevissa pikseleissä erikseen. Peittoalue rajataan kentänvoimakkuuden mukaan siten, että  $49 \text{ dB}\mu\text{V}/\text{m}$  muodostaa peittoalueen rajan. Signaalin vastaanotto lasketun peittoalueen ulkopuolella onnistuu hyvin todennäköisesti paremmalla antenniratkaisulla. [59]

Laskennassa syntynyt rasterimuotoinen peittoalue muunnettiin MapInfo Professional -ohjelmistolla vektorimuotoiseksi peittoalueeksi, jotta sen yhteensopivuus sovelluksen kanssa voitiin varmistaa. Peittoalueet tallennettiin MapInfo TAB-muotoon, joka on yksi paikkatietojärjestelmien käyttämä tiedostomuoto paikkatiedotiedon tallentamiseen. Peittoalueeseen lisättiin myös ominaisuustietona mm. peittoalueen tunnus, jotta peittoalue oli yhdistettävissä tiettyyn televisio- tai radiopalveluun. Muodostetut peittoalueet olivat pyöreästä yleisnäköisestä huolimatta monikulmioita 21, joka muodostui koordinaattipisteistä ja niiden välisistä janoista. Vektorimuotoinen MapInfo TAB tiedoston peittoalue tallennettiin sellaisenaan lopuksi relaatiokantaan.

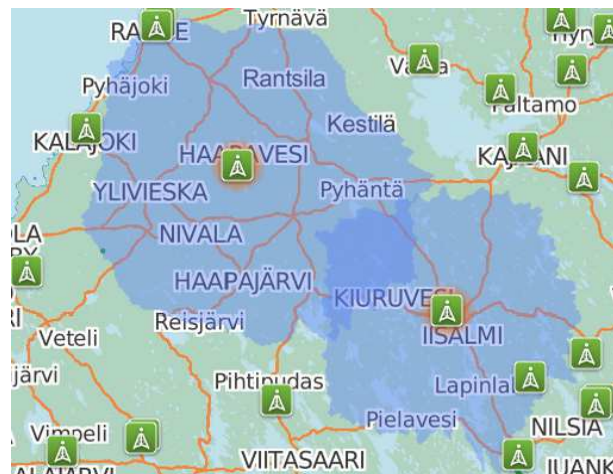
Kriittisin tarve oli tuottaa karttapalveluun käyttöliittymässä näkyvät peittoalueet, jotka kuvasivat yksittäisen kanavan peittoalueen yksittäisellä asemalla. Näiden asemakohtaisten peittoalueiden lisäksi tarvittiin myös koko maan peittoalue ka-





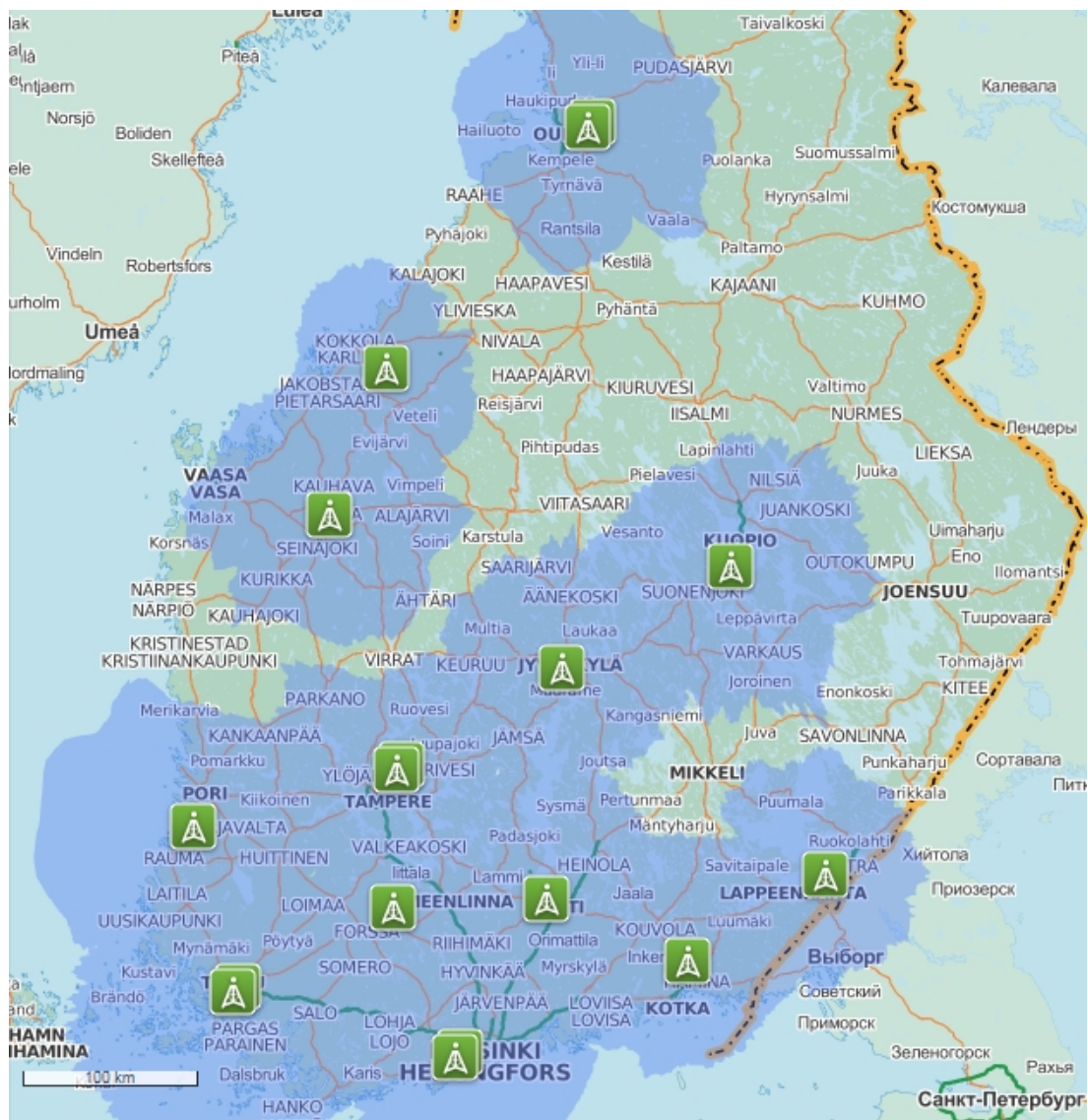
Kuva 21: Monikulmaisen peittoalueen reuna.

navittain, joka siis käytännössä on sama kaikkien asemakohtaisten peittoalueiden muodostama alue. Koko maan peittoalueessa ei haluttu näyttää miten asemakohtaiset peittoalueet leikkaavat toisensa. Koko maan peittoaluetta varten piti laskea oma peittoalue, jossa peittoalueiden leikkaukset on eliminoitu.



Kuva 22: Toisensa leikkaavat asemakohtaiset peittoalueet.

Digitan sisäisessä käytössä aseman peittoalue haluttiin jakaa vielä kahteen osaan:

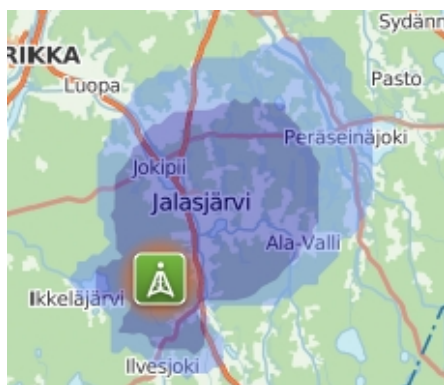


Kuva 23: Yhden kanavan peittoalue koko maan osalta.

aseman ympärillä olevaan ydinalueeseen ja peittoalueen reunamilla olevaan lievealueeseen. Tämä toteutettiin siten, että lasketusta aineistosta suodatettiin ydinalue ja lievealue toisistaan signaalin vastaanoton onnistumisen todennäköisyyden perusteella [59]. Näin muodostettiin sisempi ydinalue ja rengasmaisen lievealue, joka on esitetty kuvassa 24. Koska Digitan sisäiseen ja kuluttajille tarjottavaan karttapalveluun ei haluttu tehdä erillisiä peittoalueita, löytyy kumpikin alue myös kuluttajille tuotetusta karttapalvelusta. Siinä vain kumpikin alue on värjätty samalla värillä, jolloin eroa näiden alueiden välillä ei voida nähdä.

### 8.1.6 Paikkatietokanta

Peittoaluegeometrioita varten luotiin uusi paikkatietokanta perustuen Oracle Spatial-tietokantateknologiaan. Digitan verkkosuunnittelujelmistot mahdollistivat vektori-



Kuva 24: Aseman peittoalueen ydin- ja lievealueet.

muotoisten geometrioiden tallentamisen Oracle Spatial-tietokantaan, jolloin erillistä käyttöliittymää peittoalueiden siirtämiseen tietokantaan ei tarvittu.

Verkkosuunnitteluohjelmisto tallentaa yksittäisen peittoalueen aina yhdeksi tietokantatauluksi. Tietokantataulu nimetään sen mukaan miten peittoaluetiedosto on nimetty suunnitteluohjelmassa. Tällainen tallennustapa tuo haasteita tiedon käsittelyyn jatkossa. Jos halutaan tehdä ohjelma, joka lukee tietoa paikkatietokannasta automaattisesti, ei ohjelma voi tietää minkä nimisiä tietokantatauluja kantaan on tulossa ja mistä tietokantatauluista tietoa pitäisi hakea. Toiseksi, tietokantasuunnittelun näkökulmasta on hankalaa jos jokainen peittoalue on oma taulunsa. Taulujen määrä lisääntyy peittoalueiden määrän lisääntyessä. Näiden ongelmien välttämiseksi luotiin tausta-ajo, joka kopioi sovitulla tavalla nimettyjen tietokantataulujen sisällön yhteen peittoalueita varten perustettuun tauluun ja sen jälkeen poistaa alkuperäisen taulun kannasta. Tällä tavalla koko peittoaluedata saatiin kerättyä yhteen tauluun. Tämä helpottaa hakujen tekemistä kantaan, kun kaikkien lähetyksien peittoalueet löytyvät samasta taulusta. Ratkaisu myös estää samojen peittoalueiden tallentumisen erinimisinä eri tauluihin, koska tausta-ajo tunnistaa peittoalueen sisällön perusteella onko peittoalue täysin uusi peittoalue vai korvaako se jo olemassa olevan peittoalueen.

### 8.1.7 GeoServer

GeoServer on Java-pohjainen palvelinohjelmisto, joka mahdollistaa paikkatietodatan esittämisen ja muokkaamisen. GeoServer tukee Open Geospatial Consortiumin (OGC) määrittelemiä avoimia standardeja, joiden avulla paikkatietodataa voidaan muuntaa ja jakaa. WMS- ja WFS-standardeja toteuttamalla, GeoServer voi tarjota karttoja ja paikkatietoa monessa eri formaatissa. Se on myös ilmainen sovellus, joka perustuu Geotools-kirjastoon. Geotools on Java-kirjasto, joka tarjoaa työkaluja paikkatiedon käsittelyyn. GeoServeriin on integroitu OpenLayers-kirjasto karttojen tekemistä ja esittämistä varten.

GeoServeriin voidaan tuoda paikkatietoa monesta eri formaatista ja lähteestä. Yksi tuettu lähdeformaatti on Oracle Spatial -tietokantateknologia. Ratkaisussa toteutettiin GeoServeriin radion ja television peittoalueille omat projektit, joihin

lisättiin karttatasoksi radion ja television peittoalueet tietokantahakuina suoraan toteutettuun paikkatietokantaan. Peittoalueiden haku toteutettiin mahdollisimman parametroituna. GeoServeristä voitiin hakea tasoksi tietyn aseman peittoalue, tietyn kanavan peittoalue tai tietyn päivämäärän jälkeen muuttuneet peittoalueet. Näitä eri hakukriteerejä voitiin yhdistellä haussa.

Radion ja television peittoalueet julkaistiin GeoServeristä WFS-rajapintana. Tämä ohjelmointirajapinta mahdollisti standardin mukaiset WFS-kyselyt, jotka palauttivat halutun sisällön paikkatietokannasta GML-muodossa. WFS-kyselyssä oli käytössä samat parametrit, mitkä karttaprojektin tasossa oli määritelty parametreiksi. GeoServer pystyy tekemään myös koordinaattimuunnokset. Peittoalueet oli tallennettu tietokantaan koordinaattijärjestelmässä WGS84, mutta WFS-rajapinnan kautta palautetut peittoalueet muutettiin yhtenäiskoordinaatistoon (YKJ).

## 8.2 Järjestelmäarkkitehtuuri

Kuvassa 25 on ryhmitelty karttapalveluratkaisun elementit palvelukeskeisen arkkitehtuurin mukaisiin kerroksiin. Karttapalveluratkaisussa tarvittavat resurssit ja palvelut on jaettu viiteen tasoon: liiketoimintaprosessit, julkiset liiketoimintapalvelut, sisäiset liiketoimintapalvelut, integraatiopalvelut ja yritysresurssit. Poikkeuksena palvelukeskeisen arkkitehtuurin elementteihin kuvassa 6, karttapalveluratkaisu on jaettu viiteen kerrokseen neljän sijasta. Liiketoimintapalvelukerros on jaettu kahteen eri kerrokseen, koska on haluttu eriyttää organisaation ulkopuolelle tarjottavat liiketoimintapalvelut, pelkästään sisäisistä liiketoimintapalveluista. Lisäksi, julkiset liiketoimintapalvelut on koostettu sisäisistä liiketoimintapalveluista, jolloin julkisten liiketoimintapalveluiden asettaminen ylemmälle hierarkiassa sisäisten liiketoimintapalvelujen päälle on perusteltua.

Kuvaan elementtien väriä on valittu sen mukaan, onko komponentti ollut käytössä organisaatiossa jo ennen karttapalveluratkaisua. Punaiset komponentit eivät ole olleet käytössä ennen karttapalveluratkaisua, jolloin näiden toteuttaminen on kuulunut olennaisena osana karttapalveluratkaisuun ja tähän työhön. Vihreät komponentit taas ovat olleet yrityksen omaisuutta jo ennen karttapalvelusovelluksen käyttöönottoa, mutta komponenttien käyttäminen osana karttapalveluratkaisua on välttämätöntä. Kuvasta voidaan nähdä, että useimmat karttapalveluratkaisun kannalta tärkeät liiketoimintaprosessit ovat olleet käytössä jo ennen sovelluksen käyttöönottoa. Toisaalta myös karttapalveluratkaisun tarvitsemat yritysresurssit ovat olleet käytössä jo aiemmin. Yritysresurssien ja liiketoimintaprosessien väliset kerrokset ovat kokonaan puuttuneet, joten taustajärjestelmän suunnittelussa ja toteutuksessa on keskitytty pääasiassa näiden palvelukeskeisen arkkitehtuurin liiketoiminta- ja integraatiopalvelujen kehittämiseen.

Liiketoimintaprosessit kuvassa 25 ovat niitä prosesseja, jotka liittyvät karttapalvelun käyttöön jollain tavalla. Häiriönhallinta, muutoksenhallinta ja peittoaluesuunnittelu tuottavat karttapalveluun pääasiassa tietoa, mutta sovelluksen käyttö ei sisälly näihin prosesseihin. Karttapalvelu on taas olennainen osa Digitan ulkoista viestintää. Tiedotteissa viitataan karttapalvelusovellukseen, jossa häiriöiden ja huoltojen palveluvaikutuksen ovat nähtävissä selkeämmin kuin kirjallisessa viestinnässä. Osaksi

ulkoista viestintää voidaan nähdä myös Digita Info -kuluttajapalvelu, jossa karttapalvelusovellus on tärkeässä roolissa kuluttajaviestinnässä ja häiriöiden selvityksessä. Karttapalvelun tuoreistus on karttapalvelun sisällön päivittävä prosessi, joka on suurimmaksi osaksi automatisoitu.

Julkiset liiketoimintapalvelut ovat karttapalvelusovelluksen näkemät rajapinnat taustajärjestelmäkokonaisuuteen. Nimestään huolimatta palvelut eivät ole ainakaan toistaiseksi täysin julkisia, vaan tietoliikenneyhteydet ovat sallittuja vain tietyistä IP-osoitteista. Julkiset liiketoimintapalvelut on määritelty siten, että tulevaisuudessa niitä voisi tarjota myös muille sovelluksille tai kumppaneille.

Sisäiset liiketoimintapalvelut ovat laajuudeltaan pienempiä palveluita, kuin julkiset liiketoimintapalvelut. Julkiset liiketoimintapalvelut on hyvin pitkälle koottu näiden palveluiden pohjalta. Tavoitteena on ollut tuottaa helposti käytettäviä ja dokumentoitavia julkisia palvelukokonaisuuksia. Palveluihin liittyvät kompleksisemat ja tietojärjestelmäkohtaiset yksityiskohdat on pyritty piilottamaan enemmän sisäisten liiketoimintapalveluiden kerrokseen.

Integraatiopalvelut välittävät tietoa yritysresurssien rajapintojen tai liiketoimintapalvelujen välillä. Integraatiopalvelun yhtenä perustehtävänä on muuntaa tieto lähtömuodosta kohderajapinnan tai -liiketoimintapalvelun muotoon. Integraatiopalvelut voidaan tässäkin tapauksessa jakaa kahteen eri luokkaan sen mukaan, missä vaiheessa niitä hyödynnetään karttapalvelun tietojen tuoreistusprosessissa. Ensimmäisen vaiheen integraatiopalvelut kytkeytyvät yritysresurssien rajapintojen ja keräävät kaiken sisällön samaan GIS-relaatiotietokantaan. Toisen vaiheen integraatiopalvelu kytkeytyy GIS-tietokantaan ja sisäisiä liiketoimintapalveluja hyödyntäen muuntaa sisällön skeemamuunnoksilla julkisen liiketoimintapalvelun muotoon. Tällä tavalla julkisten liiketoimintapalveluiden käyttö kuormittaa vain toisen vaiheen integraatiopalveluita. Tietokantatasolla kuormitus kohdistuu vain GIS-tietokantaan. Tällä tavalla julkisten liiketoimintapalvelujen lisääntynyt kuormitus ei näy milloinkaan operatiivisten yritysresurssien lisääntyneessä kuormituksessa.

### 8.3 Järjestelmän sisäiset integraatiot

Vaatusmäärittelyn mukaisesti karttapalveluun liittyvät ulkoisien kumppanien tekemät haut eivät saisi kuormittaa operatiivisia tietojärjestelmiä. Tämän vuoksi kaikki karttapalvelussa esitettävä tieto piti koostaa yhteiseen GIS-tietokantaan, josta karttapalvelusovellus lopulta noutaisi tiedot Web Services -rajapintojen kautta.

Tietojen koostamisessa yhteiseen tietokantaan eli ensimmäisen vaiheen integraatiossa tietojen koostaminen perustui Digitan palveluväylään (ESB). Digitan palveluväylällä toteutettiin tarvittavat integraatiopalvelut, mutta myös luotiin tiettyjä liiketoimintapalveluita ja tietokanta-adaptereita, koska toivottuja Web Service -rajapintoja ei ollut tietojärjestelmissä sisäänrakennettuina. Ainoan poikkeuksen teki TIHA, jonka Web Service rajapintaa ei toteutettu palveluväylän työkaluilla.

Kuvassa 26 on esitetty integraation ensimmäisen vaiheen arkkitehtuuri. Kuvan punaiset komponentit edustavat ohjelmistokomponentteja, jotka tämän työn yhteydessä toteutettiin. Muut komponentit olivat osana yrityksen infrastruktuuria jo tätä aiemmin. Informaatiovirran suunta kuvassa on alhaalta ylöspäin.



Kaikki taustajärjestelmien tieto muutettiin tavalla tai toisella XML-muotoiseksi ja tietojen siirrossa käytettiin SOAP-protokollaa. TIHA, PAJA ja Exchange tarjosivat rajapintojensa kautta tietoa suoraan SOAP:lla. HP Service Manager tarjoaa myös SOAP- ja REST-rajapintoja, mutta tässä yhteydessä se ei kuitenkaan tarjonnut sellaista rajapintapalvelua, jonka kautta haluttua tietoa olisi saatu helposti. Sen vuoksi luotiin palveluväylällä tietokanta-adapteri, joka muuntaa SQL-kyselyn tuloksen adapterin oman XML-skeeman mukaiseen muotoon. Adapterin kautta pystyy tietoa hakemaan aivan samoin kuin normaalista SOAP-rajapinnasta. Access-tietokannassa olevan tiedon hakemista varten piti ensin siirtää tietokantatiedosto erilaisten käyttöjärjestelmäalustojen välillä SMB/CIFS-protokollalla. Tämän jälkeen toteutettiin palveluväylän sisälle Java-sovellus, joka haki Microsoft Access-tietokantatiedostosta halutun tiedon. Tätä tarkoitusta varten palveluväylä ei tarjonnut valmiita työkaluja tai adaptoreita. Palveluväylä pystyy kutsumaan Java-sovelluksia ja käyttämään sovelluksen palauttamia tuloksia osana integraatiota.

Integraatiopalvelu (kuvassa 26 "Data Integration Proxy Service") kutsuu liiketoimintapalveluita ja saa vastauksena karttapalvelun sisältötietoa XML-muodossa. Nämä XML-dokumentit noudattavat lähdejärjestelmien XML-skeemoja, eivätkä ne ole sellaisinaan käyttökelpoisia. Integraatiopalvelu tekee sanomamuunnokset haluttuun XML-skeemaan ja reitittää sanomat oikealle synkronointipalvelulle, joka taas tuoreistaa karttapalvelun yhteisessä GIS-tietokannassa olevat tiedot tietokanta-adapterin kautta. Tietokanta-adapteri ensin tyhjentää päivitettävän tietokantataulun, jonka jälkeen välittömästi lisää sinne tuoreimmat tiedot. Integraatiopalvelu sisältää myös tiettyjä liiketoimintasääntöjä, joilla esimerkiksi suodatetaan pois tietoa, jota ei haluta viedä GIS-tietokantaan.

Toinen integraatiopalvelu "Orchestrarion Proxy Service" vastaa integraatioiden orkestroinnista. Muuttuneita karttapalvelun tietoja pitää viedä GIS-relaatiokantaan tiettyssä järjestyksessä, jotta taulujen väliset relaatiot pysyvät eheänä. Ei voida esimerkiksi viedä lähetysasemaan liittyviä häiriötietoja kantaan, mikäli kyseistä lähetysasemaa ei kannassa vielä ole. Toinen syy orkestrointiin on se, että osa tiedoista pitää päivittyä nopeammin kuin toiset. Televisio- ja radiopalveluihin liittyvät häiriötiedot päivittyvät nopealla syklillä, kun taas harvemmin muuttuvaa tietoa viedään GIS-tietokantaan pidemmällä sykleillä.

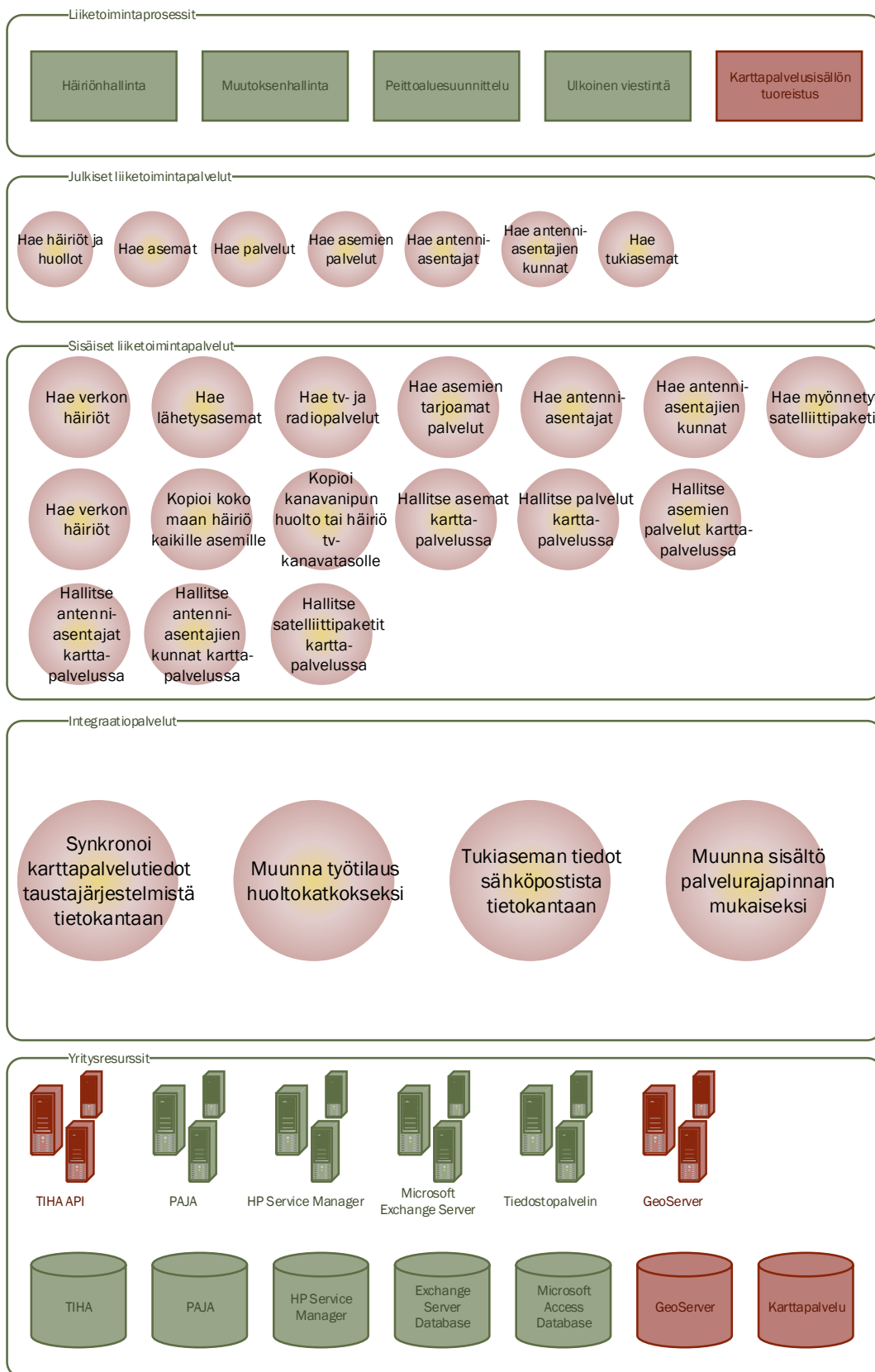
## 8.4 Ulkoiset rajapinnat

Karttapalvelusovellukselle tuotettiin HTTP/SOAP-rajapinta Digitan palveluväylän avulla. Rajapinnan avulla tiedot pysyvät karttapalvelusovelluksessa ajan tasalla. Kyselyt karttapalvelusovelluksen SOAP-asiakasohjelmasta tulevat myös ajastetusti. Karttapalvelusovelluksen kehityksen tueksi ja kumppanille toimitettu rajapintakuvaus on tämän työn liitteenä A.

Palveluväylä on kahdennettu vikasietoinen järjestelmä, jonka vuoksi HTTP-liikenne karttapalvelusovelluksesta palveluväylälle tulee kuormantasaajan kautta. Liikenne internetissä kuormantasaajalle saakka on suojattua HTTPS-liikennettä. Kuormantasaaja purkaa myös SSL-salauksen, joten Digitan sisäverkossa karttapalveluun liittyvä liikenne on suojaamatonta.

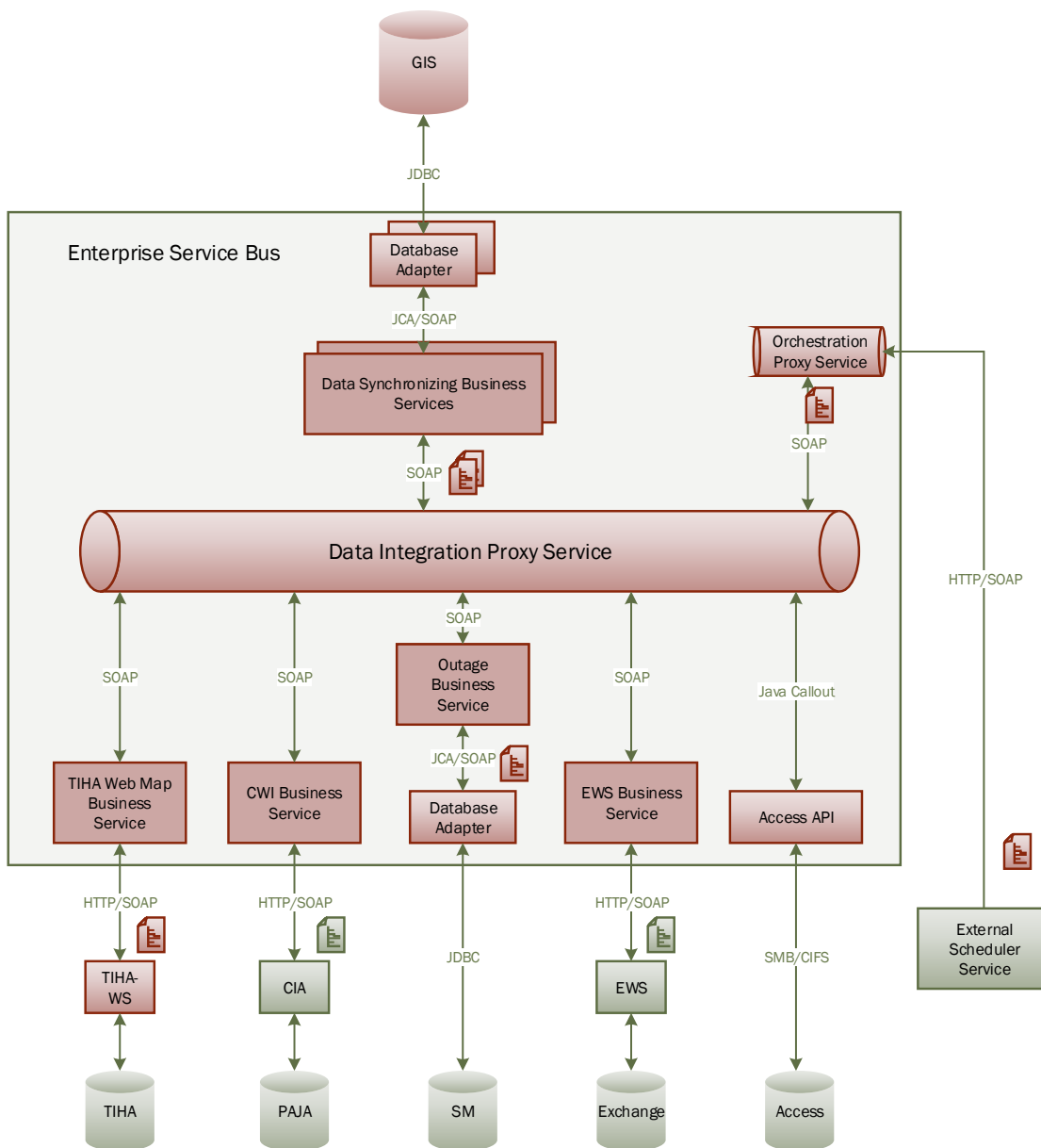
Palveluväylälle luotiin joukko tietokanta-adaptoreita, joiden kautta tietojen kysely SOAP-protokollalla on mahdollista. Integraatiopalvelu välittää palveluväylälle tulleet pyynnöt oikealle tietokanta-adapterille ja GIS-tietokantaan. Tämä toisen vaiheen integraatoratkaisu on täysin irrallinen ensimmäisen vaiheen integraatiototeutuksesta. Palveluväylä ja GIS-tietokanta ovat toki yhteisiä komponentteja kummassakin integraatiossa. Tietojen hakeminen GIS-kannasta ja niiden välittäminen karttapalvelulle kuitenkin toimii, vaikka taustajärjestelmiin ei saisikaan yhteyttä. Näissä tilanteissa ainoastaan GIS-tietokannassa oleva tieto ei päivity. Kuvassa 27 on havainnollistettu ulkoisen rajapinnan toimintaperiaate.

Karttapalvelun integraatoratkaisuun toteutettiin myös valvontaominaisuuksia mahdollisia virhetilanteita varten. Palveluväylälle lisättiin virheenkäsittelijöitä, jotka kirjoittavat virhetilanteessa virheilmoituksen palveluväylän sisäiseen raportointitietokantaan. Raportointitietokantaan voidaan lokittaa myös muitakin kuin virhetilanteita. Lokituksen lisäksi palveluväylä määrättiin lähettämään virheilmoitus sähköpostina valittuihin sähköpostiosoitteeseen. Virhe tapahtuu, kun jokin palvelu ei vastaa ollenkaan johtuen esimerkiksi tietoliikenne- tai tietokantahäiriöstä. Virhe voi tapahtua myös täysin palveluväylän sisäkkäkin esimerkiksi silloin kun XML-skeemamuutos epäonnistuu. Skaamamuutoksen epäonnistumisen voi aiheuttaa XML-dokumentti, joka ei ole sovitun XML-skeeman mukaista. Virheenkäsittely tehdään sekä tietojen koostamisessa GIS-tietokantaan, mutta myös julkisen ohjelmointirajapinnan kautta tulleissa pyynnöissä.

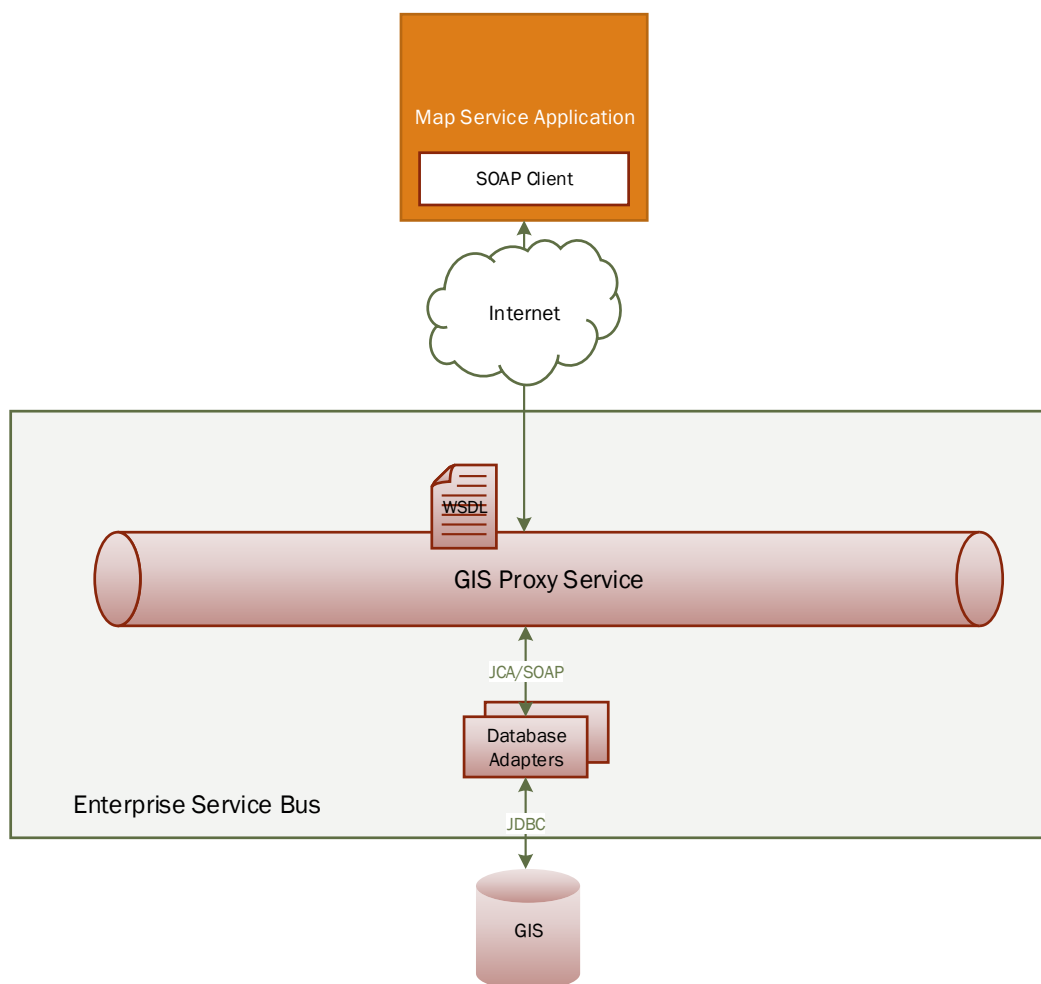


Kuva 25: Karttapalveluratkaisun SOA-mallin mukaiset elementit.





Kuva 26: Karttapalveluratkaisun integraatio: informaation koostaminen taustajärjestelmästä yhteiseen tietokantaan.



Kuva 27: Karttapalveluratkaisun integraatio: informaation tarjoaminen ulkoiselle rajapinnalle.

## 8.5 Yhteenveto karttapalveluratkaisusta

Tässä luvussa esitettiin hajautetun karttapalveluratkaisun taustajärjestelmäosuus. Tietoa piti hakea yrityksen olemassa olevista tietojärjestelmistä ja tarjota julkisen rajapinnan kautta itse karttapalvelun web-sovellukselle internetin yli. Tieto haettiin rajapintojen kautta, jotka olivat järjestelmien valmiita Web Service -rajapintoja tai tässä työssä toteutettuja rajapintapalveluita. Tässä työssä rajapinnat toteutettiin joko palveluväylän tietokanta-adapterilla, joka muuntaa tietokantakyselyn Web Service-rajapinnaksi tai muulla itse toteutetulla ohjelmalla. Web Service -rajapintojen tarjoama sisällön XML-skeema muunnettiin kohdejärjestelmän mukaiseen XML-skeemaan palveluväylällä toteutetuissa integraatiopalveluissa. Kaikki toteutetut palvelut ovat uudelleenkäytettäviä ja tulivat osaksi Digitan palvelukeskeistä arkkitehtuuria.

Kohdejärjestelmänä oli GIS-tietokanta, jonne kaikki heterogeenisista lähteistä peräisin oleva tieto koottiin homogeenisempaan muotoon. GIS-tietokantaa vasten luotiin palveluväylällä julkinen rajapinta, josta varsinainen käyttöliittymän tarjoava karttapalvelusovellus voi päivittää sisältönsä. Seuraavassa luvussa arvioidaan miten toteutettu järjestelmä vastasi sille asetettuihin vaatimuksiin.

## 9 Tulokset ja johtopäätökset

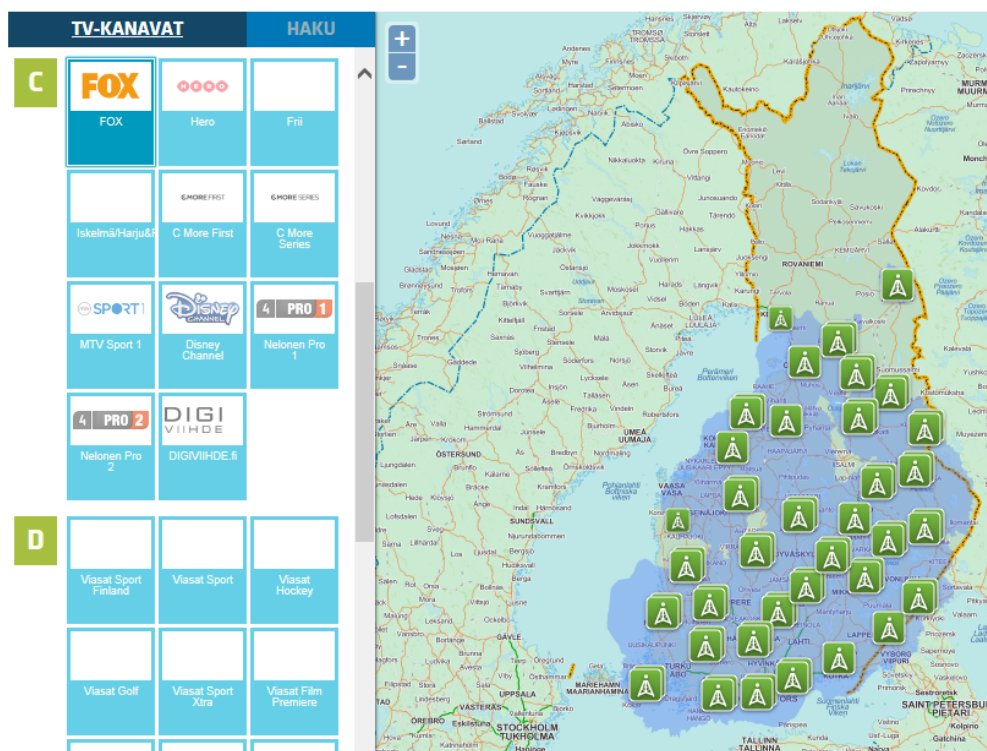
Tässä diplomityössä esiteltiin Digitan karttapalveluun liittyvä teoreettinen viitekehys, karttapalvelusovelluksen vaatimusmäärittely sekä kuvattiin karttapalvelun ratkaisu erityisesti taustajärjestelmän näkökulmasta. Ratkaisussa sovellettiin paikkatietoteknologiaa ja hajautettu järjestelmä toteutettiin palvelukeskeisen arkkitehtuurin mallin mukaisesti ja sovellettiin tyypillisiä palvelukeskeisen arkkitehtuurin teknologioita, kuten Web Services ja SOAP.

### 9.1 Vertailu vaatimusmäärittelyyn

Vaatimusmäärittelyssä haluttiin ensisijaisesti luoda karttapalvelu televisio- ja radiopalvelujen kuluttajakäyttöön. Kuluttajille tehty versio karttapalvelusta toteutettiin ja upotettiin osaksi Digitan WWW-sivuja osoitteeseen <http://www.digita.fi/karttapalvelu/>. Asetetut toiminnalliset vaatimukset toteutettiin karttapalveluratkaisuun lähes täydellisesti. Määritellyt ja toteutetut pääkäyttötapaaukset olivat seuraavat:

- Televisiokanavan näkyvyysalueen tai radiokanavan kuuluvuusalueen selvittäminen
- Television kanavanipun näkyvyysalueen selvittäminen
- Tietyissä sijainnissa tarjottavien kanavien selvittäminen
- Lähetyksessä olevan häiriön tietojen selvittäminen
- Paikkakunnalla toimivien antenniasentajayritysten hakeminen

- Television vastaanottoantennin suuntaaminen karttapalvelun avulla



Kuva 28: Kanavakohtainen peittoalue.

Ratkaisuun toteutettiin myös joitakin vaatimusmäärittelyn ulkopuolisia ominaisuuksia. Integraatiopalveluille toteutettiin kevyt sähköpostihälytyksiin perustuva valvontaratkaisu. Paikkatietojärjestelmän valvontaratkaisua olisi tulevaisuudessa syytä laajentaa osaksi Digitan muuta järjestelmä- ja palveluvalvontaa, jota nykyään tuotetaan Digitan palvelukeskuksessa ympärivuorokautisesti. Samalla olisi myös syytä pohtia vielä yksityiskohtaisemmin valvottavat kohteet. Esimerkiksi julkista rajapintaa olisi syytä valvoa suoraan ulkoa päin rajapinnan käyttäjän näkökulmasta. Tässä työssä toteutettu valvonta suoritettiin taustajärjestelmän sisältä käsin. Lisäksi valvontajärjestelmät olisivat syytä eriyttää mahdollisimman paljon itse valvottavasta järjestelmästä.

## 9.2 Julkinen ohjelmointirajapinta

Karttapalveluratkaisun käyttöliittymäsovelluksen suunnittelu ja toteutus eivät sisällyneet tämän diplomityön laajuuteen. Tässä työssä esitelty karttapalvelusovelluksen vaatimusmäärittely toimi sekä käyttöliittymäsovelluksen, että taustajärjestelmän suunnittelun ja toteutuksen pohjana. Käyttöliittymäsovelluksen toteutusprojekti kilpailutettiin usean eri alan toimijan kesken. Karttapalvelun käyttöliittymäsovelluksen toteutukseen valittiin CGI Finland Oy, joka tarjosi Navici-tuotenimellä olevaa palvelualustaa, jonka päälle Digitan karttapalvelun käyttöliittymä toteutettiin. CGI

tarjosi koko ratkaisua palveluna, joten kaikki palvelun ylläpitoon liittyvät tehtävät ja vastuut saatiin ulkoistettua. CGI sitoutui niin palvelun saatavuuden, kuin asiakaspalvelunsa vasteajoissa tiettyihin palvelutasovaatimuksiin. Tässä työssä toteutetut karttapalveluratkaisun julkiset ohjelmointirajapinnat tarjottiin nimenomaan CGI:n käyttöön. Tällöin myös vastuuraja käyttöliittymäsovelluksen ja taustajärjestelmän osalta oli rajattu hyvin selkeällä tavalla. Ohjelmointirajapinnoista oli olemassa ensimmäiset versiot CGI:n aloittaessa käyttöliittymäsovelluksen toteutuksen. Rajapinta vaati ainoastaan pientä hienosäätöä käyttöliittymän toteutuksen yhteydessä, joten rajapinnan suunnittelu perustui oikeisiin lähtökohtiin.

CGI:lle ja karttapalvelusovellukselle tarjottava ohjelmointirajapinta ei täysin piilottanut Digitan taustajärjestelmiä, vaan rajapintaoperaatiot olivat hyvin järjestelmäkohtaisia. Toisin sanoen yksi SOAP-operaatio tarjosi dataa yhden tietojärjestelmän tietokannasta, jolloin tietojen yhdistelyä integraatiokerroksessa ei suuremmin harrastettu. Toisaalta SOAP-rajapinta oli muuten täysin yhtenäinen ja rajapinnan käyttö oli identtistä riippumatta mitä tietoa haettiin. Vastuu tietojen yhdistelystä jäi itse karttapalvelusovelluksen päähän. Esimerkiksi verkon häiriötietojen kysely palautti vain tunnisteet palveluista ja asemista, joihin häiriö vaikutti. Rajapinnan kautta piti kysellä vielä erikseen aseman ja palvelun tarkemmat tiedot, jotta häiriö ja sen palvelu- ja asemavaikutukset pystyttiin esittämään karttapalvelussa. Datan yksilöivät tunnisteet olivat samanlaiset kaikkien rajapintapalveluiden välillä, joten relaatioiden rakentaminen asiakasohjelmistopäässä eri rajapintojen tietojen välillä oli yksinkertaista.

Vaihtoehtona olisi ollut valmiin rajapinnan luominen, jossa häiriön, aseman ja palvelun tiedot olisi yhdistetty jo integraatiokerroksessa, jolloin rajapinnan kuluttaja olisi yhdellä kyselyllä saanut kaiken tarvittavan tiedon häiriöön liittyen. Rajapinta kuitenkin haluttiin pitää yleisempänä ja jättää vastuu tietojen yhdistelystä rajapintaa hyödyntävälle sovellukselle. Alusta asti oli tavoitteena tehdä rajapinta, jota voitaisiin hyödyntää karttapalvelun lisäksi myös muissa sovelluksissa ja tarpeissa tuottamalla kooltaan mieluummin pieniä kuin liian suuria palveluita. Liian pitkälle jalostettujen rajapintojen uudelleenkäytön mahdollisuus olisi ollut selvästi pienempi. Rajapinnan hyödyntäminen kuitenkin vaatii toiselta osapuolelta selkeää ymmärrystä rajapinnan toiminnasta kokonaisuutena. Kattavakassakaan dokumentaatiossa ei voida ottaa kantaa mitä rajapintapalveluita on käytettävä ja yhdisteltävä kussakin tilanteessa, jotta saatavilla on riittävä määrä tietoa rajapinnan kuluttajan kannalta. Sen sijaan asianmukaisessa rajapintadokumentaatiossa voidaan osoittaa mitkä elementit ovat tiedon avainelementtejä ja mikä suhde niillä on toisiin rajapintapalveluihin.

Pitää kuitenkin huomata, että nykyiset palvelu- ja resurssikeskeiset arkkitehtuurin pyrkivät ennemminkin luomaan palveluista entistä pienikokoisempia mikropalveluita. Tällöin tässä työssä toteutetut rajapintapalvelut voidaan nähdä jopa liian monimutkaisina ja vain tiettyyn tarkoitukseen toteutettuina palveluina. Palveluiden uudelleenkäytön kannalta mahdollisimman pienet palveluyksiköt ovat perusteltuja, koska monimutkaiset palvelukokonaisuudet useimmiten toimivat vain tietyn tai ainakin rajallisten käyttötapauksien yhteydessä. REST-malli lupaa palveluiden olevan myös vahvasti itsensä kuvaavia ja keskenään linkittyviä, jolloin palveluiden käyttö taustajärjestelmää tuntematta pitäisi olla mahdollista. Tässä työssä rajapintaa teh-

tiin pääasiassa yhtä käyttötarkoitusta varten, joten palveluiden suunnittelussa ei tarvinnut huomioida palveluille asetettuja tarpeita niin laajasti, kuin täysin avoimissa ympäristöissä. Mikäli rajapinnan käyttö laajenee muihin käyttötarkoituksiin, niin REST-arkkitehtuuriin perustuvaa rajapintaa kannattaa harkita nykyisen rajapinnan rinnalle tai kokonaan korvaajaksi.

Peittoalueiden ja parhaan lähetysaseman osoittava aineisto haettiin tietokannasta GeoServer-ohjelmistolla, joka myös tarjosi standardin WFS-rajapinnan CGI:lle ja karttapalvelusovellukselle. Kun taas asemat, LTE-tukiasemat ja satelliittipaketit olivat tarjolla SOAP-palveluiden kautta palveluväylän tarjoamana rajapintana, niin paikkatieto jakaantui teknologiamielessä kahdelle erilaiselle alustalle. Tämä ero näkyi selvästi rajapintaa hyödyntävälle taholle, jossa tieto piti hakea sekä SOAP-että WFS-rajapintojen kautta. Myös näitä tietoja rajapintapalvelun kuluttajan piti pystyä yhdistelemään. Esimerkiksi aseman peittoalue WFS-rajapinnalla piti sisällään vain tiedon aseman tunnuksesta, mutta tätä tietoa piti pystyä rikastamaan SOAP-operaation avulla hakemalla aseman tarkemmat tiedot. Laajemmassa käytössä rajapinta tällaisenaan voi olla liian monimutkainen. Toisaalta SOAP ja WFS ovat protokollina erilaisia ja niiden käyttötarkoitukset ovat erilaisia. Se ei silti sulje pois sitä, etteikö aseman tietoja voisi esittää järkevästi WFS-protokollan mukaisena pistegeometriana. Mikäli rajapintapalveluiden käyttö laajenee, on hyvä pohtia voisiko rajapintaa kääriä yhtenevämmäksi kokonaisuudeksi tai ainakin siirtää selkeästi paikkatietomaista dataa GeoServer-ohjelmiston tarjoamaksi WFS-palveluksi. Vaatisi kuitenkin tarkempaa perehtymistä asiaan kannattaisiko rajapintaa kehittää enemmän SOAP:n, WFS:n tai kokonaan jonkin muun protokollan suuntaan.

Työssä valitut protokollat SOAP ja WFS voivat olla liian raskastekoisia reaaliaikaisissa sovelluksissa. Tässä ratkaisussa reaaliaikaisuudella ei ollut suurta merkitystä, koska tietoa siirrettiin pääasiassa ajastetusti. Toisin sanoen, tiedot siirretään karttapalvelusovelluksen käyttöön jo ennen kuin käyttäjä tietoa kysellee. Reaaliaikaista tiedon hakua Digitan rajapinnasta tai sanoman käsittelyä ei tehdä silloin kun palvelun loppukäyttäjä käyttää järjestelmää. Protokollien valintaa olisi pitänyt harkita eritavalla, mikäli vaatimus reaaliaikaisuuteen olisi ollut toinen. Nämä rajoitukset ovat hyvä huomioida silloin, jos ohjelmointirajapintaa tullaan myöhemmin käyttämään reaaliaikaisen sovelluksen yhteydessä. Yksi vaihtoehto HTTP-pohjaisille protokollille voisi olla WebSocket.

Yhtenä vaihtoehtona on myös tuoda rajapintaan välimuisti, koska rajapintaan tulevat kyselyt palauttavat usein saman tietojoukon. Välimuistin avulla tietoa ei kyseltäisiin joka kerta GIS-tietokannasta saakka, vaan nopeammasta välimuistista.

### 9.3 Paikkatiedon käsittely

Taustajärjestelmässä käsitellään paikkatietoa usealla eri tavalla. Lähetysasemat ja LTE-tukiasemat ovat pistemäisiä geometrioita eli ne voidaan sijoittaa koordinaattien avulla tiettyyn pisteeseen kartalla. Palveluiden peittoalueet taas ovat monikulmioita, jotka myös sijoittuvat kartalle niissä määriteltyjen koordinaattien perusteella. Parhaan aseman valinnan mahdollistava aineisto on myös paikkatietoa, joka kertoo televisiosignaalin voimakkuuden neliömäisillä alueilla koko Suomessa. Televisionkat-

sojille myönnettyt satelliittipaketit sisältävät paikkatietoa katuosoitteen muodossa.

Lähetysasemia ja LTE-tukiasemia on kuitenkin taustajärjestelmässä käsitelty perinteisenä XML-sisältönä, eikä teknisessä mielessä paikkatieto-objekteina. Toisin sanoen, asemat ovat olleet XML-dokumentteja, joissa pituus- ja leveyskoordinaatit on ilmaistu XML-elementeissä numeroarvoina. Tässä muodossa tieto ei perustu mihinkään standardiin XML-sanastoon, jolloin paikkatiedon siirrettävyys ja yhteensopivuus muiden paikkatietojärjestelmien kanssa ei ole taattua. Tämän lisäksi XML-dokumentti ei pidä tietoa koordinaattijärjestelmästä, jolloin tiedon hyödyntäminen vaatii vielä erillisen rajapintadokumentaation, koska rajapinta tai sen tekninen WSDL-kuvaus ei kerro mitään koordinaattijärjestelmistä. Tässä tapauksessa tieto muunnettiin paikkatiedoksi käyttöliittymäsovelluksen päässä. Koordinaattijärjestelmät olivat kuvattu CGI:lla toimitetussa dokumentaatiossa, joten käytännön haasteisiin ei tässä asiassa lopulta törmätty.

Haasteita sen sijaan liittyi LTE-tukiasemiin, joiden tiedot siirrettiin Digitalle erilaisissa koordinaatistojärjestelmissä riippuen kenen mobiilioperaattorin tietojärjestelmästä paikkatieto oli lähtöisin. Digitan julkisessa ohjelmointirajapinnassa oli sovittu WGS 84 -koordinaatiston käytöstä, joten kaikkien LTE-tukiasemien tiedot jouduttiin muuttamaan tähän koordinaattijärjestelmään. Koordinaattimuunnoksia varten toteutettiin Java-ohjelmakirjasto, jota hyödynnettiin integraatiokerroksessa palveluväylällä koordinaattien muunnoksessa. Tällä tavalla karttapalveluratkaisun tietokantaan saatiin kerättyä kaikki paikkatieto yhdenmukaisessa koordinaattijärjestelmässä. Koordinaattimuunnoksia olisi ollut mahdollista tehdä myös Oracle-tietokannassa olevaan dataan tietokannan tarjoamalla prosedureilla, mutta ainakin suomalaisten koordinaattien osalta muunnosten lopputulokset vaikuttivat liian epätarkoilta. Jos LTE-tukiasemat olisivat olleet paikkatieto-objekteja, niin paikkatietojärjestelmät olisivat sallineet tietuekohtaiset koordinaattimääritykset, eikä koordinaatistomuunnoksia olisi välttämättä tarvittu. Toisaalta myös löytyy paljon erilaisia sovelluksia paikkatiedon muuntamiseen koordinaattijärjestelmästä toiseen. Esimerkiksi taustajärjestelmäratkaisuun sisältyvä GeoServer-ohjelmisto muuntaa standardimuotoisen paikkatiedon koordinaatit järjestelmästä toiseen ajon aikaisesti. GeoServerin koordinaattimuunnosten tarkkuutta ei kuitenkaan tämän työn yhteydessä selvitetty.

Koordinaattijärjestelmien osalta ei tehty niin tarkkaa suunnitelmaa kuin olisi pitänyt. Paikkatietokohteet olivat erilaisissa koordinaattijärjestelmissä. Tämä lienee aivan normaalitilanne paikkatietoprojekteissa. Taustajärjestelmässä tehtiin koordinaattimuunnoksia, jotka paikoin olivat haasteellisia. Lopputuloksena ohjelmointirajapinnan kautta tarjottavat paikkatietokohteet olivat suomalaisessa yhtenäiskoordinaatistossa (YKJ). Yhtenäiskoordinaatiston valinta ei ollut CGI:lle ongelma, mutta muitakin vaihtoehtoja olisi ollut. Esimerkiksi Google Mapsin käyttämä WGS 84 voisi olla paremmin yhteensopiva useimpien paikkatietojärjestelmien kanssa. Jos rajapinta olisi tarjonnut paikkatietoa jossain muussa kuin yhtenäiskoordinaatistossa, niin järjestelmän tekemät koordinaattimuunnosten lukumäärä olisi jäänyt vähäisemmäksi. Koordinaattimuunnokset ovat aina laskutoimituksena työläitä ja vaikuttavat sovelusten suorituskykyyn.

Satelliittipaketit eivät sisältäneet koordinaatteja kuluttajien sijainneista, vaan ainoastaan katuosoitteen, postinumeron ja postitoimipaikan. Tiedon siis haluttiin näy-

tettävän karttapalvelusovelluksessa katuosoitteen mukaisessa sijainnissa. Tällaisissa tapauksissa katuosoite on tavalla tai toisella muutettava maantieteelliseksi koordinaatiksi. Tällaista toimenpidettä kutsutaan geokoodaukseksi ja geokoodaus tehdään tyypillisesti siihen tarkoitettuun palveluun. Geokoodausta ei tehty karttapalveluratkaisun taustajärjestelmässä, vaan osoitetiedot välitettiin sellaisenaan rajapinnan kautta käyttöliittymäsovellukseen. Geokoodaus tapahtui siis lopulta käyttöliittymäsovelluksen päässä. Rajapinnan monikäyttöisyyttä lisäisi jos satelliittipaketin tiedot sisältyvä dokumentti sisältäisi tiedon koordinaattijärjestelmästä ja koordinaateista. Edelleen, tiedon voisi tällöin myös muuntaa esimerkiksi GML-muotoiseksi paikkatiedoksi ja trajota WFS-rajapinnan kautta.

## 9.4 Arkkitehtuuri

Vaatimusmäärittelyn valmistumisen jälkeen saatiin tieto kaikesta siitä datasta, jota karttapalvelusovelluksessa tulotaisiin tarvitsemaan. Sen pohjalta oli helppo määrittellä järjestelmät ja tietolähteet, joissa tarvittava data oli saatavilla. Tässä yhteydessä myös tunnistettiin mitä dataa ei ollut lainkaan saatavilla. Tästä aiheutui pieneköjä muutoksia käytössä oleviin tietojärjestelmiin ja liiketoimintaprosesseihin.

Karttapalveluratkaisun kantavaksi arkkitehtuuriratkaisuksi valittiin palvelukeskeinen arkkitehtuuri. Digitan integraatoratkaisuissa oli jo aiemmin sovellettu palvelukeskeistä arkkitehtuuria, joten karttapalveluakin tukevia palveluja oli jo toteutettu tai käytössä olevat tietojärjestelmät tarjosivat niitä rajapintojensa kautta. Tällä tavalla päästiin heti kiinni yhteen palvelukeskeisen arkkitehtuurin hyötyihin eli palveluiden uudelleenkäytettävyyteen. Kirjallisuustutkimuksen perusteella palvelukeskeinen arkkitehtuuri nähtiin myös erittäin sopivaksi yrityksen paikkatietojärjestelmissä.

Kun tiedettiin mitä tietoa karttapalvelussa tulotaisiin tarvitsemaan, selvitettiin mitä valmiita tiedonhakupalveluita Digitan SOA-ympäristö pystyi tarjoamaan. Samalla myös pystyttiin tunnistamaan puuttuvat palvelut. Näiden pohjalta päästiin kuvaamaan järjestelmän haluttua ohjelmistoarkkitehtuuria, joka osoittautuikin tehokkaaksi työvälineeksi. Ohjelmistoarkkitehtuuri sisälsi olemassa olevia palveluita ja komponentteja, mutta se osoitti myös hyvin puuttuvat rajapinnat, liiketoimintapalvelut ja integraatiopalvelut. Arkkitehtuurisuunnitelman pohjalta syntyi selkeä määrittely toteutettavista liiketoiminta- ja integraatiopalveluista.

Karttapalveluratkaisu hyödynsi operatiivisia järjestelmiä, mutta karttapalvelu ei voinut aiheuttaa merkittävästi nykyistä suurempaa kuormitusta järjestelmille. Karttapalvelun taustajärjestelmän ohjelmointirajapintaan tulevia kutsuja ei haluttu välittää taustajärjestelmien rajapintoihin tai niiden tietokantoihin. Tämän vuoksi haluttiin luoda kokonaan erillinen rajapinta ja tietokanta, johon karttapalvelusovelluksen ja mahdollisesti myöhemmin muiden sovellusten kutsut kohdistuisivat. Tämän vuoksi arkkitehtuuri jakautua sisäisiin rajapintoihin ja tietojärjestelmiin sekä julkiseen rajapintaan ja sen tietokantaan. Palvelukeskeinen arkkitehtuuri ja sen palvelut rakennettiin tämän periaatteen mukaisesti. Toisaalta tarvittiin palveluita tietojen hakemiseen taustajärjestelmästä, tietojen viemiseen karttapalvelun tietokantaan ja tietojen hakemiseen karttapalvelun tietokannasta. Karttapalvelun taustajärjestelmää voi pitää myös erittäin vikasietoisena. Operatiivisten järjestelmien huolto- ja muut



katkot eivät vaikuta karttapalvelun toimintaan muuten kuin, estämällä kyseisen järjestelmän tietojen päivittymisen karttapalvelun tietokantaan ja sitä kautta itse käyttöliittymäsovellukseen. Käyttöliittymäsovellus on myös toteutettu siten, että rajapinnan ollessa saavuttamattomissa, sovellus on edelleen toiminnassa, mutta tietosisältö ei vain päivity. Karttapalvelussa on paljon hitaasti muuttuvaa tietoa, jolloin sovelluksella on arvoa, vaikka tietosisältö ei olisi täysin ajan tasalla. Karttapalvelun GIS-tietokantaa voi siis pitää hyvin perusteltuna ratkaisuna. Haittapuolena voi pitää sitä, että tiedon siirto taustajärjestelmästä GIS-tietokantaan aiheuttaa viivettä. Samalla tavalla tietojen siirto GIS-tietokannasta karttapalvelusovellukselle tapahtuu ajastetusti, jolloin karttapalvelusovelluksessa näytettävä tieto on vanhempaa kuin operatiivisissa järjestelmissä.

Integraatioratkaisun ja palvelukeskeisen arkkitehtuurin ytimenä on palveluväylä Oracle Service Bus. Palveluväylän avulla voitiin toteuttaa useimmat liiketoiminta- ja integraatiopalvelut käytännössä ilman ohjelmakoodin kirjoittamista. Palveluväylä ei ole välttämätön osa palvelukeskeistä arkkitehtuuria, vaan palvelut voidaan toteuttaa myös täysin muiden teknologioiden pohjalle. Toisaalta palveluväylä on tarkoitettu nimenomaan palvelukeskeisten ratkaisujen kovaksi ytimeksi, jollaisena se tässäkin ratkaisuna toimi. Palveluväylän etuna oli palveluiden tehokas toteuttaminen. Palveluväylällä on myös kyky kytkeytyä hyvin erilaisiin rajapintoihin. Palveluväylä ei kuitenkaan ole paikkatietojärjestelmä, vaikkakin mahdollistaisi GML:n välittämisen siinä missä muidenkin eri XML-formaattien. Palveluväylän rinnalle pystytettiin GeoServer, joka kytkeytyi relaatiotietokantaan ja muunsi tietokannassa olleen paikkatietoaineiston WFS-palveluksi. Palveluväylän ja GeoServerin rooli oli siis ratkaisussa hyvin samanlainen sillä erolla, että GeoServer käsitteli standardia paikkatietoa, kun taas palveluväylän läpi välitettiin kaikki muu data. Palveluväylä on vikasietoinen kahdentettu klusteriratkaisu, kun taas GeoServer on yksittäinen instanssi. GeoServerin kahdentaminen muiden taustajärjestelmäkomponenttien tapaan lisäisi järjestelmän vikasietoisuutta entisestään.

Palveluväylä ei täyttänyt kuitenkaan kaikki vaatimuksia mitä liiketoiminta- ja integraatiopalveluille asetettiin. Kartalle tuotavaa tietoa oli Microsoft Access-tietokannassa, johon palveluväylä ei pystynyt kytkeytymään. Palveluväylän graafinen tapa toteuttaa integraatiota ei ollut soveltuva alusta runsaasti matemaattisia laskutoimituksia vaativien koordinaatistomuunnosten tekemiseen. Microsoft Access-tietokannan rajapintaa ja koordinaatistomuunnoksia varten piti toteuttaa erilliset palvelut, joita palveluväylä pystyi kutsumaan. Tarvittavaa tietoa oli myös postipalvelimella, postilaatikon sisällä yksittäisissä sähköpostiviesteissä. Palveluväylä ei kykene hakemaan sähköposteja postipalvelimelta, mutta postipalvelimena oleva Microsoft Exchange tarjoaa myös Exchange Web Service SOAP-rajapintaa. Tällöin sähköpostijärjestelmä saatiin näyttämään rajapinnaltaan samanlaiselta järjestelmältä muiden operatiivisten tietojärjestelmien kanssa.

Lisäksi TIHA-järjestelmää varten luotiin myös standardit SOAP-rajapinnat erilliselle Java-sovelluspalvelimelle. Palvelut olisi ollut mahdollista toteuttaa myös palveluväylälle, mutta TIHA-järjestelmä ylläpito- ja kehitysvastuu oli eri organisaatioyksiköllä kuin muiden operatiivisten järjestelmien, jolloin teknologiavalinta tehtiin karttapalveluratkaisusta irrallisena. Standardien Web Services -teknologioiden vuoksi

mitään yhteensopivuusongelmia ei tullut, vaan erilaiset sovelluspalvelimet ja kehitystyökalut piiloutuivat täysin rajapintojen taakse.

Kaikki dokumentit eri palveluiden välillä liikkuvat XML-muodossa SOAP-protokollalla. SOAP valittiin protokollaksi, koska useimmat olemassa olevat rajapinnat olivat nimenaan SOAP-rajapintoja. Lisäksi SOAP on palveluväylällä paremmin tuettu integraatiotapa kuin esimerkiksi RESTful, joka on saavuttanut suosiotaan viime vuosien aikana. RESTful olisi ollut toisaalta sopiva tapa toteuttaa taustajärjestelmän julkinen rajapinta. Monet sovellukset tuntuvat nykyään suosivan dataa JavaScript Object Notation (JSON) -muodossa RESTful-rajapinnan kautta. Tässä työssä ei syvennytty SOAP:in ja RESTfulin eroihin, mutta RESTfulin käyttämä JSON on siirtomuotona SOAP:ia tehokkaampi, jossa XML-elementit muodostavat suuren osan siirrettävästä datasta. Karttapalveluratkaisussa XML-dokumentit ovat poikkeuksetta kohtalaisen suuria, joten RESTfulilla olisi voitu saavuttaa etuja ainakin siirrettävän datan määrässä ja siten sisällön käsittelyssä rajapintaa hyödyntävissä sovelluksissa. Pitää kuitenkin muistaa, että rajapinnan suorituskyky ei millään tavalla ole ratkaisussa pullonkaula, koska dataa haetaan tiheimmilläänkin harvemmin kuin kerran minuutissa. Rajapinnan käytön laajentuessa on hyvä huomata, että sen suorituskyky voi tulla jatkossa suuremmaksi haasteeksi. Jatkokehityksen yhteydessä rajapinnan suorituskyky on syytä mitata ja arvioida missä määrin se tukee tulevia vaatimuksia.

Arkkitehtuuria voidaan pitää lopulta onnistuneena, koska sen pohjalta voitiin toteuttaa toimiva taustajärjestelmäratkaisu rajapintoihin. Taustajärjestelmä taas tarjosi kaikki tarvittavat palvelut vaatimusmäärittelyn mukaisen karttapalveluratkaisun toteuttamiseksi. Suorituskykyvaatimusten osalta päästiin myös tavoitteeseen, sillä karttapalvelusovelluksen tekemillä pyynnöillä julkiseen rajapintaan ei ole vaikutuksia Digitan operatiivisiin tietojärjestelmiin. Tietoa tuodaan taustajärjestelmästä ajastetusti karttapalvelun tietokantaan. Tiedot on luokiteltu kiireellisiksi ja ei-kiireellisiksi. Kiireellisten tietojen päivityssykli on 5 minuuttia ja ei-kiireellisten tietojen päivityssykli on 3 tuntia.

## 10 Yhteenveto

Tämän työn tarkoituksena oli suunnitella ja toteuttaa tietoliikenneyritys Digitalle karttapalvelusovellus. Karttapalvelusovelluksen oli määrä olla web-sovellus, johon tarjottaisiin pääsy kuluttajille Digitan WWW-sivuston kautta. Tämän lisäksi haluttiin toteuttaa hieman erilainen sovellus Digitan sisäiseen käyttöön asiakaspalvelutyötä tukemaan. Varsinainen käyttöliittymäsovelluksen toteutus ei kuulunut tähän työhön, vaan tässä työssä keskityttiin ensisijaisesti toimivan taustajärjestelmän ja sen rajapintojen rakentamiseen.

Sovelluksen suunnittelu aloitettiin vaatimusmäärittelyllä, jossa pyrittiin tunnistamaan karttapalvelusovelluksen yleisimmät käyttötapaukset kuluttajien, asiakkaiden ja sisäisten käyttäjien keskuudessa. Vaatimusmäärittelyssä asetettiin myös tiettyjä ei-toiminnallisia vaatimuksia käytettävistä teknologioista. Tunnistetut käyttötapaukset liittyivät vahvasti televisio- ja radiopalveluiden maantieteelliseen luonteeseen. Sovelluksella haluttiin selvittää, mitä televisio- tai radiokanavia on vastaanotettava missäkin. Tähän liittyen haluttiin tarjota myös työkalua antenniasennuksia tekeville henkilöille vastaanottoantennin tarkempaa suuntaamista varten. Käyttäjien lisäksi myös viranomaiset vaativat karttaesityksenä tuotettavaa häiriötiedotusta. Sovelluksen haluttiin kertovan mahdollisimman reaaliaikaisesti verkossa olevat häiriöt ja niiden vaikutus televisio- ja radiopalveluihin. Näiden lisäksi haluttiin tiedottaa tulevista ja meneillään olevista huolloista, joilla myös voi olla häiriöiden kaltaisia palveluvaikutuksia.

Vaatimusmäärittelystä jo kävi ilmi, että karttapalveluratkaisun pitäisi olla hajautettu järjestelmä, jonka osaksi liitettäisiin useita Digitan olemassa olevia tietojärjestelmiä. Näiden rooli oli toimia ensisijaisesti tietolähteinä sovellukselle. Vaatimusmäärittelyssä tuotiin esille, että kaikkia rajapintoja pitäisi kutsua HTTP-protokollalla, joten integraatoratkaisua ohjattiin tällä voimakkaasti nykyaikaisten web-tekniologioiden suuntaan.

Karttapalveluratkaisu on myös paikkatietojärjestelmä. Paikkatieto voi olla rasterimuotoista bittikarttakuvaa tai vektorimuotoista dataa, joissa kohteet ilmaistaan joukkoina koordinaattipisteitä. Tässä työssä paikkatietoa edustivat vektorimuotoiset palveluiden peittoalueet sekä lähetysasemien kentänvoimakkuuksia esittävä aineisto. OGC on standardoinut paljon paikkatietoteknologiaa edistääkseen paikkatietojärjestelmien yhteensopivuutta. Tässä työssä paikkatieto muunnettiin lopulta GML-muotoon, joka on eräs XML-kieleen pohjautuva standardi sanasto paikkatiedon käsittelyyn. GML-muotoista paikkatietoa pystyi taas välittämään standardin WFS-rajapinnan kautta karttapalvelusovellukselle.

Olemassa olevat järjestelmät tarjosivat Web Services -rajapintoja, joita pyrittiin tässä työssä hyödyntämään mahdollisimman paljon. Lisäksi oli järjestelmiä ja tietoja, joihin ei standardin mukaista rajapintaa ollut tarjolla. Näitä varten tässä työssä toteutettiin tarvittavat Web Service -rajapinnat. Web Service -rajapintoja hyödyntäen toteutettiin järjestelmien integraatioita, joihin liittyi sanomamuotoisen liikenteen XML-skeemamuunnoksia ja sanomien reitittämistä. Suurin osa Web Service- ja integraatiopalveluista toteutettiin palveluväylällä, joka on palvelukeskeinen väliohjelmisto. Palveluväylä tukee palveluiden tuottamista ja käyttämistä

palvelukeskeisen arkkitehtuurin periaatteiden mukaisesti. Palveluväylällä tehtävät XML-skeemamuunnokset toteutettiin W3C:n XPath- ja XQuery-kielillä. Kaikki karttapalvelusovelluksen sisältö käsiteltiin siis XML-muotoisena ja välitettiin Web Service-rajapintojen välillä SOAP-protokollalla.

Lopputuloksena saatiin vaatimusmäärittelyn mukainen hajautettu järjestelmä. Palvelukeskeisen arkkitehtuurin ja palveluväylän käyttö loi myös paljon uudelleen käytettäviä palveluita ja rajapintoja, joiden hyödyntäminen tulevaisuuden sovelluskehityksessä on mahdollista. Työn aikana myös havaittiin arkkitehtuuriin liittyviä kehityskohteita. Vaikka taustajärjestelmä palvelee nykyistä karttapalvelusovellusta hyvin, voi palveluiden uudelleen käytössä tulla haasteita, mikäli tulevaisuuden sovellukset vaativat enemmän reaaliaikaisuutta tai kuormittavat taustajärjestelmän julkista rajapintaa enemmän. Näitä haasteita voi yrittää ratkoa käyttämällä kevyempiä protokollia, tietomuotoja sekä yhdenmukaistamalla koordinaattijärjestelmien käyttöä.

## Viitteet

- [1] Viestintävirasto, “Televisio & radio.” Saatavilla <https://www.viestintavirasto.fi/tvradio.html>. Viitattu 17.3.2016.
- [2] Liikenne- ja viestintäministeriö, “Verkkotoimiluvat.” Saatavilla <http://lvm.navigo.fi/verkkotoimiluvat>. Viitattu 17.3.2016.
- [3] J. C. Whitaker, ed., *Standard Handbook of Broadcast Engineering*. McGraw-Hill, 2005.
- [4] L. Lehto, *Real-Time Content Transformations in a Web Service-Based Delivery Architecture for Geographic Information*. PhD thesis, Helsinki University of Technology (Espoo, Finland), 2007.
- [5] P. S. Talegaonkar, “Service oriented architecture for gis applications,” in *The 12th International Conference of International Association for Computer Methods and Advances in Geomechanics (IACMAG)*, pp. 1980–1987, Dept. of Computer Engineering, D Y Patil COE, Pune, India, lokakuu 2008.
- [6] X. Wang, X. Pang, and Y. Luo, “The design and implementation of gis applications based on soa,” in *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*, pp. 3999–4002, IEEE, 2010.
- [7] S. Wei, G. Joos, and W. Reinhardt, “Management of spatial features with gml,” in *Proceedings of the 4th AGILE Conference on Geographic Information Science, Brno, Czech Republic*, pp. 370–375, Association of Geographic Information Laboratories, 2001.
- [8] L. Díaz, C. Granell, and M. Gould, “Spatial data integration over the web,” *Handbook of Research on Innovations in Database Technologies and Applications: Current and Future Trends*, pp. 325–333, 2009. IGI Global.
- [9] Z.-R. Peng, “A proposed framework for feature-level geospatial data sharing: A case study for transportation network data,” *International Journal of Geographical Information Science*, vol. 19, no. 4, pp. 459–481, 2005. Taylor & Francis.
- [10] L. Lehto, “Interoperable processing of gi on the web,” *Surveying Science in Finland*, vol. 16, pp. 3–18, 1998. The Finnish Society of Surveying Sciences.
- [11] J. Ponce-Rojas, S. Vidal-Beltrán, M. A. Acevedo-Mosqueda, and M. Jimenez-Licea, “A geographic information system applied to coverage maps of 3g cellular communications networks,” *Journal of Geographic Information System*, vol. 3, pp. 140–144, huhtikuu 2011. Scientific Research Publishing, Inc.
- [12] L. Yubao, L. Longchang, and L. Fei, “Service-oriented data integration using gis: An anchorage management tool for wenzhou government, china,” in *Software Engineering and Knowledge Engineering: Theory and Practice* (Y. Wu, ed.),

- vol. 2 of *Advances in Intelligent and Soft Computing*, pp. 657–662, Springer Berlin Heidelberg, 2012.
- [13] M. Treiblmayr, S. Scheider, A. Krüger, and M. von der Linden, “Integrating gi with non-gi services – showcasing interoperability in a heterogeneous service-oriented architecture,” *GeoInformatica*, vol. 16, no. 1, pp. 207–220, 2012. Springer.
- [14] L. Godin, *GIS in Telecommunications*. ESRI, 2001.
- [15] C. Fry, *Geographical Information Systems*, ch. GIS in Telecommunications, pp. 819–826. Wiley, New York, 2 ed., 2005.
- [16] J. Cowan and R. Tobin, “Xml information set (second edition),” tech. rep., World Wide Web Consortium, helmikuu 2004. Saatavilla <https://www.w3.org/TR/2004/REC-xml-infoset-20040204/>. Viitattu 19.3.2016.
- [17] J. Casademont, E. Lopez-Aguilera, J. Paradells, A. Rojas, A. Calveras, F. Barcelo, and J. Cotrina, “Wireless technology applied to gis,” *Computers & Geosciences*, vol. 30, pp. 671–682, heinäkuu 2004. Pergamon-Elsevier Science Ltd.
- [18] International Organization for Standardization, “Geographic information – geography markup language (gml),” ISO 19136:2007, International Organization for Standardization, Geneva, Switzerland, 2007.
- [19] C.-H. Huang, T.-R. Chuang, D.-P. Deng, and H.-M. Lee, “Building gml-native web-based geographic information systems,” *Computers & Geosciences*, vol. 35, no. 9, pp. 802–1816, 2009. Elsevier Ltd.
- [20] M. Rosen, B. Lublinsky, K. T. Smith, and M. J. Balcer, *Applied SOA: Service-Oriented Architecture and Design Strategies*. John Wiley & Sons, 2012.
- [21] P. Kruchten, *The Rational Unified Process: An Introduction*. Addison-Wesley Professional, 2004.
- [22] L. Guo, J. Gong, J. Sun, and X. Wei, “Study on gis architecture based on soa and ria,” in *International Conference on Information Sciences and Interaction Sciences (ICIS), 2010 3rd*, pp. 620–625, IEEE, 2010.
- [23] I. Melzer, *Service-orientierte Architekturen mit Web Services: Konzepte-Standards-Praxis*. Springer-Verlag, 2010.
- [24] K. Champion, “Soa: The new architecture that leverages the old.” Saatavilla <http://www.developers.net/node/view/106>, 2004. Viitattu 31.3.2016.
- [25] Y. V. Natis, “Service-oriented architecture scenario.” Saatavilla <https://www.gartner.com/doc/391595>, huhtikuu 2003. Viitattu 24.10.2015.
- [26] ESRI Inc., “Geospatial service-oriented architecture (soa), an esri (r) white paper.” Saatavilla <http://www.esri.com/library/whitepapers/pdfs/geospatial-soa.pdf>, kesäkuu 2007. Redlands: ESRI Inc.

- [27] D. Gordon, G. Geiger, N. Lowe, and J. Jickling, “What is an electronic patient record?,” in *Proceedings of the AMIA Symposium*, pp. 240–244, Sunnybrook Health Science Centre, 1998.
- [28] IBM developerWorks, “New to soa and web services.” Saatavilla <http://www.ibm.com/developerworks/webservices/newto/>, 2009. Viitattu 24.10.2015.
- [29] C. M. MacKenzie, K. Laskey, F. McCabe, P. F. Brown, and R. Metz, “Reference model for service oriented architecture 1.0, oasis standard.” Saatavilla <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>, lokakuu 2006. Viitattu 27.9.2015.
- [30] X. Liu, Y. Hui, W. Sun, and H. Liang, “Towards service composition based on mashup,” in *Services, 2007 IEEE Congress on*, pp. 332–339, IEEE, 2007.
- [31] Y. Bishr, “Overcoming the semantic and other barriers to gis interoperability,” *International journal of geographical information science*, vol. 12, no. 4, pp. 299–314, 1998. Taylor & Francis.
- [32] S. Scheider, “The case for grounding databases,” in *GeoSpatial Semantics*, pp. 44–62, Springer, 2009.
- [33] M. F. Worboys and M. Duckham, *GIS: A Computing Perspective*. CRC press, 2004.
- [34] P. Schut, “Opengis® web processing service.” Saatavilla [http://portal.opengeospatial.org/files/?artifact\\_id=24151](http://portal.opengeospatial.org/files/?artifact_id=24151), kesäkuu 2007. Viitattu 4.10.2015.
- [35] C. Pautasso, E. Wilde, and R. Alarcon, *REST: Advanced Research Topics And Practical Applications*. Springer, 2014.
- [36] E. Wilde and C. Pautasso, *REST: From Research to Practice*. Springer Science & Business Media, 2011.
- [37] D. Merrill, “Mashups: The new breed of web app.” Saatavilla <http://www.ibm.com/developerworks/xml/library/x-mashups/index.html>, elokuu 2006. Viitattu 31.3.2016.
- [38] S. S. Gao, C. Sperberg-McQueen, and H. S. Thompson, “W3c xml schema definition language (xsd) 1.1 part 1: Structures,” tech. rep., World Wide Web Consortium, huhtikuu 2012. Saatavilla <https://www.w3.org/TR/2012/REC-xmlschema11-1-20120405/>. Viitattu 19.3.2016.
- [39] J. Roy and A. Ramanujan, “Xml schema language: Taking xml to the next level,” *IT Professional*, vol. 3, pp. 37–40, maaliskuu 2001. IEEE.
- [40] C.-S. Ran and M.-C. Wang, “An xml schema-based data integration,” in *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, vol. 7, pp. 100–102, IEEE, 2010.

- [41] J. Robie, M. Dyck, and J. Spiegel, “Xml path language (xpath) 3.1,” tech. rep., World Wide Web Consortium, joulukuu 2015. Saatavilla <https://www.w3.org/TR/2015/CR-xpath-31-20151217/>. Viitattu 18.3.2016.
- [42] J. Robie, M. Dyck, and J. Spiegel, “Xquery 3.1: An xml query language,” tech. rep., World Wide Web Consortium, joulukuu 2015. Saatavilla <https://www.w3.org/TR/2015/CR-xquery-31-20151217/>. Viitattu 18.3.2016.
- [43] J. Robie, “Xml processing and data integration with xquery,” *IEEE Internet Computing*, vol. 11, pp. 62–67, helmikuu 2007. IEEE Computer Society.
- [44] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, *Web Services: Concepts, Architectures and Applications*. Springer-Verlag, 2004.
- [45] D. Liu and R. Deters, “Management of service-oriented systems,” *Service Oriented Computing and Applications*, vol. 2, no. 2-3, pp. 51–64, 2008. Springer U K.
- [46] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana, “Unraveling the web services web: an introduction to soap, wsdl, and uddi,” *IEEE Internet Computing*, vol. 6, no. 2, p. 86, 2002. IEEE Computer Society.
- [47] M. Gudgin, M. Hadley, N. Mendelsohn, J.-J. Moreau, H. F. Nielsen, A. Karmarkar, and Y. Lafon, “Soap version 1.2 part 1: Messaging framework (second edition),” tech. rep., W3C, huhtikuu 2007. Saatavilla <https://www.w3.org/TR/2007/REC-soap12-part1-20070427/>, Viitattu 17.3.2016.
- [48] N. Mitra and Y. Lafon, “Soap version 1.2 part 0: Primer (second edition),” tech. rep., W3C, huhtikuu 2007. Saatavilla <https://www.w3.org/TR/2007/REC-soap12-part0-20070427/>. Viitattu 17.3.2016.
- [49] L. Clement, A. Hatley, C. von Riegen, and T. Rogers, “Uddi version 3.0.2: Uddi spec technical committee draft, dated 20041019,” tech. rep., OASIS, lokakuu 2004. Saatavilla <http://www.uddi.org/pubs/uddi-v3.0.2-20041019.htm>. Viitattu 19.3.2016.
- [50] M. B. Juric, *SOA Approach to Integration: XML, Web services, ESB, and BPEL in Real-World SOA Projects*. Packt Publishing Ltd, 2007.
- [51] J. Gable, “Enterprise application integration,” *Information Management Journal*, vol. 36, no. 2, p. 48, 2002. ARMA International.
- [52] N. Erasala, D. C. Yen, and T. Rajkumar, “Enterprise application integration in the electronic commerce world,” *Computer Standards & Interfaces*, vol. 25, no. 2, pp. 69–82, 2003. Elsevier.
- [53] K. K. Venugopal, J. G. Kupper, and P. Murray, “Eai & web services—a simple guide.” Saatavilla <http://www.capescience.com/articles/content/WebServicesandEAI.pdf>, 2001. Viitattu 24.2.2016.



- [54] L. Qilin and Z. Mintian, “The state of the art in middleware,” in *International Forum on Information Technology and Applications (IFITA), 2010*, vol. 1, pp. 83–85, IEEE, heinäkuu 2010.
- [55] J. Hu, F. Luo, J. Li, X. Tong, and G. Liao, “Soa-based enterprise service bus,” in *Electronic Commerce and Security, 2008 International Symposium on*, pp. 536–539, IEEE, 2008.
- [56] L. Garces-Erice, “Building an enterprise service bus for real-time soa: A messaging middleware stack,” in *Computer Software and Applications Conference, 2009. COMPSAC’09. 33rd Annual IEEE International*, vol. 2, pp. 79–84, IEEE, 2009.
- [57] S. Ortiz Jr., “Getting on board the enterprise service bus,” *Computer*, vol. 40, pp. 15–17, huhtikuu 2007. IEEE.
- [58] Viestintävirasto, “Määräys teletoiminnan häiriötilanteista.” Saatavilla [https://www.viestintavirasto.fi/attachments/maaraykset/M66\\_2014.pdf](https://www.viestintavirasto.fi/attachments/maaraykset/M66_2014.pdf), joulukuu 2014. Viitattu 31.3.2016.
- [59] M. Jokisalo, “Digitaalisen peittoalueiden laskenta ja paikkatietoaineiston tuottaminen karttapalveluun,” maaliskuu 2016. Haastattelu 29.3.2016.
- [60] Microsoft, “Exchange web services (ews) in exchange 2010.” Saatavilla <https://msdn.microsoft.com/en-us/library/office/dd877045%28v=exchg.140%29.aspx?f=255&MSPPError=-2147217396>, syyskuu 2010. Viitattu 4.10.2015.

# A Digitan karttapalvelun Web Services -rajapinnat



1 (14)

4.2.2015

**DIGITA**

## DIGITAN KARTTAPALVELUN WEB SERVICE –RAJAPINNAT

### Sisällys

Digitan karttapalvelun Web Service –rajapinnat .....	1
Yleistä .....	2
Perustiedot .....	2
Web Service –toiminnot .....	2
GetAllActiveSites .....	2
GetAllActiveServices .....	3
GetAllActiveSiteServices .....	4
GetCurrentOutages .....	5
GetPreventiveMaintenances .....	6
GetAllDigitaPro .....	8
GetAllDigitaProMunicipalities .....	9
GetSatellitePackages .....	9
GetBaseStations .....	10
Digitan WFS-rajapinnat .....	11
Perustiedot .....	11
Peittoalueet .....	11
Paras asema ja toiseksi paras asema .....	14

## Yleistä

Karttapalvelu Web Service tarjoaa ohjelmointirajapinnan erilaisten Digitan tarjoamien aineistojen noutoon. Rajapinnan tarjoama aineisto rajoittuu siihen dataan, jota Digitan julkisessa karttapalvelussa ollaan jollain tavalla hyödynnetty.

Web Service -rajapinta tarjoaa perustietoa mm. TV- ja radiokanavien nimistä, taajuuksista, lähetyksasemista sekä verkon häiriöistä. Kysely palauttaa aina ajantasaisen tietueen.

Vektorimuotoinen paikkatieto tarjotaan standardin Web Feature Servicen (WFS) kautta. Paikkatietoa Digitan tarjoamassa rajapinnassa ovat palveluiden näkyvyys- ja kuuluvuusalueet sekä ”paras asema aineisto”, joka määrittelee minkä asemien tv-lähetyksiä kannattaa mistäkin paikasta vastaanottaa.

## Perustiedot

### Testiympäristö

URI: [https://qw2.digita.fi/GIS\\_Service/proxy\\_services/DigitaGIS](https://qw2.digita.fi/GIS_Service/proxy_services/DigitaGIS)  
 WSDL: [https://qw2.digita.fi/GIS\\_Service/proxy\\_services/DigitaGIS?WSDL](https://qw2.digita.fi/GIS_Service/proxy_services/DigitaGIS?WSDL)

### Tuotantoympäristö

URI: [https://qw.digita.fi/GIS\\_Service/proxy\\_services/DigitaGIS](https://qw.digita.fi/GIS_Service/proxy_services/DigitaGIS)  
 WSDL: [https://qw.digita.fi/GIS\\_Service/proxy\\_services/DigitaGIS?WSDL](https://qw.digita.fi/GIS_Service/proxy_services/DigitaGIS?WSDL)

## Web Service –toiminnot

Lihavoidut kentät muodostavat aineistossa yksilöivän avaimen.

### GetAllActiveSites

Kuvaus: Palauttaa tiedot kaikista aktiivisista lähetyksasemista.

Kentän nimi	Selitys	Lisätietoja
<b>SiteId</b>	<b>Aseman tunnus</b>	<b>Digitan neljämerkkinen asematunnus</b>
SiteName	Aseman nimi	
SiteType	Aseman tyyppi	Mahdolliset arvot PÄÄASEMA, ALILÄHETINASEMA, MUUASEMA.
Owner	Aseman omistajayrityksen nimi	

PostalAddress	Aseman postiosoite	
PostalCode	Aseman postitoimipaikka	
MunicipalityCode	Virallinen kuntakoodi	Tätä kenttää voidaan käyttää sitomaan tietyt DigitaPRO-antenniasentajat tämän aseman tietoihin.
CoordLongitude	Leveyskoordinaatti	YKJ/KKJ3 maantieteellinen koordinaatisto (2D)
CoordLatitude	Pituuskoordinaatti	YKJ/KKJ3 maantieteellinen koordinaatisto (2D)
ContactPersonName	Aseman yhteys henkilön nimi	
SiteStatus	Aseman tila	Vakioarvona kaikilla ACTIVE, koska rajapinnan kautta ei muissa tiloissa olevia välitetä.
PublicSiteType	Aseman julkinen tyyppi	Digitan asiakasviestinnässä käytämä nimitys asemasta. Mahdolliset arvot:  1) Radio- ja TV-asema 2) Täytelähetinasema
SiteCategory	Aseman kategoria	Ilmaisee lähettääkö asema TV- vai radiokanavia vai kumpiakin. Mahdolliset arvot:  1) TV 2) RADIO 3) TV_AND_RADIO  Jos SiteCategory on tyhjä, ei asemalta lähetetä mitään TV- tai radiopalvelua.
StartDate	Aloituspäivämäärä	Päivämäärä, jolloin lähetysasema on aloittanut toimintansa.

**GetAllActiveServices**

Kuvaus: Palauttaa tiedot kaikista aktiivisista TV- ja radiokanavista.

Kentän nimi	Selitys	Lisätietoja
-------------	---------	-------------

ServiceId	Palvelun tunnus	Digitan sisäinen palvelutunnus
ServiceName	Palvelun nimi	Kanavan virallinen nimitys.
ServiceCategory	Palvelun kategoria	Luokittelee kanavan joko televisio- tai radiopalveluksi. Mahdolliset arvot TV ja RADIO.
ServiceStatus	Palvelun tila	Vakioarvona kaikilla ACTIVE.
Mux	Kanavanippu	Kanavanippu. Radiopalveluilla kenttä on tyhjä.
SortId	Lajitteluavain	Digitan ulkoisessa viestinnässä käytetty kanavajärjestys.
CoverageId1	Peittoalueen tunnus	Palvelun peittoalueen tunnus
CoverageId2	Peittoalueen tunnus	Ei käytössä TV- tai radiopalveluilla. Kenttä poikkeuksetta tyhjä.
LogicalChannelNumber	Palvelun looginen kanavnumero	Kokonaisluku
VisibleInMapService	Näkyvissä karttapalvelussa	1 = tiedot palvelusta on nähtävissä Digitan karttapalvelun kautta. 0= Digitan karttapalvelussa ei esitetä mitään palveluun liittyviä tietoja.

#### GetAllActiveSiteServices

Kuvaus: Palauttaa tiedot asemien tarjoamista TV- ja radiokanavista. Huomaa, että asemakohtaisella palvelulla voi olla eri nimi, kuin mikä on palvelun yleisnimi.

Kentän nimi	Selitys	Lisätietoja
SiteId	Aseman tunnus	Digitan sisäinen asematunnus. Ei julkiseen käyttöön.
ServiceId	Palvelun tunnus	Digitan sisäinen palvelutunnus. Ei julkiseen käyttöön.
ServiceName	Palvelun nimi	Kanavan asemakohtainen nimi.
SiteTypeForService	Aseman palvelukohtainen	Mahdolliset arvot PÄÄASEMA ja

	tyyppi	TÄYTELÄHETINASEMA
LinkedSite	Releoiva asema	Täytelähetinasemaa releoiva pääasema. Pääasemilla tämä kenttä on tyhjä.
RemuxSite	Remux-asema	
Frequence	Palvelun lähetystaajuus	Decimal, yksikkö MHz
ChannelNumber	Kanavanumero	Ns. taajuuskanava, kokonaisluku.
SiteServiceStatus	Aseman palvelun tila	Vakioarvona ACTIVE
CoverageId1	Palvelun asemakohtainen peittoalue	Asemakohtaiset peittoalueet eivät ole käytössä, mutta koko maan peittoalueaineistosta voi hakea yksittäisen aseman peittoalueen.
CoverageId2	Palvelun asemakohtainen peittoalue	Ei ole käytössä.

#### GetCurrentOutages

Kuvaus: Palauttaa kartalla näytettävät suunnittelemattomat katkokset eli häiriöt. Jos yhdellä häiriöllä on vaikutuksia useaan lähetysasemaan ja palveluun, palauttaa rajapinta jokaisen aseman ja palvelun osalta erillisen häiriötiedon.

Ratkaistusta häiriöstä palautetaan tiedot vielä yhden vuorokauden ajan.

Häiriöitä hakiessa on mahdollista käyttää vapaavalintaista parametria *StartDaysAgo*, joka määrittelee kuinka monta vuorokautta sitten olleita häiriötietoja halutaan hakea.

Kentän nimi	Selitys	Lisätietoja
<b>OutageId</b>	<b>Katkoksen tunnus</b>	<b>Digitan sisäinen häiriötiketin tunnus</b>
<b>SiteId</b>	<b>Aseman tunnus</b>	<b>Asema, jolla katkos esiintyy</b>
<b>ServiceId</b>	<b>Palvelun tunnus</b>	<b>Kanava, johon katkos kohdistuu</b>
OutageStartDateTime	Katkoksen alkamisaika	Päivämäärä ja kellonaika minuutin tarkkuudella
OutageEndDateTime	Katkoksen päättymisaika	Päivämäärä ja kellonaika minuutin tarkkuudella

WorkStartTime	[ei käytössä]	Ei käytössä häiriöillä
WorkEndTime	[ei käytössä]	Ei käytössä häiriöillä
Area	Katkoksen tyyppi	
SubArea	Katkoksen syy/vaikutus	
EstimatedOutageEffect	[ei käytössä]	Ei käytössä häiriöillä
OutageStatus	Katkoksen tila	INCIDENT, kun aseman kanavassa on häiriö. RESOLVED, kun häiriö tai huolto on päätynyt alle vuorokausi sitten.
OutageRepairStatus	Korjauksen tilanne	<b>Korjaus valmis</b> , kun aseman palvelun häiriö on päätynyt alle vuorokausi sitten. <b>Korjaus käynnissä</b> , kun aseman palvelussa on parhaillaan katkos.
OutageEstimatedEndDateTime	Katkoksen arvioitu päättymisaika	Digitan verkonhallintakeskuksen arvio katkoksen päättymisestä
UpdateDateTime	Päivitysajankohta	Aikaleima siitä milloin tietue on päivittynyt viimeksi Digitan tietokannassa.
OutageType	Katkoksen tyyppi	Vakioarvona NORMAL, joka tarkoittaa ennakoimatonta häiriötä.
OutageSecondaryStartDateTime	Katkoksen varaalkamispäivä	Ei käytössä häiriöillä
OutageSecondaryEndDateTime	Katkoksen varapäättymispäivä	Ei käytössä häiriöillä
CompanyName	Asiakkaan nimi	Yrityksen nimi, jolle tarjottavaan palveluun häiriö vaikuttaa

#### GetPreventiveMaintenances

Kuvaus: Palauttaa kartalla näytettävät suunnitellut katkokset eli ennakkohuollot korkeintaan kaksi viikkoa ennen niiden alkamisaikaa. Huollon tiedot näytetään sen päättymisen jälkeen vielä yhden vuorokauden ajan.

Kentän nimi	Selitys	Lisätietoja
-------------	---------	-------------

OutageId	Katkoksen tunnus	Digitan sisäinen ennakkokatkoksen tunnus muotoa xxxxxx-y-z.  xxxxxx = työtilausnumero y = huoltoilmoituksen numero z = valvonnan tila (1=katkos, 0=ei katkosta)
SiteId	Aseman tunnus	Asema, jolla katkos esiintyy
ServiceId	Palvelun tunnus	Palvelu, johon katkos kohdistuu
OutageStartDateTime	Katkoksen alkamisaika	Päivämäärä ja kellonaika minuutin tarkkuudella
OutageEndDateTime	Katkoksen päättymisaika	Päivämäärä ja kellonaika minuutin tarkkuudella
WorkStartTime	Huoltotyön aloitusaika	Kellonaika, jolloin huoltotyötä aletaan tekemään. Jos katkos on useampipäiväinen, aloituskellonaika koskee jokaista päivää.
WorkEndTime	Huoltotyön päättymisaika	Kellonaika, jolloin huoltotyön tekeminen päättyy. Jos katkos on useampipäiväinen, päättymiskellonaika koskee jokaista päivää.
Area	Katkoksen tyyppi	
SubArea	Katkoksen syy/vaikutus	
EstimatedOutageEffect	Huollon arvioitu palveluvaikutuksen kesto	Arvio huoltotyön aiheuttaman palveluvaikutuksen kestosta.
OutageStatus	Katkoksen tila	INCIDENT, kun aseman kanavassa on huolto, joka on aiheuttanut katkoksen. PENDING, kun aseman palvelussa on huolto, mutta palvelu ei välttämättä ole poikki. COMING, kun aseman palveluun on tulossa huolto kahden viikon kuluessa. RESOLVED, kun häiriö tai huolto on päättynyt alle vuorokausi sitten.
OutageRepairStatus	Huollon tilanne	<b>Huolto tulossa</b> , kun ennakkohuolto tai sen varapäivä on tulossa kahden viikon sisällä. <b>Huolto valmis</b> , kun ennakkohuolto on päättynyt alle vuorokausi sitten. <b>Huollettavana, ei katkosta juuri nyt</b> , kun palvelussa on ennakkokatkos juuri nyt, mutta



		palvelussa ei välttämättä ole katkosta. <b>Huolto käynnissä</b> , kun huolto on käynnissä ja palvelussa on katkos
OutageEstimatedEndDateTime	Katkoksen arvioitu päättymisaika	Ei käytössä ennakkohuolloilla. Arvioitu päättymisaika on aina sama kuin todellinen päättymisaika.
UpdateDateTime	Päivitysajankohta	Aikaleima siitä milloin tietue on päivittynyt viimeksi Digitan tietokannassa.
OutageType	Katkoksen tyyppi	Ennakkohuolloilla vakioarvo PREV_MAINT
OutageSecondaryStartDateTime	Katkoksen vara-alkamispäivä	Toissijainen huoltoajankohta, ns. varapäivä.
OutageSecondaryEndDateTime	Katkoksen varapäättymispäivä	Toissijainen huoltoajankohta, ns. varapäivä.
WorkSecondaryStartTime	Huoltotyön varapäivän aloitusaika	Kellonaika, jolloin huoltotyötä aletaan tekemään mikäli työ tehdään varapäivänä. Jos katkos on useampipäiväinen, aloituskellonaika koskee jokaista päivää.
WorkSecondaryStartTime	Huoltotyön varapäivän päättymisaika	Kellonaika, jolloin huoltotyön tekeminen päättyy mikäli työ tehdään varapäivänä. Jos katkos on useampipäiväinen, päättymiskellonaika koskee jokaista päivää.

#### GetAllDigitaPro

Kuvaus: Palauttaa kaikkien DigitaPRO-antenniasentajien yhteystiedot.

Kentän nimi	Selitys	Lisätietoja
<b>CompanyId</b>	<b>Antenniasentajan yrityksen tunnus</b>	<b>Digitan sisäinen yritystunnus</b>
CompanyName	Antenniasentajan nimi	
PostalAddress	Antenniasentajan postiosoite	
PostalCode	Antenniasentajan postinumero	

PostalCity	Antenniasentajan postitoimipaikka	
PhoneNumber	Antenniasentajan puhelinnumero	
EmailAddress	Antenniasentajan sähköpostiosoite	
WwwSite	Antenniasentajan www-sivu	

#### GetAllDigitaProMunicipalities

Kuvaus: Kunnat, joissa antenniasentajat toimivat. Antenniasentajat liitetään asemiin kuntakoodin avulla.

Kentän nimi	Selitys	Lisätietoja
<b>MunicipalityCode</b>	<b>Kuntakoodi</b>	<b>Asemaan liitetään ne antenniasentajat, jotka toimivat samassa kunnassa missä asema sijaitsee.</b>
MunicipalityName	Kunnan nimi	
<b>CompanyId</b>	<b>Antenniasentajan yrityksen nimi</b>	

#### GetSatellitePackages

Kuvaus: Palauttaa tiedon niistä sijainneista ja osoitteista, joihin on myönnetty vastaanotto-ongelmien vuoksi satelliittipaketti.

Kentän nimi	Selitys	Lisätietoja
<b>Id</b>	<b>Tunnus</b>	<b>Rivin avain</b>
Sites	Aseman nimi	Asema, jonka vaikutusalueella ongelma esiintyy. Vapaateksti.
SubSites	Täytelähetinaseman nimi	Täytelähetinasema, jonka vaikutusalueella ongelma esiintyy. Vapaateksti.
Lon	Pituuskoordinaatti	

Lat	Leveyskoordinaatti	
ContactName	Ilmoittajan nimi	
ContactAddress	Ilmoittajan katuosoite	
ContactMunicipality	Ilmoittajan kunta	
ContactPostalCode	Ilmoittajan postinumero	
ContactPostalOfficeName	Ilmoittajan postitoimipaikka	
ContactPhoneNumber	Ilmoittajan puhelinnumero	
ProblemType	Ongelman tyyppi	Vakioarvona "SAT-paketti"
Handler	Käsittelijä	Vastuuhenkilö Digita Info-kuluttajapalvelussa
AntennaCompany	Antenniasentajayritys	
ContactDate	Ilmoitusvm	
ProblemDescription	Ongelman kuvaus	
ProblemResolution	Ongelman ratkaisu	

#### GetBaseStations

Kuvaus: Palauttaa tiedon kaikista LTE 800 MHz tukiasemista.

Kentän nimi	Selitys	Lisätietoja
<b>Id</b>	<b>Tukiasematunnus</b>	<b>Digitan sisäinen tukiasematunnus</b>
ExternalId	Operaattorin tukiasematunnus	
Name	Operaattorin tukiasemanimi	
Lat	Leveyskoordinaatti	YKJ/KKJ3 maantieteellinen (2D)
Lon	Pituuskoordinaatti	YKJ/KKJ3 maantieteellinen (2D)
StartDate	Tukiaseman	

	aloituspäivämäärä	
OperatorName	Operaattorin nimi	Elisa Oyj/DNA/TSF
Municipality	Kunta	Tukiaseman sijaintikunta
Status	Tukiaseman tila	1 = käynnissä, 0 = sammutettu

## DIGITAN WFS-RAJAPINNAT

### Perustiedot

#### Testiympäristö

<servername> = <https://gw2.digita.fi>

#### Tuotantoympäristö

<servername> = <https://gw.digita.fi>

### Peittoalueet

<servername>/geoserver/digita\_network\_coverage/ows?service=WFS&version=2.0.0&request=GetFeature&typeName=<TypeName>&viewparams=service\_id:<ServiceId>;site\_id:<SiteId>;modified\_after:<Updated>

#### TypeName

digita\_network\_coverage:tv\_coverage\_area                      TV-palveluiden peittoalueet

digita\_network\_coverage:radio\_coverage\_area                      Radiopalveluiden peittoalueet

#### Viewparams

service\_id                      Palauttaa palvelutunnuksen mukaisen palvelun peittoalueen. Palvelutunnuksina voi käyttää kaikkia GetAllActiveServices-toiminnon palauttavia palvelutunnuksia.

site\_id                      Palauttaa aseman tunnusta vastaavan aseman peittoalueen. Jos site\_id-parametrin jättää kokonaan pois, palautetaan koko maan peittoalue.

**modified\_after** Palauttaa tietyn päivän jälkeen muuttuneet peittoalueet. Päivämäärä on oltava muotoa yyyymmdd.

#### Esimerkit

Haetaan koko ULA1 peittoalue Espoon asemalla mikäli se on muuttunut 31.12.2012 jälkeen:

[https://qw.digita.fi/geoserver/digita\\_network\\_coverage/ows?service=WFS&version=2.0.0&request=GetFeature&typeName=digita\\_network\\_coverage:radio\\_coverage\\_area&viewparams=service\\_id:ULA1;site\\_id:ESPO;modified\\_after:20121231](https://qw.digita.fi/geoserver/digita_network_coverage/ows?service=WFS&version=2.0.0&request=GetFeature&typeName=digita_network_coverage:radio_coverage_area&viewparams=service_id:ULA1;site_id:ESPO;modified_after:20121231)

Haetaan YLE1:n koko maan peittoalue:

[https://qw.digita.fi/geoserver/digita\\_network\\_coverage/ows?service=WFS&version=2.0.0&request=GetFeature&typeName=digita\\_network\\_coverage:tv\\_coverage\\_area&viewparams=service\\_id:DVB1\\_TV1](https://qw.digita.fi/geoserver/digita_network_coverage/ows?service=WFS&version=2.0.0&request=GetFeature&typeName=digita_network_coverage:tv_coverage_area&viewparams=service_id:DVB1_TV1)

#### Kenttien kuvaukset

Kentän nimi	Selitys	Lisätietoja
<b>CoverageId</b>	<b>Peittoalueen tunnus</b>	Peittoalueen tunnus, joka on viittaa kanavan peittoalueeseen.
<b>MiPrinx</b>	<b>Peittoalueen osanumero</b>	Kanavn koko maan peittoalue muodostuu useasta alueesta. Nämä alueet on numeroitu 1...N.
Geoloc	Peittoalueen geometria	Peittoalue
MiStyle	Peittoalueen tyyli	Määrää peittoalueen värin, viivan paksuuden sekä peittoaluun kuvionnin MapInfossa.
Layer	Peittoaluden järjestys	Voidaan käyttää peittoalueiden kerrostamiseen sovelluksessa.  10 = Pääaseman lievealue 20 = Pääaseman ydinalue 30 = Täytelähetinaseman lievealue 40 = Täytelähetinaseman ydinalue
Sort		
SiteId	Aseman tunnus	Kertoo sen aseman, johon peittoalue kuuluu



ServiceId	Palvelun tunnus	TV- tai radiokanavan tunnus, johon peittoalue kuuluu.
ServiceName	Palvelun nimi	TV- tai radiokanavan nimi, johon peittoalue kuuluu.

## Paras asema ja toiseksi paras asema

Kyselyn avulla voidaan etsiä lähetyasema, jonka signaali on vahvin tai toiseksi vahvin tietyssä maantieellisessä pisteessä. Haku voidaan tehdä palvelukohtaisesti.

<servername>/geoserver/digita\_network\_coverage/ows?service=WFS&version=2.0.0&request=GetFeature&typeName=<TypeNime>&viewparams=service\_id:site\_id:<SiteId>

### TypeName

digita\_network\_coverage:best\_server                      Tietyssä sijainnissa oleva paras TV:n lähetyasema

digita\_network\_coverage:second best\_server                      Tietyssä sijainnissa oleva toiseksi paras TV:n lähetyasema

### Viewparams

site\_id                      Palauttaa geometrian, jonka alueella aseman TV-signaali on vahvin tai toiseksi vahvin. Jos parametrin jättää pois, rajapinta palauttaa koko maan aineiston. Parametriksi voidaan antaa myös site\_id:n alkuosa, esim site\_id:A palauttaa kaikki A:lla alkavien asemien aineiston.

Kentän nimi	Selitys	Lisätietoja
<b>Legend</b>	<b>Aseman tunnus</b>	
Threshold	Signaalin voimakkuuden raja-arvo	
Color		
Tx_Id		
Prediction_Name		
Mi_Prinx		
Geoloc	Peittoalueen geometria	