Deliverable No.: D 5

eMOTION System – Analysis of Technical Standards

Appendix 1
April 2008
Final 1.0

Project funded by the European Community under the Sixth Framework Programme for Research and Technological Development.
Deliverable title: Analysis of Technical Standards
Deliverable number: D 5
Deliverable status: Public
Number of pages: 358
Author(s): Reinhard Erstling (interactive instruments – editor)
Stefan Olk (interactive instruments – editor)
Alexio Picco (Azienda Mobilità e Infrastrutture)
Marko Jandrisits (ASFINAG)
Norbert Hainitz (ATTC - Arsenal)
Walter Schneider (ATTC - Arsenal)
Rainer Rehberger (ATTC / CNS-Solutions & Support GmbH)
Carmen Riedler (ATTC - Kapsch)
Karl Schedler (micKS)
Andreas Kochs (Momatec)
Daniel Heckert (Momatec)
Harald Doderer (OneStepAhead)
Nihat Küçük (OneStepAhead)
Michele Masnata (Softeco)
Linde Vande Velde (Tele Atlas)
Marco Annoni (Telecom Italia)
Tomas Starek (Telematix Services)
Grant MacKinnon (Transport Simulation Systems)

<table>
<thead>
<tr>
<th>Status</th>
<th>Version</th>
<th>Date</th>
<th>Change Note ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft</td>
<td>0.1</td>
<td>22.01.2007</td>
<td>First Internal Release</td>
</tr>
<tr>
<td>Draft</td>
<td>0.2</td>
<td>11.02.2007</td>
<td>Work assignments, structuring instructions for content standards</td>
</tr>
<tr>
<td>Draft</td>
<td>0.3</td>
<td>12.03.2007</td>
<td>Integration of contributions by mmt, TA,</td>
</tr>
<tr>
<td>Draft</td>
<td>Date</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>20.03.2007</td>
<td>Further integration of contributions by OSA, TA, mmt, Softeco, TSS, ATTC, AMI, GdC, TMX. Structuring by micKS.</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>24.04.2007</td>
<td>Further work by II, TMX, Softeco, ATTC, micKS. Changes due to Ghent WP3-PM1.</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>02.05.2007</td>
<td>Further work by TSS (new: 2.4.5, 2.8.6; upd: 2.2.4)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TI (new: 5.2, 5.3)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMI (note to 2.7.4)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TMX (upd 2.4.2, 2.9.3)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>mmt (upd: 2.4.1, 2.5.1, 2.5.2, 2.6.1, 2.7.1, 2.10.4; new: 2.6.5, 4.2.2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>II (upd. 3.2.4)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Also: Former subsection 2.8.6 is divided into 2.2.5, 2.9.6 and 2.12.4 - see remarks there.</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>07.05.2007</td>
<td>Further work by ATTC (new (Arsenal): 2.2.3, 2.8.2, 2.8.10; upd (Kapsch): 2.6.3; upd (Arsenal): 2.8.4); mmt (change responsibility of 2.5.2, 2.6.4);</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>micKS (upd. 2.10.1, 2.10.2; new: 2.10.3); ii (new: 2.3.1);</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TMX (upd 2.6.2, 2.7.2, 2.7.3, 2.8.9; new stub 4.7.1); Softeco (upd. 2.2.2, 2.8.1, 2.8.11)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Also: Former subsections 2.8.3, 2.8.4 and 2.14.1 are deleted, subsection 2.2.4 is divided in 2.2.4 and 2.12.5 (NaP-TAN/NPTG);</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>27.05.2007</td>
<td>Further work by ATTC (new stub 4.2.3, upd 2.6.3); TI (upd. 3.2.3, 3.2.5, 5.2, 5.3, new 5.4, 5.5); Softeco (upd. 2.2.2, 2.8.1, 2.12.6);</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TMX (upd.2.4.2, 2.6.2, 2.7.3, 2.8.7, 2.9.3); ii (new: 4.2.1, 4.11.1, 4.14.1, 4.15.1, 4.16); TSS (upd. 2.2.4, 2.2.5, 2.4.5, 2.9.6,</td>
<td></td>
</tr>
<tr>
<td>Draft</td>
<td>Date</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>Draft</td>
<td>0.9 04.06.2007</td>
<td>Further work by Softeco (upd. 2.2.2, 2.8.1, 2.12.3; new 5.6) ASFINAG (upd. 2.5.2) mmt (upd. 2.4.1, 2.5.1, 2.6.1, 2.6.5, 4.2.2; new: 2.5.3, 2.6.6, 2.7.5, 4.5.2, 6.3, 6.4, 6.7) TSS (upd. 2.2.4, 2.2.5, 2.4.5, 2.8.3, 2.8.5, 2.9.6, 2.12.4, 2.12.5, 2.12.6, 4.5.1, 6.6) TMX (upd. 2.4.2, 2.6.4 (dropped), 2.8.7, new 4.7.1) ii (upd. 2.4.6, 4.5.3, 4.5.4; new 4.7.2) ATTC (new: 2.2.7, 2.8.9, 2.12.7, 6.2, 6.6) micKS (upd. 2.10.1, 2.10.2, 2.10.3, 2.10.4, 6.5) TA (upd. 2.2.1, new 6.1, 6.8, 6.9)</td>
<td></td>
</tr>
<tr>
<td>Draft</td>
<td>1.0 08.07.2007</td>
<td>Reorganisation: creation of Appendix 1, restructuring main document; Additions in Appendix 1: ii (new 2.1.3, 2.1.4, 4.2, 4.3.2, 4.4.1, 4.4.2, 4.6, 4.6.1, 4.8, 4.8.1, 4.8.2, 4.9, 4.9.1, 4.12, 4.12.1, 4.12.2, 4.12.3, 4.13, 4.13.1, 4.14, 4.15) ATTC (new: 4.2.3; upd. 2.8.2) Softeco (new 5.1) TSS (new: 2.4.7) micKS (upd. 2.10, 6.5)</td>
<td></td>
</tr>
<tr>
<td>Final Draft</td>
<td>1.1 20.07.2007</td>
<td>Insertion of standardised literature references from all over the text. Complete “List of Abbreviations”. ii (new 2.13, 2.4.4, renumbering 2.4.x), OSA (new 2.9.7; upd. 2.9.3, 2.9.4, 2.9.5),</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Version</td>
<td>Changes and Updates</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
<td>---------------------</td>
<td></td>
</tr>
<tr>
<td>31.08.2007</td>
<td>1.2</td>
<td>SOAP and ISO 19115 added.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incorporated results of review by TSS.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MMT+II: New section 2.1.4 describing the LCL-implied network. Follow-up changes in 2.4.2 and 2.3.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TMX+MMT: Update of 2.8.6.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ATTC+MMT: Update of 2.2.3.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incorporated results of review by TSS.</td>
<td></td>
</tr>
</tbody>
</table>

THE INFORMATION IN THIS DOCUMENT IS PROVIDED AS IS AND NO GUARANTEE OR WARRANTY IS GIVEN THAT THE INFORMATION IS FIT FOR ANY PURPOSE. THE USER THEREOF USES THE INFORMATION AT ITS SOLE RISK AND LIABILITY. FURTHERMORE, DATA, CONCLUSIONS OR RECOMMENDATIONS IN THIS DOCUMENT ARE PROVIDED ON THE BASIS THAT SUCH INFORMATION IS SUBSEQUENTLY, AND PRIOR TO USE, VERIFIED BY THE PARTY WISHING TO USE THAT INFORMATION.
**List of Abbreviations**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADT</td>
<td>Abstract Data Types</td>
</tr>
<tr>
<td>Alert-C</td>
<td>Advice and Problem Location for European Road Traffic, version C</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>ASR</td>
<td>Automatic Speech Recognition</td>
</tr>
<tr>
<td>ATOC</td>
<td>Association of Train Operating Companies</td>
</tr>
<tr>
<td>ATT</td>
<td>Advanced Transport Telematics</td>
</tr>
<tr>
<td>BUFR</td>
<td>Binary Universal Form for the Representation of meteorological data</td>
</tr>
<tr>
<td>CALYPSO</td>
<td>Contact And Contactless Telematics platform Yielding a Citizen Pass integrating urban Services and financial Operations</td>
</tr>
<tr>
<td>CAT</td>
<td>Catalogue Interface standard</td>
</tr>
<tr>
<td>CEN</td>
<td>European Committee for Standardization</td>
</tr>
<tr>
<td>COURIER</td>
<td>Cross-border Organisation &amp; User Requirements for Information Exchange Review</td>
</tr>
<tr>
<td>CREX</td>
<td>Character form for the Representation and EXchange of data</td>
</tr>
<tr>
<td>CRS</td>
<td>Coordinate Reference System</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma Separated Values</td>
</tr>
<tr>
<td>CSW</td>
<td>Catalogue Services for the Web</td>
</tr>
<tr>
<td>DANAE</td>
<td>Dynamic and Distributed Adaptation of scalable multimedia content in a context-Aware Environment</td>
</tr>
<tr>
<td>DATEX2</td>
<td>(version 2 of European standard for traffic and travel) data exchange (between traffic control and information centres as well as other actors of the traffic and travel information sector)</td>
</tr>
<tr>
<td>DDA</td>
<td>Disability Discrimination Act</td>
</tr>
<tr>
<td>DELFI</td>
<td>Nationwide Electronic Time Table Information - German: Durchgängige Elektronische Fahrplaninformation</td>
</tr>
<tr>
<td>DfT</td>
<td>UK Department for Transport</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name System</td>
</tr>
<tr>
<td>DRM</td>
<td>Digital Rights Management</td>
</tr>
<tr>
<td>DSML</td>
<td>Directory Services Markup Language</td>
</tr>
<tr>
<td>ebXML</td>
<td>Electronic Business using eXtensible Markup Language</td>
</tr>
<tr>
<td>EC</td>
<td>European Commission</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>EDGE</td>
<td>Enhanced Data Rates for GSM Evolution</td>
</tr>
<tr>
<td>EFCD</td>
<td>Enhanced Floating Car Data</td>
</tr>
<tr>
<td>eMOTION</td>
<td>Europe-wide multi-Modal On-trip Traffic InformaTION</td>
</tr>
<tr>
<td>EPC</td>
<td>European Payments Council</td>
</tr>
<tr>
<td>eps</td>
<td>Electronic Payment System</td>
</tr>
<tr>
<td>ETSI</td>
<td>European Telecommunications Standards Institute</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>GDF</td>
<td>Geographic Data Files</td>
</tr>
<tr>
<td>GDS</td>
<td>Global Distribution Systems</td>
</tr>
<tr>
<td>GeoDRM</td>
<td>Geospatial Digital Rights Management</td>
</tr>
<tr>
<td>GML</td>
<td>Geography Markup Language</td>
</tr>
<tr>
<td>GPRS</td>
<td>General Packet Radio Service</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communications</td>
</tr>
<tr>
<td>GST</td>
<td>Global System for Telematics</td>
</tr>
<tr>
<td>Halogen</td>
<td>Highways Agency Logging Environment</td>
</tr>
<tr>
<td>HAPMS</td>
<td>Highways Agency Pavement Management System</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hyper Text Transfer Protocol</td>
</tr>
<tr>
<td>HTTPS</td>
<td>Hyper Text Transfer Protocol Secure</td>
</tr>
<tr>
<td>IATA</td>
<td>International Air Transport Association</td>
</tr>
<tr>
<td>ICARE</td>
<td>Integration of Contactless technologies into public transport environment</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IFOPT</td>
<td>Identification of Fixed Objects in Public Transport</td>
</tr>
<tr>
<td>IM@GINE IT</td>
<td>Intelligent Mobility AGents, Advanced Positioning and Mapping Technologies INtEgration Interoperable MulTimodal, location-based services</td>
</tr>
<tr>
<td>IMAP</td>
<td>Internet Message Access Protocol or Interactive Mail Access Protocol</td>
</tr>
<tr>
<td>IMS</td>
<td>IP Multimedia Subsystem</td>
</tr>
<tr>
<td>INFOTEN</td>
<td>Multi-modal Information and Traffic Management Systems on Trans-</td>
</tr>
</tbody>
</table>
European Networks

INSPIRE Infrastructure for Spatial Information in the European Community
InteGRail Intelligent Integration of Railway systems
INTERCEPT Intermodal Concepts in European Passenger Transport
INTRO Intelligent roads
IP Internet Protocol
IPA International Phonetic Alphabet
ISO International Organization for Standardization
ITS Intelligent Transport Systems
ITSO Interoperable smart card ticketing
J2ME Java Platform, Mobile Edition
JTDB Journey Time Database
LDAP Lightweight Directory Access Protocol
LoS Level of Service
MIME Multimedia Internet Message Extensions
NaPTAN National Public Transport Access Node Database
NPTG National Public Transport Gazetteer
OASIS Organization for the Advancement of Structured Information Standards
OGC Open Geospatial Consortium
OpenLS OpenGIS® Location Service
OSI Open Systems Interconnection
OTA OpenTravel Alliance
OTAP Open Travel Data Access Protocol
OTC Open Tourism Consortium
PCC Project Coordination Committee
PDA Personal digital assistants
PLS Pronunciation Lexicon Specification
POI Point of Interest
POIX Point Of Interest eXchange Language Specification
POP Post Office Protocol
PT Public Transport
RDS Radio Data System
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REACT</td>
<td>Realizing Enhanced Safety and Efficiency in European Road Transport</td>
</tr>
<tr>
<td>RJIS</td>
<td>Rail Journey Information System</td>
</tr>
<tr>
<td>RM-ODP</td>
<td>Reference Model for Open, Distributed Processing</td>
</tr>
<tr>
<td>RTIG</td>
<td>Real Time Information Group</td>
</tr>
<tr>
<td>SAML</td>
<td>Security Assertion Markup Language</td>
</tr>
<tr>
<td>SAMPA</td>
<td>Speech Assessment Methods Phonetic Alphabet</td>
</tr>
<tr>
<td>SDEP</td>
<td>Street events Data Exchange Protocol</td>
</tr>
<tr>
<td>SDI</td>
<td>Spatial Data Infrastructure</td>
</tr>
<tr>
<td>SEPA</td>
<td>Single Euro Payments Area</td>
</tr>
<tr>
<td>SIRI</td>
<td>Service Interface for Real Time Information</td>
</tr>
<tr>
<td>SLD</td>
<td>Styled Layer Descriptor</td>
</tr>
<tr>
<td>SMIL</td>
<td>Synchronized Multimedia Integration Language</td>
</tr>
<tr>
<td>SMS</td>
<td>Short Message Service</td>
</tr>
<tr>
<td>SMTP</td>
<td>Simple Mail Transfer Protocol</td>
</tr>
<tr>
<td>SNMP</td>
<td>Simple Network Management Protocol</td>
</tr>
<tr>
<td>SOA</td>
<td>Service Oriented Architecture</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>SRW</td>
<td>Scheduled Road Works</td>
</tr>
<tr>
<td>SSH</td>
<td>Secure Shell</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer</td>
</tr>
<tr>
<td>SSML</td>
<td>Speech Synthesis Markup Language</td>
</tr>
<tr>
<td>SyncML</td>
<td>Synchronisation Markup Language</td>
</tr>
<tr>
<td>TAP</td>
<td>Transferred Account Procedure</td>
</tr>
<tr>
<td>TC</td>
<td>Technical Committee</td>
</tr>
<tr>
<td>TCC</td>
<td>Technical Coordination Committee</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TIC</td>
<td>Traffic Information Center</td>
</tr>
<tr>
<td>TIH</td>
<td>Travel Information Highway</td>
</tr>
<tr>
<td>TLS</td>
<td>German: 'Technische Lieferbedingungen für Streckenstationen' (Technical delivery conditions for roadside controllers)</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
</tbody>
</table>
TourML  Tourism Markup Language
TPEG  Transport Protocol Experts Group
TRIDENT  TRansport Intermodality Data sharing and Exchange NeTworks
TTI  Travel Technology Initiative
TTS  Text-To-Speech
TourXML  Tourism Markup Language
TPEG  Transport Protocol Experts Group
TMC  Traffic Message Channel
UDDI  Universal Description, Discovery, and Integration
UDP  User Datagram Protocol
UML  Unified Modeling Language
UMTS  Universal Mobile Telecommunications System
URL  Uniform Resource Locator
UTMC  Urban Traffic Management and Control
VIH  Video Information Highway
W3C  World Wide Web Consortium
WAI  Web Accessibility Initiative
WAP  Wireless Application Protocol
WCS  Web Coverage Service
WCTS  Web Coordinate Transformation Service
WFS  Web Feature Service
WiMAX  Worldwide interoperability for Microwave Access
WMS  Web Map Service
WNS  Web Notification Service
WP  Work Package
WPL  Work Package Leader
WPOS  Web Pricing and Ordering Services
WS-BPEL  Web Services Business Process Execution Language
WSDL  Web Service Description Language
WSN  Web Services Notification
XACML  eXtensible Access Control Markup Language
XCPF  XML Configuration and Pricing Format
XML

Extensible Markup Language
Table of Content

1. OVERVIEW ..................................................................................................................... 23

2. ANALYSIS OF CONTENT STANDARDS ...................................................................... 24
   2.1 ROAD NETWORK DATA .............................................................................................. 24
       2.1.1 ISO 14825: Geographic Data Files (GDF) - Road Network ................................. 24
       2.1.2 ISO 19133 - Chapter 9: Network ......................................................................... 29
       2.1.3 EuroRoadS - Road Network Information Model .................................................. 31
       2.1.4 ISO 14819: RDS-TMC - Chapter 3: Location Table Implied Network .................. 33
       2.1.5 Comparison and Evaluation ................................................................................. 39
   2.2 PUBLIC TRANSPORT NETWORK DATA ................................................................. 41
       2.2.1 ISO 14825: Geographic Data Files (GDF) - Public Transport ............................... 41
       2.2.2 ENV 12896: Transmodel ....................................................................................... 44
       2.2.3 Railway Markup Language (RailML®) .................................................................. 58
       2.2.4 Rail Journey Information System (RJIS) ............................................................... 60
       2.2.5 National Public Transport Access Node Database (NaPTAN) ................................ 61
       2.2.6 Identification of Fixed Objects in Public Transport (IFOPT) ................................. 64
       2.2.7 Comparison and Evaluation ................................................................................. 70
   2.3 INTER-MODAL TRANSPORT NETWORK DATA ...................................................... 71
       2.3.1 ISO 19134 - Chapter 6: Multimodal Network ........................................................ 72
       2.3.2 ENV 12896: Transmodel - “Connection Links” ....................................................... 75
       2.3.3 Identification of Fixed Objects in Public Transport (IFOPT) - “Connection Links” .... 76
       2.3.4 Rail Journey Information System (RJIS) - Fixed Leg ............................................ 77
       2.3.5 Comparison and Evaluation ................................................................................. 78
   2.4 LOCATION REFERENCING ....................................................................................... 79
       2.4.1 DATEX 2................................................................................................................. 79
       2.4.2 ISO 14819: RDS-TMC - Chapter 3: Location referencing for ALERT-C .................... 84
       2.4.3 ISO 19133 - Section 6.6: Package Linear Reference Systems .................................. 87
       2.4.4 ISO 19133 - Sections 6.2.2 & 8.2: Position and Address Package ........................... 89
       2.4.5 OGC OpenLS - Core Services, Abstract Data Types (ADT) .................................... 92
       2.4.6 ISO 18234-6 / ISO 24530-2: TPEG-LOC - Location Referencing Applications ........ 95
       2.4.7 ISO 17572-3: AGORA-C ..................................................................................... 102
       2.4.8 Comparison and Evaluation ................................................................................. 106
   2.5 TRAFFIC FLOW DATA ............................................................................................. 108
       2.5.1 DATEX 2................................................................................................................. 108
       2.5.2 Journey Time Database (JTDB) ............................................................................. 113
2.5.3 Comparison and Evaluation ........................................................................................ 115

2.6 TRAFFIC MESSAGES ................................................................................................. 115

2.6.1 DATEX 2 .................................................................................................................. 115


2.6.3 ISO 14819 - RDS-TMC ........................................................................................... 127

2.6.4 Scheduled Road Works (SRW) ............................................................................... 135

2.6.5 Comparison and Evaluation .................................................................................... 143

2.7 PARKING .................................................................................................................... 143

2.7.1 DATEX 2 .................................................................................................................. 144

2.7.2 ISO 18234-7 / ISO 24530-5: TPEG-PKI - Parking Information ................................. 145

2.7.3 UTMC Standard - TS004: Car Park Monitor MIB .................................................. 146

2.7.4 Comparison and Evaluation .................................................................................... 150

2.8 PUBLIC TRANSPORT SERVICE DATA ................................................................... 150

2.8.1 ENV 12896: Transmodel .......................................................................................... 150

2.8.2 Service Interface for Real Time Information (SIRI) .................................................. 155

2.8.3 Transport Exchange (TransXChange) .................................................................... 168

2.8.4 Railway Markup Language (RailML®) ..................................................................... 171

2.8.5 Rail Journey Information System (RJIS) .................................................................. 176

2.8.6 ISO 18234-5 / ISO 24530-4: TPEG-PTI - Public Transport Information .................. 177

2.8.7 OpenTravel Alliance (OTA) Standard and Ferry XML ............................................ 186

2.8.8 Comparison and Evaluation .................................................................................... 186

2.9 POI AND OTHER DIRECTORIES .............................................................................. 188

2.9.1 ISO 14825: Geographic Data Files (GDF) - Feature Theme “Services” (POIs) ........... 188

2.9.2 OGC OpenLS - Core Services, Abstract Data Types (ADT) ..................................... 191

2.9.3 Directory Services Standards (DSML) v1.0 ............................................................... 192

2.9.4 Point Of Interest eXchange Language Specification (POIX) .................................... 195

2.9.5 Tourism Markup Language (TourML) ..................................................................... 199

2.9.6 Identification of Fixed Objects in Public Transport (IFOPT) .................................... 200

2.9.7 National Public Transport Gazetteer (NPTG) .......................................................... 202

2.9.8 Comparison and Evaluation .................................................................................... 202

2.10 ROAD WEATHER .................................................................................................... 203

2.10.1 TLS Standard - Environmental Data (FG3) ............................................................ 203

2.10.2 Binary Universal Form for the Representation of meteorological data (BUFR) / Character form for the Representation and EXchange of data (CREX) ...................... 207

2.10.3 ISO 14819: RDS-TMC - Road Weather related ALERT-C Codes ......................... 212

2.10.4 DATEX 2 .................................................................................................................. 213

2.10.5 Comparison and Evaluation .................................................................................... 217
2.11 DATA FOR ROUTING .............................................................................................................. 218
  2.11.1 ISO 19133 - Chapter 7 Navigation and Chapter 9 Network ........................................ 218
  2.11.2 OGC OpenLS - Core Services, Route Service and Abstract Data Types (ADT) ....... 221
  2.11.3 Comparison and Evaluation ....................................................................................... 223
2.12 DATA FOR PUBLIC TRANSPORT JOURNEY PLANNING ....................................................... 223
  2.12.1 ENV 12896: Transmodel ............................................................................................. 224
  2.12.2 Identification of Fixed Objects in Public Transport (IFOPT) ...................................... 227
  2.12.3 National Public Transport Gazetteer (NPTG) ............................................................ 228
  2.12.4 Comparison and Evaluation ....................................................................................... 231
2.13 DATA FOR INTER-MODAL JOURNEY PLANNING ................................................................. 232
  2.13.1 ISO 19134 – Multimodal Routing and Navigation ...................................................... 232
  2.13.2 Comparison and Evaluation ....................................................................................... 234
2.14 DATA FOR FREIGHT TRAFFIC .............................................................................................. 234
2.15 GENERAL METADATA ........................................................................................................... 235
  2.15.1 ISO 19115 – Metadata ................................................................................................ 235
  2.15.2 Comparison and Evaluation ....................................................................................... 238
3. ANALYSIS OF ENCODING STANDARDS ............................................................................... 239
  3.1 CONTENT ENCODINGS .................................................................................................... 239
    3.1.1 Geography Markup Language (GML) ......................................................................... 239
  3.2 PRESENTATION ENCODINGS ......................................................................................... 242
    3.2.1 Synchronized Multimedia Integration Language (SMIL) ........................................... 243
    3.2.2 Voice Extensible Markup Language (VoiceXML) ...................................................... 244
    3.2.3 Speech Synthesis Markup Language specification (SSML) ..................................... 246
    3.2.4 Speech Assessment Methods Phonetic Alphabet (SAMPA / X-SAMPA) ................ 246
    3.2.5 Pronunciation Lexicon Specification (PLS) ................................................................ 247
4. ANALYSIS OF SERVICE STANDARDS ..................................................................................... 248
  4.1 APPLICATION SERVICES .................................................................................................. 248
  4.2 DATA SERVICE ................................................................................................................. 249
    4.2.1 OGC Web Feature Service (WFS) ............................................................................. 249
    4.2.2 OGC Web Coverage Service (WCS) ......................................................................... 252
    4.2.3 Datex 2 Data Service ............................................................................................... 253
    4.2.4 Service Interface for Real Time Information (SIRI) ................................................... 260
  4.3 MAPPING SERVICE .......................................................................................................... 268
    4.3.1 OGC Web Map Service (WMS) ................................................................................ 268
    4.3.2 OGC Web Map Service (WMS) with Styled Layer Descriptor (SLD) ....................... 270
  4.4 ROUTING SERVICE ......................................................................................................... 272
### 4.4.1 ISO 19133 - Section 7.3, Package: Navigation Service ........................................ 272

### 4.4.2 OGC OpenLS - Core Services, Part 5 Route Service ........................................ 274

### 4.5 PUBLIC TRANSPORT JOURNEY PLANNING SERVICES ........................................ 275

#### 4.5.1 JourneyWeb ........................................................................................................ 276

#### 4.5.2 Nationwide Electronic Time-table Information - Durchgängige Elektronische Fahrplaninformation (DELFI) ................................................................. 280

#### 4.5.3 OpenTravel Alliance (OTA) Standard .................................................................. 288

#### 4.5.4 Ferry XML .......................................................................................................... 292

### 4.6 POSITIONING SERVICE ...................................................................................... 295

#### 4.6.1 OGC OpenLS - Core Services, Part 2 Gateway Service .................................... 295

### 4.7 DIRECTORY SERVICE .......................................................................................... 296

#### 4.7.1 Directory Services Standards (DSML) v2.0 ...................................................... 296

#### 4.7.2 OGC OpenLS - Core Services, Part 1 Directory Service .................................... 298

### 4.8 GEOCODING SERVICE ......................................................................................... 299

#### 4.8.1 OGC OpenLS - Core Services, Part 3 Location Utility Service ......................... 299

#### 4.8.2 OGC Web Feature Service (WFS) - Gazetteer Service Profile ....................... 300

### 4.9 COORDINATE TRANSFORMATION SERVICE .................................................... 301

#### 4.9.1 OGC Web Coordinate Transformation Service (WCTS) .......................... 301

### 4.10 REGISTRY / CATALOGUE SERVICE .................................................................. 302

#### 4.10.1 OASIS ebXML Registry Services and Protocols ........................................... 303

#### 4.10.2 OGC Catalogue Services Specification ......................................................... 306

#### 4.10.3 OGC Catalogue Services - ebRIM Profile of CSW .................................... 307

#### 4.10.4 OGC Catalogue Services - ISO19115/ISO19119 Profile of CSW .................. 308

### 4.11 EVENT NOTIFICATION SERVICE ....................................................................... 309

#### 4.11.1 OASIS Web Services Notification (WSN) ...................................................... 310

#### 4.11.2 OGC Web Notification Service (WNS) .......................................................... 312

### 4.12 DIGITAL RIGHTS MANAGEMENT AND SECURITY ........................................... 314

#### 4.12.1 OGC Geospatial Digital Rights Management Reference Model (GeoDRM RM) .......... 314

#### 4.12.2 OASIS eXtensible Access Control Markup Language (XACML) / GeoXACML ............................................................... 315

#### 4.12.3 OASIS Security Assertion Markup Language (SAML) ....................................... 316

### 4.13 PRICING AND ORDERING ................................................................................ 317

#### 4.13.1 OGC Web Pricing and Ordering Service (WPOS) ........................................... 317

### 4.14 PAYMENT AND BILLING .................................................................................. 318

#### 4.14.1 Electronic Payment System (eps) ................................................................. 319

#### 4.14.2 Single Euro Payments Area (SEPA) ............................................................... 320

### 4.15 WORKFLOW SUPPORT ..................................................................................... 321
4.15.1 OASIS Web Services Business Process Execution Language (WS-BPEL)............. 321
4.16 NETWORK MANAGEMENT .....................................................................................322
4.16.1 Simple Network Management Protocol (SNMP) ................................................. 322
4.17 NATURAL LANGUAGE TRANSLATION.................................................................323
4.17.1 OASIS Translation Web Service (Trans-WS) ....................................................... 324
4.17.2 SYSTRANLinks ..................................................................................................... 324

5. ANALYSIS OF NETWORK AND COMMUNICATION STANDARDS .................325

5.1 INTERNET COMMUNICATION STANDARDS .................................................325
5.1.1 Hyper Text Transfer Protocol (HTTP) ................................................................. 326
5.1.2 Hyper Text Transfer Protocol Secure (HTTPS) .................................................... 327
5.1.3 Secure Sockets Layer / Transport Layer Security (SSL/TLS)............................... 327
5.1.4 Secure Shell (SSH) ............................................................................................. 328
5.1.5 Domain Name System (DNS) ............................................................................. 328
5.1.7 File Transfer Protocol (FTP) ................................................................................. 331
5.1.8 Transmission Control Protocol (TCP) .................................................................. 332
5.1.9 User Datagram Protocol (UDP) ......................................................................... 333
5.1.10 IPv4/IPv6/IPsec .................................................................................................. 334
5.1.11 SOAP Version 1.2 ............................................................................................. 334
5.2 PUBLIC WIRELESS NETWORKS(GSM/GPRS/EDGE/UMTS) .........................336
5.2.1 Standards ............................................................................................................. 336
5.2.2 Data Service Throughput .................................................................................... 337
5.2.3 Cell radius, Coverage and capacity ..................................................................... 338
5.2.4 Mobility ............................................................................................................... 339
5.2.5 Application protocols .......................................................................................... 339
5.3 IP MULTIMEDIA SUBSYSTEM (IMS) .................................................................339
5.3.1 IMS general description ...................................................................................... 339
5.3.2 Architecture ....................................................................................................... 340
5.3.3 Quality of Service (QoS) .................................................................................... 343
5.3.4 Security .............................................................................................................. 343
5.3.5 Charging ............................................................................................................. 344
5.3.6 Service opportunities ......................................................................................... 344
5.4 WORLDWIDE INTEROPERABILITY FOR MICROWAVE ACCESS (WiMAX) ......346
5.4.1 The WiMAX Forum ............................................................................................ 346
5.4.2 WiMAX performances ....................................................................................... 346
5.4.3 WiMAX and services .......................................................................................... 348
5.4.4 Interworking with the mobile network: short term scenario ........................................... 349
5.4.5 Interworking with the mobile network: long term scenario ......................................... 349

5.5 WIRELESS COMMUNICATION PROTOCOLS .................................................................... 350
5.5.1 IP Versions - Transition tools ...................................................................................... 350
5.5.2 IP Versions in GSM/GPRS/EDGE/UMTS Standards .................................................. 351
5.5.3 IP Versions in Mobile Telecom Networks ................................................................. 352

5.6 HAND-HELD DEVICES AND USER TERMINALS ............................................................ 354
5.6.1 J2ME ............................................................................................................................ 354
5.6.2 SyncML ........................................................................................................................ 357

List of Figures

Figure 2-1: GDF - Road Network Model .................................................................................. 25
Figure 2-2: GDF - Features ................................................................................................... 27
Figure 2-3: GDF - Feature Relations ....................................................................................... 28
Figure 2-4: ISO19133 - Simplified Network Model ................................................................. 29
Figure 2-5: EuroRoadS – Road Network (simplified) .............................................................. 32
Figure 2-6 RDS-TMC Implied Network .................................................................................. 35
Figure 2-7: GDF - Public Transport Features ........................................................................ 42
Figure 2-8: GDF - Public Transport Relationship ................................................................. 44
Figure 2-9: Transmodel - Transport network .......................................................................... 46
Figure 2-10: Transmodel - Types of POINTs and LINKs .......................................................... 48
Figure 2-11: Transmodel - Infrastructure .............................................................................. 50
Figure 2-12: Transmodel – Link sequences ............................................................................ 51
Figure 2-13: Transmodel - Grouping ...................................................................................... 52
Figure 2-14: Transmodel – Basic elements in the definition of ROUTEs .................................. 53
Figure 2-15: Transmodel - Definition of LINEs ...................................................................... 57
Figure 2-16: UML diagram of the RailML infrastructure ....................................................... 59
Figure 2-17: RailML infrastructure elements ......................................................................... 59
Figure 2-18: RJIS Master Station Names Record .................................................................... 61
Figure 2-19: NaPTAN - Overview Class Diagram (excluding administrative classes) .......... 62
Figure 2-20: IFOPT - Stop Place Basic Model ....................................................................... 66
Figure 2-21: IFOPT - Stop Place Model View. Describes the most basic stop information ...67
Figure 2-22: IFOPT - Stop Place Assignment Model .............................................................68
Figure 2-23: IFOPT - Stop Place Components Type Hierarchy .............................................69
Figure 2-24: IFOPT - Stop Place Components Containment Hierarchy ................................69
Figure 2-25: ISO 19133 - package “Combined Networks” .....................................................72
Figure 2-26: ISO 19134 - multi-modal network ...................................................................74
Figure 2-27: IFOPT - Example Navigation Paths possible with IFOPT within and between Stop Places ............................................................................................................................77
Figure 2-28: RJIS Fixed Leg ..................................................................................................77
Figure 2-29: DATEX 2 - Location : compositestructure diagram ........................................81
Figure 2-30: DATEX 2 - Point : compositestructure diagram ...............................................82
Figure 2-31: DATEX 2 - Linear : compositestructure diagram .............................................82
Figure 2-32: DATEX 2 - Area: compositestructure diagram ..............................................83
Figure 2-33: DATEX 2 - PredefinedLocationsPublication : compositestructure diagram ......84
Figure 2-34: RDS-TMC - Location Reference ........................................................................85
Figure 2-35: ISO 14819-3 - Primary and Secondary Locations ...........................................86
Figure 2-36: ISO 19133 - Linear Referencing .....................................................................88
Figure 2-37: ISO 19133 - Position .......................................................................................90
Figure 2-38: ISO 19133 - Address Model ............................................................................91
Figure 2-39: OGC OpenLS - Location ...............................................................................93
Figure 2-40: OGC OpenLS - Address ...............................................................................94
Figure 2-41: OGC OpenLS - Position ................................................................................95
Figure 2-42 : TPEG-LOC - Location Container Class Diagram .........................................96
Figure 2-43 : TPEG LOC - Location Coordinates Class Diagram ......................................97
Figure 2-44: TPEG-LOC - Location Description Class diagram .......................................98
Figure 2-45 : TPEG-LOC - Area Reference Class Diagram ...............................................98
Figure 2-46: TPEG-LOC - Network Reference Class Diagram .........................................99
Figure 2-47: TPEG LOC - Node Reference Class Diagram ..............................................100
Figure 2-48 : TPEG-LOC - Reference object .....................................................................101
Figure 2-49 : AGORA-C - Dynamic Location Reference ..................................................102
Figure 2-50 : AGORA-C - Linear Location .........................................................................104
Figure 2-51 : AGORA-C - Implicit Areas .............................................................................105
Figure 2-52: AGORA-C - Explicit Areas .............................................................................105
<table>
<thead>
<tr>
<th>Figure Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-53</td>
<td>DATEX 2 - BasicDataValue : compositestructure diagram</td>
</tr>
<tr>
<td>2-54</td>
<td>DATEX 2 - TrafficValue : compositestructure diagram</td>
</tr>
<tr>
<td>2-55</td>
<td>DATEX 2 - XML schema generation process</td>
</tr>
<tr>
<td>2-56</td>
<td>JTD - Features</td>
</tr>
<tr>
<td>2-57</td>
<td>DATEX 2 - SituationPublication : logical diagram</td>
</tr>
<tr>
<td>2-58</td>
<td>DATEX 2 - SituationRecord : compositestructure diagram</td>
</tr>
<tr>
<td>2-59</td>
<td>DATEX 2 - Advice : logical diagram</td>
</tr>
<tr>
<td>2-60</td>
<td>DATEX 2 - TrafficElement : logical diagram</td>
</tr>
<tr>
<td>2-61</td>
<td>DATEX 2 - Roadworks: logical diagram</td>
</tr>
<tr>
<td>2-62</td>
<td>TPEG-RTM - Application Event/Status RTM Container</td>
</tr>
<tr>
<td>2-63</td>
<td>RDS-TMC - Message Structure</td>
</tr>
<tr>
<td>2-64</td>
<td>RDS-TMC - UML Classes for RDS-TMC events</td>
</tr>
<tr>
<td>2-65</td>
<td>RDS-TMC - Subset of real events</td>
</tr>
<tr>
<td>2-66</td>
<td>RDS-TMC - Classification of forecast events</td>
</tr>
<tr>
<td>2-67</td>
<td>RDS-TMC - Typical format of an RDS message according to IEC 62106:2000</td>
</tr>
<tr>
<td>2-68</td>
<td>SRW - Data Model</td>
</tr>
<tr>
<td>2-69</td>
<td>HAPMS network - Data Model</td>
</tr>
<tr>
<td>2-70</td>
<td>DATEX 2 - NonRoadEventInformation : compositestructure diagram</td>
</tr>
<tr>
<td>2-71</td>
<td>UTMC - Car Park MIB</td>
</tr>
<tr>
<td>2-72</td>
<td>Transmodel – Timetabled passing times</td>
</tr>
<tr>
<td>2-73</td>
<td>SIRI - ProductionTimetableDelivery</td>
</tr>
<tr>
<td>2-74</td>
<td>SIRI - EstimatedTimetableDelivery</td>
</tr>
<tr>
<td>2-75</td>
<td>SIRI - StopTimetableDelivery</td>
</tr>
<tr>
<td>2-76</td>
<td>SIRI - ConnectionTimetableDelivery</td>
</tr>
<tr>
<td>2-77</td>
<td>SIRI - DisruptionGroup</td>
</tr>
<tr>
<td>2-78</td>
<td>SIRI - OperationalInfoGroup</td>
</tr>
<tr>
<td>2-79</td>
<td>SIRI - VehicleJourneyInfoGroup</td>
</tr>
<tr>
<td>2-80</td>
<td>SIRI - StopMonitoringDelivery</td>
</tr>
<tr>
<td>2-81</td>
<td>TransXChange - Standard Service Model</td>
</tr>
<tr>
<td>2-82</td>
<td>TransXChange - Flexible Service Model</td>
</tr>
<tr>
<td>2-83</td>
<td>TransXChange - Interchange Model</td>
</tr>
<tr>
<td>2-84</td>
<td>railML - Timetable</td>
</tr>
</tbody>
</table>
Figure 2-85: railML - Rollingstock
Figure 2-86: RJIS - Basic Timetable Detail Data Record
Figure 2-87: RJIS - Train Schedule Record
Figure 2-88: TPEG-PTI - Application Event/Status PTI Container
Figure 2-89: Public Transport Information diagram
Figure 2-90: service_information diagram
Figure 2-91: transport_service_identifier diagram
Figure 2-92: class diagramm "event_reason" with descriptive enumerations
Figure 2-93: transport_submode diagramm
Figure 2-94: route_description diagram
Figure 2-95: GDF - ServiceFeature
Figure 2-96: GDF - Service Relationships
Figure 2-97: OGC OpenLS - POI
Figure 2-98: DSML - DSML document
Figure 2-99: POIX - Features
Figure 2-100: TourML - Features
Figure 2-101: IFOPT - Point of Interest Model
Figure 2-102: TLS - Data Model of Application Layer
Figure 2-103: typical architecture of information chain
Figure 2-104: BUFR / CREX - Data Model for Road Weather Information
Figure 2-105: DATEX 2 - WeatherValue : compositestructure diagram
Figure 2-106: ISO19133 - Route Type
Figure 2-107: ISO19133 Instruction List
Figure 2-108: OGC OpenLS – Route Description Objects
Figure 2-109: Transmodel - Passenger Trips
Figure 2-110: IFOPT - Topographical and Gazetteer Model
Figure 2-111: NPTG - Overview Class Diagram (excluding administrative elements)
Figure 2-112: NPTG - Discovery Schema relevant classes
Figure 2-113: ISO 19134 - Multi-modal Route
Figure 2-114: ISO 19134 - Multi-modal Trip Scheme
Figure 2-115: ISO 19115 - Package “metadata entity set information”
Figure 3-1: GML - RISE UML model
Figure 3-2: VoiceXML - Sequence and timing during an example of a bridge transfer
Figure 4-1: DATEX 2 - Exchange Metadata.................................................................254
Figure 4-2: DATEX 2 - subscription: operating mode 0........................................256
Figure 4-3: DATEX 2 - Exchange Data Reference: operating mode 3.................257
Figure 4-4: DATEX 2 - SupplierPushOnOccurrence: operating mode 1..............258
Figure 4-5: DATEX 2 - SupplierPushPeriodic: operating mode 2..........................259
Figure 4-6: DATEX 2 - ClientPull: operating mode 3.............................................259
Figure 4-7: SIRI - SubscriptionRequest.................................................................264
Figure 4-8: SIRI - TerminateSubscriptionRequest ................................................265
Figure 4-9: SIRI – ProductionTimetableRequest.....................................................266
Figure 4-10: SIRI - EstimatedTimetableRequest.....................................................266
Figure 4-11: SIRI - StopTimetableRequest..............................................................267
Figure 4-12: SIRI - StopMonitoringRequest............................................................267
Figure 4-13: SIRI - ConnectionTimetableRequest....................................................268
Figure 4-14: DELFI - simplified process...............................................................282
Figure 4-15: FerryXML - Message flow.................................................................294
Figure 4-16: EPS Workflow..................................................................................319
Figure 4-17: SEPA “replacement” of Direct Debit and Credit Transfer schemes.......320
Figure 5-1: Internet Communication Standards - TCP/IP model and the OSI model325
Figure 5-2: DNS - Principles..................................................................................329
Figure 5-3: FTP - functionality..............................................................................332
Figure 5-4: SOAP Message structure....................................................................335
Figure 5-5: IMS - the new control layer...............................................................340
Figure 5-6: IMS - architecture.............................................................................341
Figure 5-7: IMS - Service Architecture...............................................................345
Figure 5-8: IEEE 802.16 and WiMAX.................................................................346
Figure 5-9: WiMAX - theoretic performances (gross bit rates).............................347
Figure 5-10: WIMAX - WLAN-UMTS Interworking Architecture: short term scenario ...349
Figure 5-11: WIMAX-WLAN-UMTS Interworking Architecture: long term scenario...350
Figure 5-12: Wireless Communication Protocols – Dual stack Stack....................351
Figure 5-13: Wireless Communication Protocols - Extranet with “6 to 4”.............351
Figure 5-14: Wireless Communication Protocols - Dual Stack UE connecting to IPv4 and IPv6 nodes ..........................................................352
Figure 5-15: Wireless Communication Protocols - IPv6 UE connecting to an IPv6 node
through an IPv4 network ..................................................................................................................352

Figure 5-16: J2ME - Layers for the Java 2 Platform Micro Edition......................................................356
1. Overview

Appendix 1 of D5 presents a detailed analysis of standards which have been deemed relevant for eMOTION.

The primary criteria for the decomposition were chosen as follows:

- Content
- Encodings
- Services
- Network and Communication

The “Content” chapter views the various standards according to different technical domains. As noted in the main document of D5, the criteria analysed in all these domains for all standards were

- “Feature Catalogue”, which was carried out as elucidated UML diagrams and constitutes the main results of the analysis. XML defined service specifications were stripped from their purely service related parts.
- “Encoding”, which was usually not elaborated in depth,
- “Portrayal Catalogue”, which was aimed at detecting mapping standards (but gave no results),
- “Service Reference”, which was to document the use of the content in a service context.

The chapter “Encodings” logically belongs to chapter “Services”. It accommodates the analysis of a few important encodings, which will or may play a role in the eMOTION specification. The chapter is in no way complete, because most encodings have been deemed part of the services that are using them and were described alongside with these.

Presentations of "Services" are also described in different categories. Here the criteria were

- “Functionality”, pointing out the overall range of functions the service under considerations performs,
- “Interface”, describing the way the service can be invoked.

The standards treated in the “Communication and Network” chapter do not overlap with the road and traffic oriented domains of eMOTION. The analysis of these standards has been done in anticipation of the Engineering Viewpoint of the specification in D6.
2. Analysis of Content Standards

Based on the service definition (as set out in deliverable D2) and considering the requirements and policies (as described in deliverable D3) existing and emerging standards, both international and European, are evaluated.

This chapter begins with the analysis of content related standards, focusing on the eMOTION specific information domains. The analysis concentrates on the following issues:

- Application schema (including spatial and temporal references)
- Specific metadata
- Encodings (exchange formats)
- Styling and presentation

Since the distinction between content related standards and service related ones (which are treated in the next chapter) is not always strict and clear, service related issues are discussed, if these are specific to the individual information domain.

For some of the information domains listed below no suitable and usable content standards currently exist. On the other hand, often information domains are covered by more than one content standard.

Each of the sections concerning the standards for one content domain is followed by a short discussion called “Comparison and Evaluation”. The idea is to find out, which standards seem suitable as the basis for the development of the eMOTION data model, and to detect possible needs for harmonisation.

2.1 Road Network Data

Road network data is static data describing the road network. It contains the geometry of the road elements, the topologic relations and possibly attributes regarding traffic rules, constructional aspects and road infrastructure entities.

Road network data is related to most other content types, because it delivers reference data items for these.

2.1.1 ISO 14825: Geographic Data Files (GDF) - Road Network

ISO14825:2004 / GDF describes a data model for representing real-world geographic phenomena on a conceptual level, a semantic level, and a physical level. An overview of GDF is available in the main document at subsection 3.5.7.

Feature Catalogue

The road network model is specified in different parts in ISO14825, namely the feature catalogue, attribute catalogue, and relationship catalogue. The following UML diagrams are taken from the XGDF project in ISO, depicting the most relevant feature classes, attributes types
and relationship types which are defined within the overall conceptual GDF model for topological networks.

Figure 2-1: GDF - Road Network Model

The road network is seen primarily from the viewpoint of transportation and traffic. The most basic features constituting the road network are Road Elements and Junctions, representing individual carriageways and their connectivity. Additionally, Ferry Connections (representing either boat or train ferries) are therefore placed together with road network elements in one theme. Furthermore, Enclosed Traffic Areas (for traffic on areas) and Address Areas (for addresses bounding an area) can be modelled. In a future extension (XGDF) Pathway features are added for complementing the roads & ferries network for non-motorized users,
covering both exterior and interior connectivity, such as pedestrian walkways, bicycle paths, or in-building passageways.

Next to this elementary level of road network representation, referred to as GDF Level 1, aggregation of Level 1 features is supported in order to compose a higher (generalized) level view, called GDF Level 2. On Level 2, the “simple features” are aggregated to a higher-level feature. For instance at level 1 all road elements of an intersection should be represented. At level 2, the intersection is only represented with a single point.

The higher-level (complex) road network is composed of Roads, Intersections, and Ferries, each being assembled from Level 1 (simple) features. A complex Intersection is assembled from Junction(s) and/or Road Element(s) and/or Enclosed Traffic Area(s), while a Road is assembled from Road Element(s). A Road Element may not be part of more than one assembly into a Road, or Intersection respectively, thus enabling a set of fully complementary complex features.

Apart from this, other complex features can be assembled, including Roundabouts, Interchanges (entire motorway exists), and Aggregated Ways (e.g. designated routes). These usually correspond to higher-level aggregations than Intersections or Roads.

Level 2 is mostly used when a simplified description of the road network is sufficient. For instance inter-urban route calculation does not require a high level of detail. Vehicle location by means of a GPS receiver however, does need the detailed description of the road network.
Figure 2-2: GDF - Features

Level 1 features like Road elements, Rivers, Boundaries, signposts etc. can have attributes that are specific to the feature. i.e. one way, width of the road, number of lanes. Addressing information (street names, house numbers, etc.) are modelled by means of a composite attribute called Address Information.
Features can also have relations. These relations are very important for navigation systems. Relationships can be “prohibited turn from road element 1 to road element 2” or “road element 1 has priority over road element 2”.

**Encodings**

ISO14825 provides for an ASCII-based format based on data records. XGDF extensions with include an XML schema, and a SQL-MM compliant realization of table CREATE statements for a standardized layout of relational tables.

**Portrayal Catalogue**

There is no portrayal associated with this standard.

**Service Reference**

In a strict sense, there is no specific service definition associated with this standard. In a
wider sense, however, the GDF model is adopted for designing future incremental map update services (prototypes stage), i.e. provision of change-only information conceptually and semantically compliant with ISO14825 to serve mapping and navigation applications which “run on GDF maps”.

2.1.2 ISO 19133 - Chapter 9: Network

ISO 19133:2005 describes the data types, and operations associated with those types, for the implementation of tracking and navigation services. An overview of the standard is available in the main document at subsection 3.4.5.


Feature Catalogue

The following UML diagram is a simplified formulation of the model in ISO 19133 targeting the network model. It leaves out transient constructs from 19133 for representing the additional constraints necessary for specifying a proper routing request. For full details, please see the ISO standard.

Figure 2-4: ISO19133 - Simplified Network Model
An instance of a `NT_Network`, which is derived from `TP_Complex` (from 19107) is essentially two separate topologies. Its underlying geometric topology is the value of the associated “complex”. The second topology is the graph of the `NT_Link`, `NT_Junction`, and `NT_Turn` entities that comprise it. Although the links, junctions and turns have the underlying geometry of the `TP_Complex` they have their own connectivity based on usable “vehicle” routes. If a link comes to a cross-roads and U-turns are allowed there are up to 4 turns which exit that link and enter into one of the links associated to a directed edge leaving that node including the one that reversed the incoming link. This micro topology does not have a different geometric existence from the “road” network but represents the potential traffic patterns through each of the “intersections” represented by the `TP_Nodes` of the associated `TP_Complex`.

Dimensionality is 1, i.e. faces do not play a role in the network. `NT_Junctions` are derived from `TP_Node` and `NT_Links` are derived from `TP_DirectedEdge`.

`NT_Links`, standing for the basic oriented curve elements in the network topology, have a few attributes, which describe the navigability of each link (or edge in the graph). These are the Boolean attributes `isTraversable` and `isUTurnPossible`, which speak for themselves, as well as `routeSegmentCategory`, which is a set of codelist items as follows:

- road, ferry, rail, walkway, subway, lighttrail, bus, bikepath, toll, tunnel, endway, bridge

In terms of navigation the instances of type `NT_Junction` are where turns occur. A `junction-Type` attribute describes junctions as follows:

- intersection, roundabout, enclosedTrafficArea, exitRamp, entranceRamp, changeOver, boardingRamp, station, transfer

`NT_Turns` model the possible turns from one `NT_Link` to another one emanating from that `NT_Junction`. Each turn\(^1\) will be located at a node of the underlying topology and will form a bridge for a link entering the node and a link exiting the node. The word “turn” does not necessarily reflect a change of direction. Going straight through an intersection is one such viable option. The latter is expressed by the attribute `turnDirection`, which can take the following values:

- straight, keepLeft, slightLeft, left, sharpLeft, uTurn, sharpRight, right, slightRight, keepRight

Other attributes with evident meaning are `isTraversable` and `isManeuver`.

The type `NT_Maneuver` is used to describe maneuvers. A maneuver is a legal sequence of actions, given by a sequence of turns, each of which terminates on the link that is the start of the next turn. Most maneuvers are single turns, but, for some turns, traversability is dependent on route history (as a ending turn in a U-turn across a divided highway). Since in this case the turn’s traversability cannot be decided by a single link’s history, the maneuver extends that history to a sufficient length to determine traversability.

---

\(^1\) In public transport such as a rail system the `NT_Turns` are routes through stations connecting an incoming train with one outgoing train. `NT_Turns` are therefore called “connections” in these environments and they have temporal constraints based on the availability of the links (trains).
Encodings
There is no encoding associated with this standard.

Portrayal Catalogue
There is no portrayal associated with this standard.

Service Reference
There is no specific service definition associated with this standard. It is, however, intended that this standard is to be used by Location Based Services as defined in ISO 19133.

2.1.3 EuroRoadS - Road Network Information Model

EuroRoadS was a project aiming at the definition of common specifications for the quality assured provision and exchange of road data. An overview of the project is available in the main document at subsection 3.1.3.

Feature Catalogue
The EuroRoadS-project results comprise a general information model for a road network, the EuroRoadS Core Model. The model is abstract enough to encompass most national road databases in Europe, it is however complex enough to support advanced structures like the two-level modelling of the road network in GDF.

The model is strictly based on the ISO TC 211 series of standards, especially

- ISO 19103 for using UML as conceptual schema language,
- ISO 19107 for the spatial schema including topology,
- ISO 19108 for the temporal schema,
- ISO 19109 for rules for defining application schemas, and
- ISO 19133 for linear referencing (see 2.4.3).

To make things better comparable to other models, we present a somewhat simplified excerpt form the EuroRoadS road network model. Left away were mainly the objects to describe paths through networks – routes.

The following UML diagram describes the EuroRoadS road network model:
The road network is made up of ER_RoadnetElements, which are identifiable ER_RoadFeatures. The ER_RoadnetElements are either ER_RoadNodes or ER_RoadnetLinks, which represent the nodes and links of the road network. ER_RoadnetLinks are realised as either ER_RoadLinks or ER_FerryLinks.

Abstract and arbitrary attributes can be attached to ER_RoadnetElements via the association to ER_RoadAttributes. However, some attributes are specific to nodes and links.

ER_RoadNodes own a formOfNode property, which can take values as follows:

- ER_EnclosedTrafficArea, ER_GradeSeparatedCrossing, ER_Junction, ER_PseudoNode, ER_RoadEnd, ER_Roundabout, ER_TrafficSquare, ER_Unknown.

ER_RoadnetLinks optionally own a validity period attribute, i.e. links underly historisation.

ER_RoadLinks own both a formOfWay:

- ER_BicycleRoad, ER_DualCarriageway, ER_EnclosedTrafficArea, ER_EnteranceOrExitCar-Park, ER_EnteranceOrExitService, ER_Freeway, ER_Motorway, ER_PedestrianZone, ER_-Roundabout, ER_ServiceRoad, ER_SingleCarriageway, ER_SlipRoad, ER_Tractor, ER_-TrafficSquare, ER_Walkway, ER_Unknown

and a functionalRoadClass property:

- ER_MainRoad, ER_FirstClass, ... ER_NinthClass, ER_Unknown
**2.1.4 ISO 14819: RDS-TMC - Chapter 3: Location Table Implied Network**

RDS-TMC (Radio Data System – Traffic message Channel) is a Europe wide standard for transferring traffic relevant information to mobile devices. An overview of the standard is available in the main document at subsection 3.1.7.

This subsection focuses on RDS-TMC Location Tables, which among other structures define a linked network of points, onto which ALERT-C messages are referenced.

For the location referencing aspect see 2.4.2, for the general message structure see 2.6.3.

Most RDS-TMC messages provide information about a location (e.g. a stretch of road, an intersection, a region) and they refer to it by using a location reference. This is an identifier that can be interpreted without ambiguity by the receiving system. In RDS-TMC, locations are pre-defined and pre-coded, and the codes are stored in location code tables. The maximum number of codes in one table is determined by the field length for location codes in RDS-TMC, i.e. 16 bit which corresponds to 65,536 possible codes.
The disadvantage of these tables is that they need to be created and maintained. The receiving system must use exactly the same Location code table as the one used for encoding of the message. Otherwise the message is not receivable. The rules for location reference coding are specified in ISO 14819-3. These rules apply to ALERT-C messages only.

Pre-defined locations are referenced by their location code which is the tabular address of a number of pre-stored location details. Each table of stored locations must be given a unique location table number by one unique agency in each country or state. A country code (note: RDS-TMC uses the RDS country codes) identifies the agency responsible for location reference coding and which defined the location table and its number.

Many location references extend through several adjacent areas or road sections. The concept of primary and secondary locations is then used to indicate the extremities of the affected sections, without having to list all the intervening places. For example, if an accident occurs at “km 14.2” on the E15 (A26 road in France) and the resulting queue extends back to “km 10.9”, the situation location can be defined as E15, “km 14.2 - 10.9”. Where “km 14.2” is defined as the primary location and “km 10.9” is then the secondary location. The primary location is taken to be where the cause of the problem can be found, whenever a cause can be pin-pointed geographically. However, both, primary and secondary location shall lie on the same road. For the primary location, the location reference is the nearest downstream location in the direction of travel. The secondary location is indicated in terms of extent, i.e. the number of steps back along the road, through other pre-defined locations. Alternatively, a distance marker may be used.

All location codes belong to a unique location table. Within any particular location code table each location has one unique number in the range 1 - 63,487. The other 2048 numbers are reserved for EUROAD, an agreed concept used for coding messages to international travellers on the Trans European Road Network (TERN).

RDS-TMC uses a hierarchical structure of pre-defined locations. A system of pointers provides upwards references to higher level locations containing the specified location. For example, Kent would have an upward area reference to South-East England which will be upwards referenced to the UK, then the British Isles, then Europe.
Feature Catalogue

The central class of the RDS-TMC network model is the RdsLocationTable. This class is described by two attributes, the countyCode which points to the country in the location table and the locationTableNumber that describes the version of the table. Typically the table is updated once a year.

The RdsLocationTable is a composition of different predefined locations. RdsPredefinedLocations owns a RdsLocationCode, a RdsLocationType and a firstname as attributes. RdsLocationType described the kind of location. The location type is mainly describe by the categoryCode (A = Area location, L = Linear location, P = Point location), a typeCode and a sub-typeCode that represents different kinds of Point, Linear and Area Locations. These codes are shown in the following table:

![Figure 2-6 RDS-TMC Implied Network](image-url)
<table>
<thead>
<tr>
<th>AREA LOCATIONS</th>
<th>LINEAR LOCATIONS</th>
<th>POINT LOCATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Code</strong></td>
<td><strong>Represented object</strong></td>
<td><strong>Code</strong></td>
</tr>
<tr>
<td>A1.0</td>
<td>Continent</td>
<td>L1.0</td>
</tr>
<tr>
<td>A2.0</td>
<td>Country group</td>
<td>L1.1</td>
</tr>
<tr>
<td>A3.0</td>
<td>Country</td>
<td>L1.2</td>
</tr>
<tr>
<td>A5.0</td>
<td>Water area</td>
<td>L1.3</td>
</tr>
<tr>
<td>A5.1</td>
<td>sea</td>
<td>L1.4</td>
</tr>
<tr>
<td>A5.2</td>
<td>lake</td>
<td>L2.0</td>
</tr>
<tr>
<td>A6.0</td>
<td>Fuzzy area</td>
<td>L2.1</td>
</tr>
<tr>
<td>A6.1</td>
<td>tourist area</td>
<td>L2.2</td>
</tr>
<tr>
<td>A6.2</td>
<td>metropolitan area</td>
<td>L3.0</td>
</tr>
<tr>
<td>A6.3</td>
<td>industrial area</td>
<td>L4.0</td>
</tr>
<tr>
<td>A6.4</td>
<td>traffic area</td>
<td>L5.0</td>
</tr>
<tr>
<td>A6.5</td>
<td>meteorological area</td>
<td>L6.0</td>
</tr>
<tr>
<td>A6.6</td>
<td>carpool area</td>
<td>L6.1</td>
</tr>
<tr>
<td>A6.7</td>
<td>park and ride site</td>
<td>L6.2</td>
</tr>
<tr>
<td>A6.8</td>
<td>car park area</td>
<td></td>
</tr>
<tr>
<td>A7.0</td>
<td>Order 1 area</td>
<td></td>
</tr>
<tr>
<td>A8.0</td>
<td>Order 2 area</td>
<td></td>
</tr>
<tr>
<td>A9.0</td>
<td>Order 3 area</td>
<td></td>
</tr>
<tr>
<td>A9.1</td>
<td>rural county</td>
<td></td>
</tr>
<tr>
<td>A9.2</td>
<td>urban county</td>
<td></td>
</tr>
<tr>
<td>A10.0</td>
<td>Order 4 area</td>
<td></td>
</tr>
<tr>
<td>A11.0</td>
<td>Order 5 area</td>
<td></td>
</tr>
<tr>
<td>A12.0</td>
<td>Application Region</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P3.15</td>
<td>customs post</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------</td>
<td></td>
</tr>
<tr>
<td>P3.16</td>
<td>toll plaza</td>
<td></td>
</tr>
<tr>
<td>P3.17</td>
<td>ferry terminal</td>
<td></td>
</tr>
<tr>
<td>P3.18</td>
<td>harbour</td>
<td></td>
</tr>
<tr>
<td>P3.19</td>
<td>square</td>
<td></td>
</tr>
<tr>
<td>P3.20</td>
<td>fair</td>
<td></td>
</tr>
<tr>
<td>P3.21</td>
<td>Garage</td>
<td></td>
</tr>
<tr>
<td>P3.22</td>
<td>underground garage</td>
<td></td>
</tr>
<tr>
<td>P3.23</td>
<td>retail park</td>
<td></td>
</tr>
<tr>
<td>P3.24</td>
<td>theme park</td>
<td></td>
</tr>
<tr>
<td>P3.25</td>
<td>tourist attraction</td>
<td></td>
</tr>
<tr>
<td>P3.26</td>
<td>university</td>
<td></td>
</tr>
<tr>
<td>P3.27</td>
<td>airport</td>
<td></td>
</tr>
<tr>
<td>P3.28</td>
<td>station</td>
<td></td>
</tr>
<tr>
<td>P3.29</td>
<td>hospital</td>
<td></td>
</tr>
<tr>
<td>P3.30</td>
<td>church</td>
<td></td>
</tr>
<tr>
<td>P3.31</td>
<td>stadium</td>
<td></td>
</tr>
<tr>
<td>P3.32</td>
<td>palace</td>
<td></td>
</tr>
<tr>
<td>P3.33</td>
<td>castle</td>
<td></td>
</tr>
<tr>
<td>P3.34</td>
<td>town hall</td>
<td></td>
</tr>
<tr>
<td>P3.35</td>
<td>exhibition/convention centre</td>
<td></td>
</tr>
<tr>
<td>P3.36</td>
<td>community</td>
<td></td>
</tr>
<tr>
<td>P3.37</td>
<td>place name</td>
<td></td>
</tr>
<tr>
<td>P3.38</td>
<td>dam</td>
<td></td>
</tr>
<tr>
<td>P3.39</td>
<td>dike</td>
<td></td>
</tr>
<tr>
<td>P3.40</td>
<td>aqueduct</td>
<td></td>
</tr>
<tr>
<td>P3.41</td>
<td>lock</td>
<td></td>
</tr>
<tr>
<td>P3.42</td>
<td>mountain crossing/pass</td>
<td></td>
</tr>
<tr>
<td>P3.43</td>
<td>railroad crossing</td>
<td></td>
</tr>
<tr>
<td>P3.44</td>
<td>wade</td>
<td></td>
</tr>
<tr>
<td>P3.45</td>
<td>ferry</td>
<td></td>
</tr>
<tr>
<td>P3.46</td>
<td>industrial area</td>
<td></td>
</tr>
</tbody>
</table>
The RdsPredefinedLocation have three child elements, which are RDSAreaLocation, RdsLinearLocation and RdsPointLocation.

The RdsAreaLocations can be differ in Administrative Areas und Other Areas. There is a hierarchy for Administrative Areas from Continent via Country Group and Country up to Order 5 Area. Every RdsAreaLocation except Continent has a upwardAreaRef relationship to the locationType of the next hierarchy level. The Other Areas can be differ in Water Areas, Fuzzy Areas, Application Regions, etc. These areas have a Area Reference to the smallest possible surrounding Administrative Area.

RdsLinearLocations describe parts of the road network. They can be divided in Roads, Ring Roads, Order 1 Segments, Order 2 Segments, Urban Streets and Vehicular Links. RdsLinearLocations can be described by the attributes roadNumber, roadName and secondName. Roads, Order 1 Segments, Order 2 Segments and Vehicular Links have a direction: They have a named positive and a negative end. Every linear segment knows his antecessor and his follower by the relationship upwardLinearReference, assumed they exist. Also RdsLinearLocation can have a Area Reference to a RdsAreaLocation, in which they are located. Order 1 Segments divide Roads and Ring Roads; Order 2 Segments divide Order 1 Segments. The affiliation of a Linear Location to a hierarchy higher positioned Linear Location is expressed by a Linear Reference.

RdsPointLocations describe points and junctions in a road network. They are specifiable into the three child elements RdsJunction, RdsIntermediatePoint and RdsOtherPoint. The RdsJunctions describe nodes in a linear network in which the linear elements start and end. Therefore RdsJunctions are basic elements of the topological network. RdsJunctions are identifiable by the attributes intersectingRoadNumber and junctionNumber. The attribute intersectingRoadNumber declare which linear elements coincide at the RdsJunction and the junctionNumber identifies this point in the network. Also RdsJunction can own the number of a crossing road (roadNumber). Every RdsPointLocation has a Area Reference to the hierarchy lowest Administrative Area and Linear Reference to which he belongs. Besides every RdsPointLocation knows his antecessor- and his follower- RdsPointLocation.

**Encodings**

The encodings are accomplished according to the Coding protocol for Radio Data System — Traffic Message Channel (RDS-TMC) using ALERT-C. The encodings are documented in the documentation of ISO 14819-1 (2003-06-01).

**Portrayal Catalogue**

There is no portrayal associated with this standard.
Service Reference

Service References exist specially for Traffic Massage Channel developed services. TMC Services are used to disspread traffic information using the Radio Data System. This intelligent transport service is also called and used for dynamic navigation.

2.1.5 Comparison and Evaluation

The survey of road network data in this chapter deliberately has concentrated on only a few definitions, each of world-wide or at least Europe-wide significance.

On the other hand this means that the survey has ignored a multitude of national and regional road network standards and de-facto standards. Matter-of-factly, these national and regional standards, which are usually employed by regional road and traffic management administrations carry useful data for eMOTION. It will soon become clear, why it is justified to ignore them.

The lead standard concerning road networks is clearly ISO 14825, better known as Geographic Data Files (GDF). The standard was originally drawn up by CEN in co-operation with digital map providers, automotive and electronic equipment manufacturers. The outcome of these standardisation efforts (CEN GDF 3.0) has formed the major input to the world standard ISO GDF 4.0, created by ISO/TC204.

ISO 14825 is the generally accepted basis for representing road network data in the navigation and location services market. High quality data following the GDF standard is available from major vendors in a nearly world-wide coverage. It constitutes the geographic data infrastructure for today’s commercial navigation and routing applications.

GDF data is, however, only seldom used for administrative tasks concerning the road network (e.g. maintenance of road infrastructure). This is the realm of national and regional road databases. GDF data is often the basis for digital maps used in traffic management systems. In this cases GDF data are often mapped with other data e.g. describing the technical infrastructure for traffic management.

The next standard analysed in this chapter is the network defined in ISO 19133, which itself is based on the network definition in ISO 19107. This standardisation was done by ISO/TC211.

ISO 19133 defines a conceptual model for Location Based Services (a genuine geo-spatial issue), part of which is necessarily navigation and thus a network, which allows routing operations for vehicles being performed.

Not surprisingly, the models of GDF and ISO 19133 are quite similar regarding the routable network core. Both build on a topological network of nodes and edges, which are called

- Junctions and RoadElements in the GDF case and
- NT_Junctions and NT_Links in ISO 19133.
In both models these network elements carry geometry and attributes qualifying the meaning and characteristics of the elements.

What makes the approaches still more similar is that the topological graphs are in both cases enriched by a second topology, which defines the usable vehicle routes in the nodes of the graph:

- GDF does this with a multivalued relation named Manoeuvre, which links enforced and forbidden RoadElement adjacent in a Junction.
- ISO 19133 couples usable routes through an NT_Junction with NT_Turns, which can be assembled to NT_Maneuvers to express enforced multi-turn connections.

Please note that the similarity of approaches is only true for the routable network definition. Except from this both standards are quite dissimilar, the main difference being that ISO 19133 is a conceptual model concentrating on the technical Location Based Services aspect, while GDF in general is a concrete vector map, which can also create well readable maps for humans in navigation systems.

It is interesting to see that the GDF standardization process is currently taking steps to get harmonized with the ISO/TC211 standards.

The EuroRoadS model is very different from the approaches pointed out above. It is the output of an EU funded project and is not an international standard.

The project created a harmonised road information model capable of representing European road databases. The modelling was done on the basis of the modelling framework given by the ISO 191xx standards. The goal was to make the data from all European road databases available in a uniform format from a single portal by the end of 2012. This harmonization effort is the reason why it is justified not to investigate those various national and regional road databases in the eMOTION context. EuroRoadS has already done this.

Since the road databases in Europe are actually rather different, the EuroRoadS model is abstract and tolerant. Networks may have explicit topology (which is then derived from the graph model in ISO 19107) or may just have geometry (ISO 19107 geometry). In this case topology is not represented and is considered implicit.

Since the model is based on ISO 19107, it is also compatible to the ISO 19133 network. However, usable routes in Junctions (turns and maneuvers) are not directly represented and have to be inherited from ISO 19133 to make the EuroRoadS network routable for vehicles.

GDF is also considered in EuroRoadS and can be represented with a few exceptions which concern the level 2 representation of GDF.

The last road network structure treated in this chapter is the one implied by RDS-TMC Location Code Tables, as specified by ISO 14819-3.

This network is very different from the the other ones analysed, because its structure is very much determined by its representation as a table. It does not define a topological structure comparable to the the other ones described here. The solution is, however, widely employed across Europe to provide dynamic traffic and travel information.
The basis of the Location Code Table are the Locations, the network-relevant ones of which are doubly-linked along roads by means of pointers. Locations can also carry coordinates, however, there is no geometry attached to the shape of the road between two adjacent Locations. In Junctions, which are special Locations, a switch to the linked list of other Locations can be made.

There is a European standard, called CentroMap (short for Central European Regional Telematics Road Map), which links the structure of the Location Code Table to a GDF network of RoadElements and Junctions. The aim of that standard is the support of visualisation processes for RDS-TMC messages.

See http://www.centromap.org/ for more information on the CentroMap definition. This eMOTION analysis does not treat this standard because road networks are not in focus and the essential features are already covered by GDF and RDS-TMC.

2.2 Public Transport Network Data

Public transport network data is static data comprising lines, line directions, stop points, connecting stations, both for rail and bus connections.

The depiction of the data models for transfers between public transport networks of different modes will not be treated here, but will be carried out in the next chapter 2.3 (Inter-modal Transport Network Data) because the mechanisms are general enough to allow for inter-modality/multi-modality.

2.2.1 ISO 14825: Geographic Data Files (GDF) - Public Transport

ISO14825:2004 / GDF describes a data model for representing real-world geographic phenomena on a conceptual level, a semantic level, and a physical level. An overview of GDF is available in the main document at subsection 3.5.7.

Feature Catalogue

The ISO14825 GDF standard has adopted the TRANSMODEL approach for representing public transport lines, stops, etc. The corresponding GDF feature theme provides for the following set of feature classes, corresponding to any properties that TRANSMODEL may store against such features in a Public Transport database.
These UML class entities can be considered interfaces between alternative views on public transport networks. Originally, within TRANSMODEL, these entities (together with many entities intentionally not reflected in GDF) were designed to model the domain semantics of public transportation. From a GDF point of view, designing a public transport model means modelling spatial, topological, and select operational properties (e.g., means of transport) of such networks, including interrelations with other feature themes, such as the road network data, and administrative and address information.

The commonality between TRANSMODEL and GDF is the set of corresponding entities (as shown in the above GDF UML figure) which represent the “infrastructure”, i.e. the actual physical transportation network in terms of rail tracks, bus stops, (real or virtual) connectivity, names and identifiers of such network elements, etc.
As GDF offers a rich base for describing geography, its public transport “extension” (in terms of the set of defined GDF feature classes, attribute types and relationship types) is the interface to connect with the rich domain standard, TRANSMODEL, and other standards compatible with it. In other words, TRANSMODEL properties described for, let’s say, a route link can be mapped to its spatial and topological rendering in GDF, which in turn is tied to other geographic data by means of (implicit) spatial and topological relationships and (explicit) semantic relationships.

In essence, the GDF model for public transport and TRANSMODEL are compatible by nature. Correspondence between the two has been achieved on the level of equivalent feature classes (using identical taxonomy) and equivalent inter-feature relations. The overlap between GDF and TRANSMODEL covers the area of interest for both which is the physical infrastructure of public transport networks and some basic properties thereof. While TRANSMODEL further extends the model towards full domain semantics, GDF further extends the model with infrastructure models for other transportation domains.
2.2.2 ENV 12896: Transmodel

Transmodel defines an extensive data model for relevant Public Transport related information. An overview of Transmodel is available in the main document at subsection 3.2.14. Within the various sub-domains included in Transmodel, the network description is a widely
adopted reference. More Transmodel sub-domains relevant for data in Public Transport applications and systems are described in subsections 2.8.1, 2.3.2 and 2.12.1.

Feature Catalogue

Items included in the feature catalogue take into account the structuring principles of the Transmodel abstract specification with a hierarchical, hereditary organization of data items that can be subtype from and inherit characteristics from other data items.

In a basic description, the transport network (road, railway, ...) is represented as a collection of POINTs and LINKs. The composite structure diagram below provides an overview of the main entities and their relations. The NETWORK VERSION entity is feature for handling different versions of the same transport network.

```
composite structure Points and Links

POINT
- Id
- name [0..1]

LOCATION
- coordinate_1
- coordinate_2
- coordinate_3 [0..1]

LOCATING SYSTEM
- name

LINE SHAPE
- formula

OPERATING DAY
- calendar
- date
- earliest time [0..1]
- latest time [0..1]

POINT ON LINK
- order
- distance from start [0..1]
- passing through 0..*

TYPE OF POINT
- name
- description [0..1]

TYPE OF LINK
- Id
- length [0..1]
- classified as 1..*
- locating 1
- located by 0..*
- located on 1
- classified by 0..*

NETWORK VERSION
- name
- valid for 0..1
- starts at 0..*
- ends at 0..*
- start of 0..*
- end of 0..*
- comprising 0..*
- classified 1..*
- classifying 0..*
- referring to 1
- a reference for 0..*
- from 1
- to 1
```
### POINT

**Specialises** --

**Description**
A 0-dimensional node of the network used for spatial description of the network. POINTs may be identified by a LOCATION in a given LOCATING SYSTEM. There may be different types of POINTs depending on the functional view of the network.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>A unique ID</td>
<td>static data, up-dated only when changes are made into the PT network</td>
</tr>
<tr>
<td>Name</td>
<td>A unique ‘name’ (char string)</td>
<td>static data</td>
</tr>
<tr>
<td>Valid-for</td>
<td>The ‘name’ or ID of a NETWORK VERSION</td>
<td>static data - Each POINT belongs to a particular version of the road network</td>
</tr>
<tr>
<td>Classified-as</td>
<td>TYPE OF POINT</td>
<td>static data - e.g. ROAD JUNCTION, STOP POINT, etc.</td>
</tr>
<tr>
<td>Located-by</td>
<td>a LOCATION</td>
<td>static data - Can be more than one (different location systems)</td>
</tr>
<tr>
<td>Start-of</td>
<td>One or more LINKs</td>
<td>static data</td>
</tr>
<tr>
<td>End-of</td>
<td>One or more LINKs</td>
<td>static data</td>
</tr>
</tbody>
</table>

### LINK

**Specialises** --

**Description**
An oriented spatial object of dimension 1 describing a connection between two POINTs. LINKs may be of different types depending on the particular functional view of the network.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>A unique ID</td>
<td>static data, up-dated only when changes are made into the PT network</td>
</tr>
<tr>
<td>Name</td>
<td>A unique ‘name’ (char string)</td>
<td>static data</td>
</tr>
<tr>
<td>Valid-for</td>
<td>The ‘name’ or ID of a NETWORK VERSION</td>
<td>static data - Each LINK belongs to a particular version of the road network</td>
</tr>
<tr>
<td>Classified-as</td>
<td>TYPE OF LINK</td>
<td>static data - e.g. ROAD ELEMENT, ROUTE LINK, etc.</td>
</tr>
<tr>
<td>Length</td>
<td>e.g. meters</td>
<td>static data</td>
</tr>
<tr>
<td>From</td>
<td>a POINT</td>
<td>static data</td>
</tr>
<tr>
<td><strong>To</strong></td>
<td><strong>a POINT</strong></td>
<td>static data</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Driving Direction</strong></td>
<td>A pair of POINTs</td>
<td>static data</td>
</tr>
<tr>
<td><strong>Passing Through</strong></td>
<td>a POINT ON LINK</td>
<td>static data - A POINT on the LINK used e.g. for purposes of public information, vehicle location, etc.</td>
</tr>
<tr>
<td><strong>Described by</strong></td>
<td>a LINE SHAPE</td>
<td>Static data, the shape for a LINK</td>
</tr>
</tbody>
</table>

TYPE OF POINT is introduced with the aim to describe common roles associated to POINTs. Each POINT can be of one or more types but in some cases, in presence of relevant TYPES of POINT a separate, dedicated entity is introduced. STOP POINT, TIMING POINT and ROUTE POINT are the most relevant cases.

Similar considerations can be made for LINKs and TYPE OF LINKs.

The main types of POINTs and LINKs that, for their relevance, are defined as a separate entity are:

- **STOP POINT**: a POINT where passengers can board or alight
- **TIMING POINT**: a POINT used to record timing information
- **ROUTE POINT**: a POINT defining the shape of a ROUTE in the network
- **SERVICE LINK**: LINK between a pair of STOP POINTs
- **TIMING LINK**: a pair of TIMING POINTs (run times are recorded)
- **ROUTE LINK**: a pair of ROUTE POINTs (defines a unique path through the network)
**Versions, validity and layers**

An entity NETWORK VERSION is introduced for modifications, production of different instances, existence of multiple sets of POINTs and other requirements for network data. These requirements may vary for a PT company depending on various factors. A validity property may then be assigned to a NETWORK VERSION for a specified period of time.

The most fundamental entity owned by a NETWORK VERSION is the POINT and the set of entities dependent on the (such as POINT LINK, ROUTE POINT, ROUTE LINK, TIMING POINT etc.) also implicitly depends on the NETWORK VERSION.

Several layers are defined when more functions operated in the company require different representations of the network. With this practice all the needs, constraints and interfaces of a system are defined in terms of layers (one layer for each system).
**Infrastructure Points and Links**

The infrastructure of the PT network is described by POINTs and LINKs. This is similar to Journey description.

Any POINT describing the infrastructure network is an INFRASTRUCTURE POINT, and the necessary LINKs are defined as INFRASTRUCTURE LINKs with a start and an end INFRASTRUCTURE POINT. This doesn’t give to the direction of the traffic flow, but is an arbitrary orientation. The optional attribute ‘driving direction’ can be used to specify such a direction, if needed.

---

**Link sequences**

A sequence of LINKs can define a path through the network. LINK SEQUENCE is an ordered sequence of LINKs.

A LINK SEQUENCE can be described in two ways:

- There is a unique link between its two end points (e.g a ROUTE LINK) : a LINK SEQUENCE can be described as an ordered sequence of POINTs, with LINKs in be-
There could be more than one LINK identified by its two end POINTs (e.g. a TIMING LINK) : a LINK SEQUENCE is defined as an ordered set of LINKs IN LINK SEQUENCE.

When several LINKs exists between two POINTs a LINK SEQUENCE is defined as an ordered set of LINKs IN LINK SEQUENCE. The order attribute specifies which LINK is used between two POINTs.

**Grouping and Stop Areas**

Objects like POINTs and LINKs can be grouped to form GROUP OF POINTS, GROUP OF LINKS etc. respectively for a particular functional purpose (PURPOSE OF GROUPING).
A relevant example is the STOP AREA. Several STOP POINTs can be grouped together under and referred to a common name. This can occur when different stops are close to each other for example at a railway station.

The generic concept of grouping POINTs can also arise in several other situations such as data modelling solutions to cover multi-modal operations.

**Network linear features**

Public transport routes are modelled as ordered sequences of POINTs, normally including STOP POINTs, ROAD JUNCTIONs or other PLACEs.

ROUTEs are defined by POINTs ON ROUTE, ROUTE POINTs and ROUTE LINKs elements. The following composite structure diagram shows the basic elements in the definition of ROUTEs:

Figure 2-13: Transmodel - Grouping
Note that ROUTE POINTs, defined to identify the path to be followed by services may be ends of a route or points expressing that a route is passing “via” this ROUTE POINT (more routes may exist between two ROUTE POINTs). The “via_flag” attribute expresses this func-

Figure 2-14: Transmodel – Basic elements in the definition of ROUTEs
Description
An ordered list of located POINTs defining one single path through the road (or rail) network. A ROUTE may pass through the same POINT more than once.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>A unique ID</td>
<td>Semistatic data; updated at each service planning phase (more often in demand-responsive schemes)</td>
</tr>
<tr>
<td>Name</td>
<td>A unique ‘name’ (char string)</td>
<td>see above</td>
</tr>
<tr>
<td>Valid-for</td>
<td>The ‘name’ or ID of a NETWORK VERSION</td>
<td>static data</td>
</tr>
<tr>
<td>Through</td>
<td>A set of POINTs ON ROUTE</td>
<td>The POINTs defining the ROUTE</td>
</tr>
<tr>
<td>Oriented by</td>
<td>A DIRECTION</td>
<td>Expresses the orientation for the ROUTE</td>
</tr>
</tbody>
</table>

**POINT ON ROUTE**

Description
A POINT used in the definition of a ROUTE through the network

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>A unique ID</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>A unique ‘name’ (char string)</td>
<td></td>
</tr>
<tr>
<td>Valid-for</td>
<td>The ‘name’ or ID of a NETWORK VERSION</td>
<td></td>
</tr>
<tr>
<td>On-route</td>
<td>a ROUTE</td>
<td></td>
</tr>
<tr>
<td>View-of</td>
<td>a ROUTE POINT</td>
<td></td>
</tr>
</tbody>
</table>

**ROUTE POINT**

Description
A POINT used in the definition of a ROUTE through the network.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>A unique ID</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>A unique ‘name’ (char string)</td>
<td></td>
</tr>
<tr>
<td>Valid-for</td>
<td>The ‘name’ or ID of a NETWORK VERSION</td>
<td>Each ROUTE POINT belongs to a particular version of the road network</td>
</tr>
<tr>
<td>Start-of</td>
<td>a ROUTE LINK</td>
<td>Can be several</td>
</tr>
</tbody>
</table>
End-of | a ROUTE LINK | Can be several
Via_flag | boolean | Expresses if a route is passing “via” that ROUTE POINT

ROUTE LINK

Specialises | LINK
Description | An oriented link between two ROUTE POINTs allowing the definition of a unique path through the network between those two route points.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>A unique ID</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>A unique ‘name’ (char string)</td>
<td></td>
</tr>
<tr>
<td>Valid-for</td>
<td>The ‘name’ or ID of a NETWORK VERSION</td>
<td>Each ROUTE LINK belongs to a particular version of the road network</td>
</tr>
<tr>
<td>Distance</td>
<td>A length in eg metres</td>
<td></td>
</tr>
<tr>
<td>From</td>
<td>a ROUTE POINT</td>
<td></td>
</tr>
<tr>
<td>To</td>
<td>a ROUTE POINT</td>
<td></td>
</tr>
<tr>
<td>Safely traversed by</td>
<td>A VEHICLE TYPE</td>
<td>Static data. The type of vehicle that can safely traverse the ROUTE LINK</td>
</tr>
</tbody>
</table>

DIRECTION

Specialises | --
Description | The general orientation of a ROUTE. May be used for passenger information and for scheduling or fare collection

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>A unique ID</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>A unique ‘name’ (char string)</td>
<td></td>
</tr>
<tr>
<td>For</td>
<td>ROUTE</td>
<td>The ROUTE the DIRECTION is referring to.</td>
</tr>
<tr>
<td>The opposite of</td>
<td>A DIRECTION</td>
<td>Two DIRECTIONs may be defined as being opposite to each other.</td>
</tr>
</tbody>
</table>

Lines

A LINE as it is known by the public by a similar name or number is a grouping of ROUTE{s}. LINE{s} may be grouped into GROUPs OF LINES for some purposes, such as vehicle or driver scheduling, merged timetables, fare collection etc. A PURPOSE OF GROUPING defines the unique purpose assigned for a GROUP OF LINES. A LINE may belong to more
than one GROUP OF LINES.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specialises</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>A grouping of ROUTEs that is generally known to the public by a similar name or number. These ROUTEs are usually very similar to each other from the topological point of view, being variants of a core route with some deviations on certain parts only. Often the vehicle journeys on these ROUTEs are scheduled jointly with tight synchronisation, in order to provide a regular service on this specific LINE. They are often grouped together for presentation of the timetable to the public.</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>A unique ID</td>
<td>Holds only for regular bus (or tram) services, not for DRT services</td>
</tr>
<tr>
<td>Name</td>
<td>A unique ‘name’ (char string)</td>
<td></td>
</tr>
<tr>
<td>Made up of</td>
<td>a ROUTE</td>
<td></td>
</tr>
<tr>
<td>Included in</td>
<td>a set of GROUP OF LINES</td>
<td>LINEs may be grouped into GROUPs OF LINES for particular purposes, eg fare collection, day type assignment, etc.</td>
</tr>
<tr>
<td>Main line for</td>
<td>a set of GROUP OF LINES</td>
<td>The LINE is the main one of a (set of) GROUP OF LINES</td>
</tr>
</tbody>
</table>
Encodings

Transmodel is a concept data model and does not mandate any particular implementation at the logical or physical level. Other Standards are based on Transmodel and uses specific encodings. For example TransXchange is XML based. For specific purposes, interfaces between topological network representation in the reference data model and other models may be defined. For example to allow data exchange with a Geographical Information System a
GDF interface has been defined.

An interface between the GDF data model and the topological network representation in the reference data model has been defined to allow data exchange with a Geographical Information System

**Portrayal Catalogue**

None

**Service Reference**

Data content defined and described in this standard is related with public transport information services. Transmodel has been used as a reference for SIRI (section 4.2.4) and JourneyWeb (section 4.5.1).

### 2.2.3 Railway Markup Language (RailML®)

RailML.org is a development partnership of independent businesses and institutions. One goal of RailML is to designate the totality of all XML schemas that have been written for railway data, including all the individual formats. An overview of RailML is available in the main document at subsection 3.2.10.

The intention of RailML is to give a data model for simulating several conditions of the railway system. Many elements are designed for simulation and visualisation purpose but can be used by train control systems.

**Feature Catalogue**

The current complete RailML Specification can be downloaded from the RailML Website: [http://www.railml.org](http://www.railml.org).

The RailML-Schema consists of three sub-schemas which are „infrastructure“ for infrastructure data, „rollingstock“ for rolling stock data, and „timetable“ for timetable data. For detailed information about rolling stock and timetable see subsection 2.8.4. This chapter outlines the rail infrastructure described by RailML.
The RailML Infrastructure-Schema consists of the two parts *operationControlPoints* and *lines*. The focus is on the *lines* element because this part describes the rail infrastructure network.

The infrastructure in RailML is specified by a line based network. The rail-network is based on the *line* elements. These elements describe the topology of the network. A *line* is a composition of several *tracks*, which can consist of several *trackElements*. The position of a *track*...
is defined by a kilometre-designation and can be further specified by additional attributes. 

*TrackElements* represent different sections of a *track* and are specified by a starting point as well as an end point which define the location and an implicit direction of a *trackElement*. The starting point and end point of a *trackElement* is an attribute which has a reference to the exact position which is described by the attribute *geoCoord*.

<table>
<thead>
<tr>
<th>geoCoord</th>
<th>Specialises</th>
<th>--</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Geographic coordinates</td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>long</td>
<td>decimal</td>
<td>Geographic longitude of the place [-180 .. 0 .. 180 degree]</td>
</tr>
<tr>
<td>lat</td>
<td>decimal</td>
<td>Geographic latitude of the place [-90 .. 0 .. 90 degree]</td>
</tr>
<tr>
<td>alt</td>
<td>decimal</td>
<td>Geographic altitude of the place [-1000 .. 10000 meters]</td>
</tr>
</tbody>
</table>

**Encodings**

The standard is based on XML schemas and uses XML documents for data exchange.

**Portrayal catalogue**

None

**Service reference**

unknown

### 2.2.4 Rail Journey Information System (RJIS)

The RJIS Datafeeds Interface Specification for Timetable Information describes the extraction format for exchange of data with the UK Rail Journey Information System. An overview of RJIS is available in the main document at subsection 3.2.9.

**Feature Catalogue**

The Master Station Names Record describes stations and other related locations within RJIS. Stations are located by OS Grid References.
Figure 2-18: RJIS Master Station Names Record

**Encodings**

Data is exchanged in CIF (Common Interchange Format) format, an ASCII file format of fixed length records.

**Portrayal catalogue**

There is no portrayal information associated with this standard

**Service reference**

There is no service associated with this standard.

### 2.2.5 National Public Transport Access Node Database (NaPTAN)

NaPTAN provides a unique identifier for every point of access to public transport in the UK, together with systematic meaningful text descriptions of the stop point and its location, allowing their unambiguous discovery and reference. An overview of NaPTAN together with NPTG is available in the main document at subsection 3.2.6.
StopPoint and StopArea provide the key classes for the NaPTAN schema.

A StopPoint represents a point of access to public transport, for any mode of travel – bus, rail, air, taxi, etc – so includes bus stops, stations, and ferry ports. The following table describes the principal Stop Point elements.

<table>
<thead>
<tr>
<th><strong>Stop Point</strong></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Specialises</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>The NPTGLocality represents the fundamental entity of the NPTG, and represents a city, suburb, town, district, village or other settlement.</td>
<td></td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td><strong>Type</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>ATCO Code</td>
<td>External ID</td>
<td>Unique code used by Association of Train Operating Companies (UK)</td>
</tr>
<tr>
<td>NaPTAN Code</td>
<td>A unique ID</td>
<td>Unique NaPTAN code.</td>
</tr>
</tbody>
</table>
### Stop Classification

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Three character code classifying the stop type. These include Off Street Types such as Station Entrances and Platforms, or On Street types such as Marked Bus Points, Hail and Ride sections or Taxi Ranks.</td>
</tr>
</tbody>
</table>

### Descriptor

<table>
<thead>
<tr>
<th>Code</th>
<th>Group of textual elements to systematically describe the stop. The descriptor is composed of: Common Name (compulsory), Short Common Name, Landmark, Street, Crossing, and Indicator.</th>
</tr>
</thead>
</table>

### Common Name

<table>
<thead>
<tr>
<th>String</th>
<th>Name by which stop commonly known</th>
</tr>
</thead>
</table>

### Short Common Name

<table>
<thead>
<tr>
<th>String[FIXED]</th>
<th>Short name for stop with length fixed by administrative area</th>
</tr>
</thead>
</table>

### Landmark

<table>
<thead>
<tr>
<th>String</th>
<th>Text describing adjacent landmark (e.g. building, destination, crossing street)</th>
</tr>
</thead>
</table>

### Street

<table>
<thead>
<tr>
<th>String</th>
<th>Name of Street where stop is</th>
</tr>
</thead>
</table>

### Crossing

<table>
<thead>
<tr>
<th>String (optional)</th>
<th>Nearest street crossing to the stop</th>
</tr>
</thead>
</table>

### Indicator

<table>
<thead>
<tr>
<th>String (optional)</th>
<th>Indicative description of relative position of stop to landmark (eg. outside, opposite)</th>
</tr>
</thead>
</table>

### Alternative Descriptors

<table>
<thead>
<tr>
<th>List</th>
<th>Optional link to alternative Descriptor instances used to describe alternative names or names in other languages.</th>
</tr>
</thead>
</table>

### Place element

<table>
<thead>
<tr>
<th>Place</th>
<th>Description of location (geocode as WGS84 or OS grid reference) and other locality information.</th>
</tr>
</thead>
</table>

### Adjacent NPTG localities

<table>
<thead>
<tr>
<th>List of references to Unique ID for NPTG localities</th>
<th>Optional NPTG localities close to the stop location where necessary to describe hail and ride operations</th>
</tr>
</thead>
</table>

### Stop Area

<table>
<thead>
<tr>
<th>Unique ID for Stop Area</th>
<th>The assigned area for this Stop Point. The Stop Point may belong to several hierarchical Stop Areas through the Stop Area's Parent Stop Area.</th>
</tr>
</thead>
</table>

---

**Place**

<table>
<thead>
<tr>
<th>Specialises</th>
<th>--</th>
</tr>
</thead>
</table>

**Description**

Describes where a Stop Point is, and associates it with an NPTG Locality

**Property**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>

**NPTG Locality**

Reference to Unique ID for NPTG locality

Used to assign the most specific relevant NPTG locality

**Alternative NPTG Localities**

List of NPTG locality Unique IDs

Optionally describes nearby NPTG localities. Example of use is given for hail and ride services where footprint extends of several localities

**Main NPTG Localities**

List of NPTG locality Unique IDs

Optionally describes nearby NPTG localities which this stop place serves as a main Stop Place (e.g. for a main railway station outside the city limits)

**Town**

String (optional)

To be used where town differs from that in the NPTG Locality
A Stop Area represents a group of related Stop Points

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop Area Code</td>
<td>A unique ID</td>
<td>Unique Identifier for the Stop Area</td>
</tr>
<tr>
<td>Stop Area Name</td>
<td>String</td>
<td>The name of the stop area</td>
</tr>
<tr>
<td>Stop Area Type</td>
<td>Type Code Enumeration</td>
<td>Four character code classifying the Area Type. Area Types include Coach Station, Rail Station, Airport and On Street Stop Cluster</td>
</tr>
<tr>
<td>Location</td>
<td>Location</td>
<td>WGS84 or OS grid reference</td>
</tr>
<tr>
<td>Alternative Names</td>
<td>List</td>
<td>Used to describe alternative names or names in other languages</td>
</tr>
<tr>
<td>Parent Stop Area</td>
<td>Reference to Unique ID of Stop Area</td>
<td>Used to describe the Stop Area which this Stop Area is a part of.</td>
</tr>
</tbody>
</table>

**Encodings**

Both NPTG and NaPTAN data are exchanged using XML documents complying with the NPTG and NaPTAN Schema, or alternatively using a NPTG and NaPTAN CSV formats respectively.

**Portrayal Catalogue**

N/A

**Service Reference**

JourneyWeb is built upon NaPTAN and NPTG standards. TransXChange also shares common elements.

**2.2.6 Identification of Fixed Objects in Public Transport (IFOPT)**

IFOPT defines a model and identification principles for the main fixed objects related to public access to Public Transport (e.g. stop points, stop areas, stations, connection links, en-
trances, etc.). An overview of the standard is available in the main document at subsection 3.2.4.

**Feature Catalogue**

**STOP PLACE MODEL**

The Stop Place Model describes the structure of the Stop Place (from complex interchanges such as stations and airports, down to isolated bus stops) describing in detail place features and the paths between them. Stop Place is separate from Transmodel's logical Stop Points and Connection Links, but facility exists to link these via assignment models (as shown in Figure 2-22).
Figure 2-20: IFOPT - Stop Place Basic Model

Figure 2-20 describes the essential Stop Place Model. A Stop Place consists of Stop Place Spaces: Quays (areas where passengers have access to PT vehicles, such as platforms and on-street bus stops); Access Spaces (where they do not, such as entrance halls), and Boarding Positions. Quays and Access Spaces may be recursively specified, while Boarding Positions must be specified within Quays. Stop Places may be assigned a level, corresponding to floors within buildings. Path Links describe links between Access Spaces and Quays, and may be assigned from and to levels, which should correspond to those of the related Stop Place Spaces. Stop Places may be associated with transport modes in 2 ways: the ‘transport for’ relationship describes the transport mode the Stop Place is for, e.g. a railway station would be, minimally, associated the mode train; the ‘access by’ relationship describes Modes which maybe used within the Stop Place, which will normally be pedestrian.
Location information is specified in the parent Place (conforming to the Transmodel concept of Place), where Point, Link and Zone projections may be specified in WGS84 or another GML compatible manner. There is also the possibility to associate Place with a postal Address.

Stops may be identified in 3 different ways, namely: (i) a machine readable Unique System identifier, which may be globally unique; (ii) Public labels, recognisable by humans; and (iii) Public Codes, such as stop code suitable for SMS next departure services. Aliases are supported where multiple names may be associated with Stop Places. The most basic stop information is shown in Figure 2-21.

Figure 2-22 describes the relationship of Stop Place Elements to timetabled Stop Points in a Vehicle Journey (both of which are described in Transmodel). The main Stop Place components may be related to the main Scheduled Stop elements using relevant Assignment elements.

- The Passenger Stop Point Assignment allows association of Scheduled Stop Point with a specific Boarding Position, Quay or Stop Place.
- The Connection Link Assignment allows the association of Connection Link with one or more specific Navigation Paths (an ordered collection of Path Links), or a Stop Place or Quay to indicate average connection times.

© eMOTION Consortium
• The Dynamic Stop Point Assignment allows the dynamic association of Scheduled Stop Point with a Quay or Boarding Point, or definition of those available for dynamic assignment.

• The Train Stop Point Assignment allows association of a specific Boarding Position with a specific Train Element or Train Block Part, which is required for fixed seating operations and certain Coupled Journey operations (where trains might split).

The Stop Place model also includes detailed information on vehicle stopping places and vehicle stopping place assignment.

The principal structural components of a stop place are spaces, entrances and paths. Stop Place Components include: associations with Levels and Check Points Hazards; elements to describe their projection inherited from Place; validity conditions; references to tariff zones; association with transport modes and mobility hazards.
Figure 2-23: IFOPT - Stop Place Components Type Hierarchy

Figure 2-23 describes the Stop Place Component Type Hierarchy, while Figure 2-24 describes the Stop Place components containment within a Stop Place.

Figure 2-24: IFOPT - Stop Place Components Containment Hierarchy

ADMINISTRATIVE MODEL

This is an organisational model to support provision and maintenance of data items within
the other models, given the many potential stakeholders administering information. Essentially concerned with data provision and audit, it is thought to be beyond the scope of this section of eMotion.

**Encodings**

The fixed object model may be represented as an XML schema.

**Portrayal Catalogue**

There is no portrayal associated with this standard.

**Service Reference**

Not associated to a service.

**2.2.7 Comparison and Evaluation**

From the passenger point of view, standard public transport services travel between nodes. These nodes are connected by links, over which public transport vehicles travel. Nodes and Links describe the public transport network. Of relevance to the end user are Stop Points, places on the public transport network where travellers may gain access to public transport services of a particular mode. Traditionally such Stop Points are associated with nodes on the network. It should however be noted that certain types of service may stop at many places along a particular link (such as in Hail and Ride style operations), or anywhere on the network within a particular area (for certain Demand Responsive Transport operations). Stop Points need to be identifiable to travellers, so apart from a Unique System Identifier, Stop Points may be referred to by one or more public names. These public names might refer to the settlement, street, or POI with which the stop is typically associated. Stop Points should also carry information such that their location is space may be determined, and that they might be projected onto a map, and so that their physical proximity with relation to other Stops and Features may be calculated. The most common method is some absolute coordinate reference.

Transmodel provides an abstract model for public transport systems. The following standards are either based on the Transmodel, are designed to work with Transmodel, or have been revised to adopt Transmodel: ISO14825 GDF (explicitly adopts Transmodel); NaPTAN (revised to adopt Transmodel terminology, addresses specific differences); and IFOPT (explicitly references Transmodel entities, and adds refinements). IFOPT appears to have drawn on NaPTAN, and explicitly states that it draws heavily on the French CERTU study, Étude des systèmes de localisation pour les transports – Clarification des concepts liés aux arrêts de transports en commun’.

IFOPT’s Stop Place model offers the most detailed information on Stop Places, explicitly separating Transmodel ’s Stop Point from the assigned Stop Place, which details features such as Quays, Access Spaces, links between these, Stop Place Entrances and links be-
tween these entrances. Information relating to facilities, equipment (eg. ticket machines) and accessibility issues is also explicitly included. Within NaPTAN, Stop Place Entrances are treated as Stop Points.

Both Transmodel and GDF describe the network links between nodes, to which public transport routes may be assigned. GDF also describes such links projection onto road elements. This might be contrasted with the TransXChange (presented later) capability to map links onto points on a path.

In summary, the essential features from Transmodel may be summarised as follows:

- The transport network (road, railway, ...) is described as a collection of POINTs and LINKs.
- A POINT is located by a LOCATION using a LOCATION SYSTEM. A POINT can be of several TYPES and certain types of POINT (such as STOP POINTS, TIMING POINT and ROUTE POINT) are represented by a separate entity.
- Objects like POINTs and LINKs can be grouped to form GROUP OF POINTS and GROUP OF LINKS respectively.
- Public transport routes are modelled as ordered sequences of POINTs, normally including STOP POINTs.

2.3 Inter-modal Transport Network Data

The static network data required for inter-modal transport and transport planning requires linking networks of different transportation modes, which is road networks and public transport networks. The various modes of transportation include walking, automobile, park-and-ride, mass transit and taxi, for example.

The usage of the terms “inter-modal” and “multi-modal” is often not very clear. The terms are often used interchangeably, however, “multi-modal” seems to be the more restricted term. A “multi-modal” trip planner is one that can plan trips for car and public transport, but cannot combine these.

We will here use the term “inter-modal” to express that we want to combine modes, even though some of the visited standards might be restricted to public transport modes only.

The linking of networks can be done by constructing “transfer-nodes”, which allow passengers to change from one network to the next. Transfer-nodes resemble a “generalised junction” in that they allow “transfers” in a way “turns” are allowed in a street network junction. These transfer objects (in other standards called “connection links”) form the glue between the networks of different modes.

In this section we essentially compare the ISO 19133/19134 multi-modal (inter-modal) network to the “network gluing” features of several public transport standards.

2.3.1 ISO 19134 - Chapter 6: Multimodal Network

ISO 19134:2007 describes a model for multi-modal (in the sense of inter-modal) routing and
navigation. An overview of the standard is available in the main document at subsection 3.4.6.

To comply with the structure of this eMOTION document, this text only refers to the “Multi-modal Network” (Chapter 6), which describes the static (network parts) of the definition. Other parts of the ISO 19134 standard are explained in section 2.13 (Data For Inter-Modal Journey Planning).

See subsection 2.1.2 for ISO 19133.

**Feature Catalogue**

The following UML diagram is an excerpt of ISO 19133 package “Combined Networks”. It defines the general means of creating larger networks by joining smaller ones.

![UML Diagram](image_url)

*Figure 2-25: ISO 19133 - package “Combined Networks”*

The type `NT_CombinedNetwork` models networks built from smaller component networks as defined in 2.1.2 through the use of transfer links and transfer nodes. If the component networks are all of the same mode, the combined network will be single mode. If not, the transfers would be mode changes within the combined multi-modal network.

The combined network represented by `NT_CombinedNetwork` is itself a `NT_Network` as by means of specialization. On the other hand it is a container for any number of those networks, the `componentNetworks` of the `NT_CombinedNetwork`.

The linking of the various networks is done by re-using the types of an `NT_Network`. The basic concept is to introduce so-called `NT_TransferNodes`, which aggregate `NT_Junctions` of the constituent networks, themselves being `NT_Junctions` by specialization. Its links and turns contain all the links and turns of the component junctions plus transfers that move from a link in one component network to a link in one of the other component networks. In terms of classes: `NT_TransferNodes` add `NT_Transfer` objects to the `NT_Junctions` assembled in a `NT_TransferNode`. These `NT_Transfer` object are derived from `NT_Turn`, i.e. they represent
the required additional connection structure of a *NT_TransferNode*.

Besides the *NT_TransferNode* objects the constituent networks can also be glued together by means of additional *NT_TransferLinks*, which are derived from *NT_Link*. They can add additional links between any of the *NT_Junctions* involved in the union of all networks. This means they represent links, whose boundary nodes are in different component networks.

The following UML diagram from ISO 19134 employs the combined network structures from ISO 19133 to set up a useful multi-modal (inter-model) network.

![Figure 2-26: ISO 19134 - multi-modal network](image-url)
The standard picks up the combined network types \textit{NT\_CombinedNetwork}, \textit{NT\_TransferNode}, \textit{NT\_Transfer}, and \textit{NT\_TransferLink} to define multi-modal variants of these, namely: \textit{MM\_MultimodalNetwork}, \textit{MM\_TransferNode}, \textit{MM\_Transfer}, and \textit{MM\_TransferLink}. The inherited associations between these types are overridden by ones, which are constrained to the derived types.

The \texttt{disabledAccessible} Boolean attribute is added to \textit{MM\_Transfer} and \textit{MM\_TransferNode}. The \texttt{junctionType} attribute of \textit{MM\_TransferNode} indirectly inherited from \textit{NT\_Junction}, is extended to contain additional codelist items. Inherited are:

\begin{verbatim}
intersection, roundabout, enclosedTrafficArea, exitRamp, entranceRamp, changeOver, boardingRamp, station, transfer
\end{verbatim}

Added items are:

\begin{verbatim}
bustop, taxiStop, subwayStation, lighttrailStation, railwayStation, interCityBusTerminal, platform, gate, port, airport, parkAndRide, publicParkingLot, privateParkingLot.
\end{verbatim}

This is the combining structure. Each contained \textit{MM\_SingleModeNetwork} shows up in a slightly modified version compared to ISO 19133 by deriving \textit{MM\_SingleModeNetwork}, \textit{MM\_SingleModeLink}, \textit{MM\_SingleModeTurn}, \textit{MM\_SingleModeJunction} from the respective ISO 19133 types: \textit{NT\_Network}, \textit{NT\_Link}, \textit{NT\_Turn}, \textit{NT\_Junction}.

\textit{MM\_SingleModeNetwork} receives an additional \texttt{mode} attribute, which is from the following codelist:

\begin{verbatim}
IntracityBus, intercityBus, shuttleBus, rovingBus, subway, lightrail, tram, railway, highSpeed-Railway, ferry, airway, walk, bike, taxi.
\end{verbatim}

\textit{MM\_SingleModeLinks} takes a multiple attribute named \texttt{routeSegmentCategory} which uses the same codelist.

\section*{Encodings}

There is no encoding associated with this standard.

\section*{Portrayal Catalogue}

There is no portrayal associated with this standard.

\section*{Service Reference}

There is no specific service definition associated with this standard. It is, however, intended that this standard is to be used by Location Based Services as defined in ISO 19133 / ISO 19134.

\subsection*{2.3.2 ENV 12896: Transmodel - “Connection Links”}

Transmodel defines an extensive data model for relevant Public Transport related information. An overview of Transmodel is available in the main document at subsection 3.2.14.

As stated above here only the mechanisms contained in Transmodel which serve to transfer
passengers from one stop point of a public network to another are presented.

**Feature Catalogue**

*Connection Links*

The entity CONNECTION LINK provides description of the connection to proceed from a specific STOP POINT to another, when an interchange is necessary.

This is of particular relevance for a trip calculation and optimization (see 2.12.1) when, due to interchanges operation, a passenger must move from an arrival point to the next departure point. When this happens on different STOP POINTs (already defined in sect 2.2.2) and these STOP POINTS are suitable for such an interchange, a CONNECTION LINK is introduced to link these points.

The main attributes for a CONNECTION LINK are: a distance defined for the connection, the duration to cover the distance (a default value or different values for different types of transports), and possible constraints for users.

For a more complete description of this entity and its relation with Trip and Trip Calculation see section 2.12.1.

**Encodings**

Transmodel is a concept data model and does not mandate any particular implementation at the logical or physical level.

**Portrayal Catalogue**

None

**Service Reference**

Data content defined and described in this standard is related with public transport information services.

**2.3.3 Identification of Fixed Objects in Public Transport (IFOPT) - “Connection Links”**

IFOPT defines a model and identification principles for the main fixed objects related to public access to Public Transport (e.g. stop points, stop areas, stations, connection links, entrances, etc.). An overview of the standard is available in the main document at subsection 3.2.4.

In this place we only view the mechanisms of IFOPT which serve to transfer passengers from one stop point of a public network to another.

**Feature Catalogue**

*Connection Links*
The ConnectionLink class provides description of the connection to proceed from a specific StopPoint to another, when an interchange is necessary.

The IFOPT Navigation Path Model provides a more detailed model which fits over the Transmodel’s STOP POINTS and CONNECTION LINKS, see 2.3.2. Navigation Paths consist of Stop Place Spaces and Path Links within a Stop Place, while Access Path Links describe links between separate Stop Places. Links may optionally describe links to specific Stop Place Space Entrances. Mobility Hazards may be attached to Navigation Path elements to enable journey planners to make use of extra Navigation Paths provided between Stop Place Spaces. An example navigation path is provided in the next figure.

See Figure 2-22 in section 2.2.6 to see the classes of the example in their context, particularly their relations to the IFOPT network model.

![Figure 2-27: IFOPT - Example Navigation Paths possible with IFOPT within and between Stop Places](image)

**Encodings**

The fixed object model may be represented as an XML schema.

**Portrayal Catalogue**

There is no portrayal associated with this standard.

**Service Reference**

Not associated to a service.

### 2.3.4 Rail Journey Information System (RJIS) - Fixed Leg

The RJIS Datafeeds Interface Specification for Timetable Information describes the extraction format for exchange of data within the UK Rail Journey Information System. An overview of RJIS is available in the main document at subsection 3.2.9.
Feature Catalogue

<table>
<thead>
<tr>
<th>Fixed Leg</th>
</tr>
</thead>
<tbody>
<tr>
<td>+Link method</td>
</tr>
<tr>
<td>+Leg Origin: CRS Code</td>
</tr>
<tr>
<td>+Leg Destination: CRS Code</td>
</tr>
<tr>
<td>+Leg Time</td>
</tr>
</tbody>
</table>

*Figure 2-28: RJIS Fixed Leg*

The Fixed Leg record within the Timetable Information feed is used to describe links between public transport access points without fixed timetable information, by link methods such as walk legs or regular bus, metro, ferry, etc.

### 2.3.5 Comparison and Evaluation

Under the headline “Inter-modal Transport Networks” this section compares the ISO 19133/19134 “multi-modal” network definition to the “network gluing” features of several transport standards, particularly Transmodel and IFOPT.

The ISO multi-modal network model re-uses the definition of a network graph of *junctions (nodes)* and *links (edges)*, which is explained in 2.1.2. This original network uses objects named *turns* to describe the allowed connections between the adjacent links meeting in an *junction*.

Such a network also describes a public transport network, if we understand the *junctions* as *stop points*. The *turns* are then interpreted as *transfers* – however, recall that the *turns* connect in-coming and out-going *links*. This is usually too much flexibility, because any *link* ending in a *stop point* is usually connected to any other starting in the *stop point* because the transfers are executed by pedestrians.

The “trick” in constructing a combined network out of several single ones is done in ISO 19133 by aggregating *junctions* of different networks to so-called *transfer nodes* and by adding additional *turns*, which are then explicitly called *transfers* and which describe the connectivity in these *transfer nodes*. Again, *transfers* connect *links*, which might deliver more flexibility than is needed in practice.

ISO 19134 only adds a few useful attributes to this setting, like the mode of a single mode network, and accessibility to disabled persons, etc.

As already hinted above, “real” public transport standards, particularly Transmodel and IFOPT, which has adopted and extended Transmodel’s concepts (but also RIJS), have a much simpler basic model. They simply connect *stop points* by means of *connection links*, which means that all links ending in one *stop point* are effectively connected to each link emerging from the other.

The IFOPT Navigation Model includes information to accessibility of the connection links (if step free, wheelchair accessible), presence of check points and delays, as well as timing
estimates for Frequent, Occasional and Mobility Restricted travellers. A Mode for the Connection Link might also be specified, although this will almost always be Walk. Path Links describe links between platforms, access spaces and entrances to the stop place.

For eMOTION it has to be decided, whether the simplified connection model of Transmodel is sufficient. It is not, if for example, individual costs have to be attached to the transfers in the ISO sense or if different acceptability for disabled persons has to be included on this individual link-by-link basis.

2.4 Location Referencing

Location referencing stands for various object models employed for describing location. These objects usually are auxiliary data structures, destined for determining the location of features proper, but this is not necessarily so.

Location can naturally be directly given by geometry. The latter can be a point, a line, or an area, depending of the nature of the feature whose location is to be described.

Another common method of specifying location is by street addresses or street address ranges. Alternatively, geographical names, or the names of so-called Points-of-Interest, are also a way to specify location.

In connection with road networks, the method of linear referencing is often found to specify point or line locations. Linear referencing describes a location on an edge of the road network by addressing that edge and the length of the arc leading from the starting end of the edge to the location.

2.4.1 DATEX 2

The main aim of DATEX 2 is to establish standardised links between Road Operators, TICs and Service Providers to allow the easy and efficient communication of traffic and traveller Information. An overview of the standard is available in the main document at subsection 3.1.2.

One part of DATEX 2 is the description of the location the information is provided for. DATEX 2 uses different established methods for location referencing like Alert-C, TPEG-LOC or localisation by coordinates.

Feature Catalogue

DATEX 2 provides different options to define the location of traffic messages or traffic data. The most important features are:

- information can be located with one group of locations where each location must be physically different,
- each individual location can be defined with several location referencing systems like Alert-C, TPEG-Loc or xy-coordinates
• the group of locations and each location can be expressed by references defined in a so called *PredefinedLocationsPublication*,

• A location can correspond to the road network (point or linear) or to an area,

• A group of locations can relate to several destinations, each being a point or an area,

• When the location is on the network, supplementary positional description information can be given (e.g. characteristics at the lane level, …)

For point locations the following location referencing systems can be used:

• Point by coordinates (ETRS89 / WGS84)

• Reference Point: Reference Point identifier + details such as road name and direction (same coding as Alert-C for direction).

• Alert C Point (method 2 or 4)

• TPEG Point Location

For linear locations the following location reference systems can be used:

• Reference Point linear location (reference point for primary point plus reference point for secondary point – not shown in figure Figure 2-31)

• Alert C linear

• TPEG Linear Location

For area locations the following location referencing systems can be used:

• Alert C area

• TPEG area location

The general data model for locations in DATEX 2 is based on *NetworkLocation, AreaLocation* and *GroupOfLocations*. In addition it is possible to use references to predefined locations.

*NetworkLocation* is a specification of point or linear location on the road network.

An *AreaLocation* is a geographic or geometric defined area which may be qualified by height information to provide additional geospatial discrimination (e.g. for snow in an area but only above a certain altitude).

*GroupOfLocations* represent a group of one or more physically separate locations. Locations maybe related, as in an itinerary or route, or may be unrelated. It is not for identifying the same physical location using different referencing systems.
Figure 2-29: DATEX 2 - Location : compositestructure diagram

The following UML diagrams show the specification of point, linear and area location.
Figure 2-30: DATEX 2 - Point : compositestructure diagram

Figure 2-31: DATEX 2 - Linear : compositestructure diagram
With DATEX 2 it is possible to define “predefined location”. The `PredefinedLocationsPublication` is a collection of data providing a definition of one particular defined group of sections of the network. With this feature it is possible to define features such as specific routes or parts of a network.
2.4.2 ISO 14819: RDS-TMC - Chapter 3: Location referencing for ALERT-C

RDS-TMC (Radio Data System – Traffic message Channel) is a Europe wide standard for transferring traffic relevant information into mobile devices. An overview of the standard is available in the main document at subsection 3.1.7.

This subsection focuses on additional information regarding location referencing. The location referencing rules defined in ISO 14819-3:2004 address the specific requirements of Traf-
fic Message Channel (TMC) systems. This specific standard can be obtained via:


For the location reference table see 2.1.4, for the general message structure see 2.6.3.

**Feature Catalogue**

The Structure of a RDS-TMC location reference consists of three attributes - countryCode, tableNumber and locationCode.

Country code identifies the agency responsible for location coding, which defined the location table and its number. All location tables belong to the Country Code in the range 1-16 mentioned in EN50067 of CENELEC. Within any particular country code, each location table has one, unique number in the range 1-63. Each country code is shared by more than one country within Europe and the surrounding territories. To avoid ambiguity in European RDS-TMC location referencing, ranges of location table numbers are allocated to specific countries.

The combination of country code (4 bits), location table number (6 bits) and location code (16 bits) defines an extended location code, which is unique throughout Europe.

![Figure 2-34: RDS-TMC - Location Reference](image-url)
Primary and secondary locations

Many location references in ATT applications extend geographically through several adjacent sections of a road. The concept of primary and secondary locations is used to indicate the extremities of the affected sections, without having to list all the intervening places. For situation locations, by convention, the primary location is taken to mean the end where the cause of the problem can be found, whenever a cause can be pin-pointed geographically (see figure below).

![Figure 2-35: ISO 14819-3 - Primary and Secondary Locations](image)

Primary location indicates the origin of the situation (e.g. the accident site). Where situations have no distinctive origin, either end of the location can be designated the primary location. The secondary location is at the other end of the affected length or area.

**Direction and extent code in RDS-TMC**

In many cases events affecting road traffic cover a number of locations, such as where an accident results in long tail backs. These occurrences are defined by addressing the location of the accident as the primary location, and then identifying the end of the tailback by using the direction and extent fields of the standard ALERT-C message.

Extent is the number of steps along the road, from the pre-defined primary location, through
other pre-defined locations, to reach the pre-defined secondary location. Extent data comprise a sign (+ or -) and a number of steps. In distance marker (kilometre/milepost reference) systems, extent can be used in a similar way, except that in this case, it is measured in kilometres.

This field consists of four bits of information: 1 direction bit and 3 extent bits. The direction bit indicates the direction of queue growth, not the direction of traffic flow affected by the event. The interpretation of the direction bit is as follows:

**Direction bit = 0 (positive)**
Direction of queue growth is in the positive road direction, traffic driving in the negative road direction is affected, the positive offset column is to be used to indicate the extent.

**Direction bit = 1 (negative)**
Direction of queue growth is in the negative road direction, traffic driving in the positive road direction is affected, the negative offset column is to be used to indicate the extent.

The extent bits identify the number of locations along the road that are affected by the problem with a maximum of 8 (primary location and 7 related locations). An extent of 1 would identify the secondary location (the end of the event's extent) as being the next location along the road from the primary location. An extent of 3 would force the receiver to search the database for the third location along the road from the primary location.

**Encodings**
The encoding of RDS-TMC messages is described in chapter 2.6.3.

**Portrayal Catalogue**
There is no portrayal associated with this standard.

**Service Reference**
Data content defined and described in this standard is in liaison with the RDS-TMC traffic and traveller information service. This intelligent transport service is also called and used for dynamic navigation.

**2.4.3 ISO 19133 - Section 6.6: Package Linear Reference Systems**
ISO 19133:2005 describes the data types, and operations associated with those types, for the implementation of tracking and navigation services. An overview of the standard is available in the main document at subsection 3.4.5.

The text in this eMOTION standards description section refers to ISO 19133:2005, Section 6.6: Package Linear Reference Systems.

**Feature Catalogue**
The following is a slightly simplified excerpt from ISO 19133.
Linear referencing emanates from the type \textit{LR\_PositionExpression}, which essentially combines the reference to one \textit{LR\_Element} (in the role of a \textit{referenceDomain}) with a \textit{Measure}. \textit{LR\_Element} stands for some curvilinear element on the geometry of which linear referencing is to be applied. The \textit{Measure} tells us how far from the beginning of the linear feature the referenced position is supposed to lie.

The rest of the object model enhances this simple schema in a couple of ways.

There is first the optional reference to a \textit{LR\_LinearReferenceMethod} object. It describes the manner in which measurements are made along (and optionally laterally offset from) a curvilinear element. The \textit{LR\_LinearReferenceMethod} is described by \textit{name} and \textit{type} (both strings), the \textit{units} of measurement along the curvilinear element, the \textit{offsetUnits} of measurement laterally to the element, and the \textit{positiveOffsetDirection}, which stipulates whether positive \textit{offsets} measure to the \textit{right} (default) or to the \textit{left}.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{iso_19133_linear_referencing.png}
\caption{ISO 19133 - Linear Referencing}
\end{figure}
The offset itself can optionally be attached to a *LR_PositionExpression* by means of the role *offset* and is expressed by a *LR_OffsetExpression* object. It specifies the *offset* and an enumerated *offsetReference*.

A somewhat complex part of the Linear Reference model is given by the concept of *LR_ReferenceMarker*, which stands for a known position on the curvilinear element, from where the measure is taken for the linear reference method. Consequently, a *LR_PositionExpression* may optionally refer to such a *LR_ReferenceMarker*. In case such a reference is not given, measurement takes place from the start.

One or more *LR_ReferenceMarkers* belongs to a *LR_LinearReferenceMethod* via role *marker*. The *LR_ReferenceMarkers* are also associated with the linear element (role *datum-Markers* of *LR_Element*), where they appear as an ordered aggregation of markers along this element. The ordering of the markers is consistent with the order in which the markers would be found in traversing the *LR_Element* from the beginning to the end.

**Encodings**

There is no encoding related to this standard.

**Portrayal Catalogue**

None associated.

**Service Reference**

None.

### 2.4.4 ISO 19133 - Sections 6.2.2 & 8.2: Position and Address Package

ISO 19133:2005 describes the data types, and operations associated with those types, for the implementation of tracking and navigation services. An overview of the standard is available in the main document at subsection 3.4.5.

The text in this eMOTION standards description section refers to ISO 19133:2005, Section 6.2.2, which contains a definition of “Position” for tracking purposes and Section 8.2: Package Address.

**Feature Catalogue**

ISO 19133 defines a generalised type for representing positions in tracking and associated applications. An instance of that type locates a position or place within the network.

*TK_Position* is a “union-type” (which is presented here in generic UML syntax without using the “union” stereotype of the original definition in the standard).
A TK_Position type instance possesses exactly one of the associated attributes of different types:

- **directPosition**: The position is given directly by the means of a DirectPosition object, which represents a point given by its coordinates.

- **placeName**: The position is defined by an SI_LocationInstance object. It is based on a place name as described in the ISO Gazetteer model (ISO 19112), which is not treated in this analysis.

- **featureID**: The position is given by the identifier of a feature (FD_FeatureName), which is informatively depicted in ISO 19133, chapter 10.

- **linearReference**: The position is defined by linear referencing object, LR_PositionExpression, as described in 2.4.3.

- **networkPosition**: The position is given by an NT_NetworkPosition. This construct is described in the context of the NT_Route construct in 2.11.1.

- **address**: The position is defined by an abstract AD_AbstractAddress object, which is described below.

- **phone**: The position is defined by the phone number of a mobile user.

We will now have a closer look on the AD_AbstractAddress object definition.

AD_AbstractAddress is an abstract type with no attributes. The idea behind this construct is to derive country specific concrete address types from the abstract type, which make use of arbitrary AD_AddressElements for describing addresses complying to the conventions of a country.

The standards gives one realisation of these intended country specific addresses, called AD_USAddress. (We do not show this informative definition in the following class diagram.)

The standard also defines a generic concrete address type, called AD_Address. It is con-
structured in a way that will fit the information content of any instance of the abstract class.

An `AD_Address` instance optionally contains an `addressee`, which is realised by an `AD_Addressee` object, any number of `addressElements`, which can contain any concrete object derived from `AD_AddressElement`, and finally any number of `phoneNumbers`, realised by `AD_PhoneNumber`.

The list of concrete address elements is as follows:

- `AD_Addressee` contains the name of the `addressee`.
- `AD_Street` describes a street given by (all optional) `directionalPrefix`, `typePrefix`, `officialName`, `typeSuffix`, `directionalSuffix`, `muniQuadrant`, and `postalCode`. Most of these are typical for US addressing conventions, but since everything is optional, European streets may also be described this way. The `trailingSpaces` flag indicates whether the street name contains trailing spaces.

Figure 2-38: ISO 19133 - Address Model

© eMOTION Consortium
**AD_StreetIntersection** describes an address by means of the intersection of **AD_Streets**.

**AD_PostalCode** contains a postal *code* and optionally an *addonCode*.

**AD_StreetLocation** is the root element for any address element that specifies a position on a street. A number, as can be specified in its concrete derivative **AD_Building**, is the most common pattern. However, other positioning data definitions may be added. **AD_Building** lets one describe the location by *number* or *buildingName* or *subdivision*, which is specified by a **AD_NamedPlace** object.

**AD_StreetAddress** is an address element containing both a *street* (**AD_Street**) and *location* (**AD_StreetLocation**).

**AD_NamedPlace** contains a place *name* as might be found on a gazetteer, and optionally a *code* for *regionOrCountry*, a *type* classification of the place and a *level*, which the depth in place hierarchy.

**AD_PhoneNumber** finally contains a phone *number*.

**Encodings**
There is no encoding related to this standard.

**Portrayal Catalogue**
None associated.

**Service Reference**
None.

### 2.4.5 OGC OpenLS - Core Services, Abstract Data Types (ADT)

The OGC OpenLS standard defines Location Based Services for a “GeoMobility Server” platform. It defines the interfaces for a couple of core service types, namely Directory Service, Gateway Service, Location Utility Service, Presentation Service, and Route Service. An overview of OGC OpenLS is available in the main document at subsection 3.4.7.

This subsection describes Abstract Data Types (ADT) definition of the standard, especially the definition of an object model for location referencing.

**Feature Catalogue**
The following is “backward-engineered” from the XML Schema definitions of the ADT definition of the standard.

A **Location** is an abstract object, which can either be an **Address**, a **POI** (Point-of-Interest), or a **Position**. Consequently, the standard allows to freely choose the representation of locations among these alternatives in most of its interface definitions.
The second alternative, named POI (Point-of-Interest), is a feature of its own right, and will therefore be discussed later in section 2.9.2.

We will therefore only treat Address and Position at this place.

Address stands for a reference to the ordering system in a municipality, it comprises streets and house numbers, or alternative description, like the crossing of streets.

Position determines locations by co-ordinates, basically points.

Let us first look at Address:

There are two alternative representations of Addresses, namely FreeFormAddresses, being simply a text string, and StructuredAddresses, which organise the usual constituents of an address.

A StructuredAddress must contain a StreetAddress, which itself consists of one or more Street-Objects and possibly the description of a Building. The model for Street is optimised for the use in U.S.A., however, European use seems possible by omitting part to the attributes. The Building carries the house number and possibly more precise descriptions.

Place optionally adds the description of higher order geo-codes, like municipality and country.

Another optional constituent is the PostalCode.
We now come to Position:

A Position in OpenLS necessarily contains a Point (from GML and thus from ISO19107), but can be enriched by more information. Especially, movement information can be added as direction and speed.

The description can also be enhanced by giving it a geometric Shape out a selection of geometry choices (ISO19107). Also a QoP (Quality-of-Position) data structure can be added to the Position definition.

**Portrayal Catalogue**

None associated.

**Service Reference**

This is part of the OpenGIS® OpenLS service definition, see 4.7.2.

### 2.4.6 ISO 18234-6 / ISO 24530-2: TPEG-LOC - Location Referencing Applications

The Transport Protocol Experts Group (TPEG) developed the TPEG specifications for transmission of language independent multi-modal Traffic and Travel Information. An overview of these specifications is available in the main document at subsection 3.1.8.

TPEG-LOC is part of the TPEG standards and provides a unified location referencing system suitable for use by all TPEG applications.

TPEG-LOC allows the service provider to give to human end users an impression of where an event has taken place, even where they may not be familiar with the location. This is achieved in a language independent way by the use of TPEG-LOC tables (using language independent dictionaries). These references supplied in the form locY_X, where Y is the table number determining the type of entry, and X is the row number determining the exact
value. For example, loc07_11 signifies entry 11, “outskirts of”, from table loc07 which is the “area qualifier” table. loc07_x would signify any entry from this table.

TPEG-LOC supports the following location referencing methods: referencing by link-ID or node-ID; referencing by geographic coordinates; linear referencing, referencing by cross streets; referencing by address information; and combinations of above.

**Feature Catalogue**

The principal objects within a TPEG-Loc message are described below:

**LOCATION CONTAINER**

The Location Container contains all information about the event location, and can completely describe locations including points, arcs, or areas. It is divided into location coordinates and location descriptions parts. The language attribute defines a default language for the descriptive parts of the message. Languages throughout the structure are defined in table loc41_x.
LOCATION COORDINATES

Figure 2-43 : TPEG LOC - Location Coordinates Class Diagram

Location Coordinates objects contain the machine readable parts of the location such as WGS84 and tpeg_ilcs (ILOC Intersection location points as defined in the EVIDENCE location referencing model). The structure of the Location Coordinates part is described in Figure 2-43.

The type of Location Coordinate is defined by a location_type code, which may be any value from table loc01_x. The following location types are currently defined: unknown; large area; segment; intersection; framed point; non-linked point; connected point; unknown.

The direction object defines a direction associated with the location. Values may be chosen from those defined in table loc02_x, which describes directions such as “anticlockwise” and “southbound”. The mode_type_list defines a list of relevant mode_of_transport associated with the location. mode_of_transport may be any value from table loc05_x. The location is defined by location_points. location_points may reference language specific location_descriptors, which themselves contain a descriptor_type, as defined in table loc03_x. This descriptor_type includes values such as “link name” and “intersection name”. The other part of the location point describes the location proper. The WGS84 object defines one or more longitude/latitude pairs for the point. The location point height may also be described in metres along with a suitable descriptor from table loc04_x, which contains entries such as “below” or “above sea level”
LOCATION DESCRIPTIONS

Location descriptions provide the “human understandable” information about a location, including more vague descriptions such as “near”, or “on the outskirts of”. Such descriptions are supported by area, node and network reference elements.

AREA REFERENCE

The Area Reference model describes the location in terms of geographic areas. The structure of the area reference is shown in Figure 2-45. Referencing is achieved through a defined area search tree.

The area_tree_entry defines the position of the entry within the tree. The area_type defines
the type of area using values as defined in table loc06_x, and includes value for types such as “district”, “town” and “urban district”. The area_qualifier may be used to further qualify the area, using entries from the table loc07_x, which includes entries for qualifiers such as “northern part of” and “low lying areas of”. The area_descriptors contain a language specific area_name string, the relevant languages for the descriptor being defined by the language object.

**NETWORK REFERENCE**

![Network Reference Class Diagram](image)

Figure 2-46: TPEG-LOC - Network Reference Class Diagram

The network_reference describes the location in terms of network links. The structure permits message filtering based on link types (both transport mode and hierarchical category) and numbers. The network_layer attribute defines the network being used from table loc08_x which contains entries for such as “road network” and “air network”. The link_number contains an unsigned integer corresponding to the route or road number, while the link_number_suffix allows a character to be suffixed to this link number. link_type defines the type of link being described. The link_type may any value from tables loc09_x to table loc16_x. Each table relates to a different transport mode e.g. table loc09_x relates to road link types, loc12_x to tram links. Each entry within the table relates to specific link type, e.g. loc09_3 “secondary road”. The network reference may be assigned more than one direction, which may contain any value from table loc02_x. The network reference may not only contain language specific link_descriptors, but also a specific segment may be defined using language specific segment_names described by a from_descriptor and to_descriptor.
The node description is a recursive (through the associated nodes element) data structure which describes single points or complex multimodal nodes which are part of a network. Node type may be mono-modal or multimodal, with modes of transport listed from those in table loc05_x. The node descriptors provide language specific names for the node. The object is located with WGS84 coordinates where node height and expansion may also be specified. A floor may also be specified.
The reference object model allows the node position to be described in a language independent descriptive way (using TPEG loc tables) by referencing known locations and landmarks, from which a distance and orientation may be specified. The reference objects are themselves given language dependent descriptors. The directional glue elements describe the direction of the node in relation to the reference object, while the proximity glue elements describe the proximity. The intermediate glue provides additional referential descriptions. Reference objects may be road objects, buildings, geographical sites and sightseeing attractions.

**Encodings**

TPEG locations are designed to describe locations in TPEG messages. TPEG messages are exchanged in two formats. The binary transmission format is described in the CEN / ISO TS 18234 series, while the XML transmission format is described in the CEN / ISO TS 24530.

**Portrayal Catalogue**

None
Service Reference

TPEG-Loc is designed for use within TPEG services, as defined in ISO TS 18234 and ISO TS 24530. TPEG-Loc may also be used within DATEX 2.

2.4.7 ISO 17572-3: AGORA-C

AGORA-C is a method for on-the-fly location referencing. An overview of the standard is available in the main document at subsection 3.5.8.

Feature Catalogue

The following description is based on the CD of the ISO document, UML diagrams have been enhanced for better readability.

AGORA-C defines DynamicLocationReferences, which can take the form of LinearLocations, PointLocations (which are a constrained variant of LinearLocations), and locations of areas, which again come in two forms: ImplicitArea and ExplicitArea.

See the following diagram pointing out these matters.

```
DynamicLocationReference
  ▲
  ▼
 LinearLocation  ▼
  ▼
 PointLocation

 ImplicitArea

 ExplicitArea
```

Figure 2-49 :AGORA-C - Dynamic Location Reference

LinearLocations, PointLocations and ImplicitAreas are defined on the basis of references to a street network. The references occur through co-ordinate values and additional attributes. ExplicitAreas are generally free-form regions, however, they may also refer to a street network, at least in parts of their description.

Let us first consider LinearLocations and implicitly PointLocations (see next figure):

A LinearLocation in AGORA-C possesses a mandatory object of type CoreLocation, an optional AttributeList, and also zero or more objects of type ExtendedLocation.
The `LinearLocation` attribute “DrivingDirections” relates the implicit direction given by the sequence of `CorePoints` in the `CoreLocation` object to the directions meant to be defined for the whole `LinearLocation`. The “LocationType” attribute specifies whether the location reference is meant for an `Intersection`, `Road`, `LimitedAccessRoad`, `Ferry`, `Settlement`, `Point-of-interest`. Other attributes may be attached to a `LinearLocation` by means of the optional `AttributeList`.

The `CoreLocation` object is described by a sequence of `CorePoints` each being optionally attached to an `AttributeList`. One of these attributes is “Dperp_max”, which stands for the maximal perpendicular distance between the approximated path and the original path in the network. The other attributes depend on the point types and the local situation. A `CorePoint` consists of a coordinate pair in WGS84 coordinates and has one to three point types.

Possible point types are:

- a `location point (LP)`, which represents the start, an intermediate, or the end point of the location to be referenced,
- an `intersection point (IP)`, which represents an intersection where the road signature changes, and
- a `routing point (RP)`, which are used to reconstruct the location by route calculation.

---

**Figure 2-50 : AGORA-C - Linear Location**
Additional attributes can be attached to a **CorePoint** by means of **RPSignature**, **IPSignature** and **SideRoadSignature** objects. Each carry specific attributes for the possible point types.

A **LinearLocation** may also have zero or more **ExtendedLocations**.

These usually stand for additional paths which do not belong to the location reference proper, but which may aid as a geometric pattern for better recognizing the position in the location decoder. **ExtendedLocations** can, however, also be employed to describe the location, for example, if they lead into terrain not covered by the target map. The whole construct is linked to one **CorePoint** – this is where the **ExtendedLocation** starts.

**ExtendedLocations** consist of sequences of **ExtendedPoints**, which are principally constructed analogously to **CorePoints**.

**ImplicitAreas** are constructed by ordered references to the network, called **SubNetworkLocation**. Each of these consists of a **LinearLocation** and **LocationReferenceConnection**, which properly links the given **LinearLocations**.

![Diagram](Figure 2-51 : AGORA-C - Implicit Areas)

We finally describe **ExplicitAreas**.

These can either be a **SimpleGeometricArea**, which constitutes a point either given as **PointLocation** (i.e. by means of a network reference) or directly by co-ordinates. The **GeometricFigure** is to be interpreted as a rectangle, a circle, or an ellipse at this point.
Another way to create an *ExplicitArea* is by forming an *OutlinedArea*. Such an area is formed by *PolylineSegments* (sequences of co-ordinates) and/or references to *LinearLocations*. All these descriptive elements are properly linked to form an area boundary.

**Encodings**

The AGORA-C standard defines several encodings, among these a physical format named DLR1LocationReference which allows AGORA-C encodings being transferred using a small number of bytes.

There is also an XML Schema definition contained in the CD standard.

**Portrayal Catalogue**

None

**Service Reference**

Decoder and encoder services are currently being developed by the institutions involved in the AGORA-C standardisation process.

**2.4.8 Comparison and Evaluation**

Common methods according to ISO LRM 17572 (Location Referencing for Geographic Databases) are:

- Reference by LinkID or NodeID
- Reference by Geographic Coordinates
- Reference by Grid
- Linear Referencing
- Referenced by cross streets
- Referencing by address info

plus combinations of the above.

**DATEX 2** uses
- for point: Reference Point(Reference Point Id plus details); Alert-C(method 2 or 4); TPEG-Loc and; x-y coordinates (ETRS89/WGS84).
- for linear: 2 Reference Points; Alert-C linear, TPEG Linear Location
- for area: Alert C area, TPEG area location

Also there are used predefined locations to describe groups of sections on network to describe a route. A Supplementary Positional description includes carriageway, lanes, length affected, description.

**Alert-C** relies on precoded locations, with ability to describe a primary (location of incident) and secondary (extent of queue) location.

**ISO 19133** linear referencing describes basis for description of linear referencing, which includes ability to describe the location on a linear feature by its curvilinear distance from a reference point and lateral offset from the linear feature.

**ISO 19133** also specifies a mechanism for describing positions by multiple methods including coordinates, linear referencing, feature names, etc. for the use in a location based services setting. This includes also a flexible data model for addresses.

**OGC OpenLS** allows description of locations as (structured or freeform) Addresses, POI or Positions. Position includes ability to describe a shape (Ellipse, Circle, Circular Arc, Polygon or Multi-Polygon), plus an expression of Quality of Position (accuracy).

**TPEG-Loc** supports all the common methods mentioned in ISO LRM 17572. The emphasis is on providing not only a point position and expansion, but also the language aware textual descriptions of the area, network links, and nodes, the latter referencing known locations and landmarks.

**AGORA-C**, designed for dynamic location referencing between client and service map databases from different suppliers. It has the ability to describe Points, Linear locations, Implicit Areas (allowing description of subnetworks connected in a hierarchical manner), or Explicit Areas (enclosed by a rectangle, circle or ellipse, or by describing enclosing Linear locations and polylines). The facility exists to attach a "precise geometry description" for use in certain cases including those where a lack of information is present in the client's map.

The method may describe not only locations on links on the client network, but also by Location Reference Extensions, to describe guidance to locations which might not be present on
the client, with reference to known points on the client's network.

We should note that other standards mentioned within this document provide information which might be relevant to resolve object locations. NaPTAN stops, apart from specifying a grid reference, may also contain a structured description describing the stop in terms of cross streets and indication of position with respect to local landmarks. Hierarchical and neighbouring relationships are also described in NaPTAN and IFOPT.

We will now try a valuation of the suitability of the various location referencing methods:

Referenced locations include single points, linear locations describing lengths of network links, and areas defined by simple geometric shapes, 2 dimensional areas enclosed by a polygon of points, or 2 dimensional areas enclosed by linear locations.

All methods mentioned, except Alert-C (where locations are predefined for client and server), use some form of coordinate system to describe the referenced location. For national standards these will include local grid reference systems, but for international or European standards these will be standards such as WGS84 and ETRS89. This raises the requirement for delivery of coordinates in the client's chosen coordinate system, or delivery in multiple coordinate systems to be filtered based on client capabilities, or additional services, which can perform the necessary transformations between systems.

Coordinates may not be sufficient to describe locations, either due to potential for misleading positioning due to mismatch between client and server mapping systems, or simply client inability to display or resolve such information. Location referencing methods therefore include additional information to locate the feature. This information could be broadly described as relating to certain Geographic Identifiers (see ISO19112)

Textual descriptions seem to be more appropriate in cases where the client does not possess the ability to display a location on a map, but also appears appropriate for public transport applications, where a user may be more likely to refer to a settlement name as origin or destination. Whereas on the public transport side NaPTAN stops possess a fixed structure description (describing relation to relevant streets and landmarks), and both NaPTAN/NPTG and IFOPT are able to describe neighbours, hierarchical relations and location in relation to settlements and points of interest, rich structured textual descriptions suitable for both traffic and public transport information may also be provided in TPEG-Loc. ISO 19133 includes a refined address model, which can also supply reasonably readable addresses. OGC OpenLS includes the ability to specify a Point of Interest, as well as an address (either unstructured or structured).

Such verbal descriptions, however well structured, may not be helpful to position the location on a client device map, where the feature should be located on or with reference to a particular place on the network. Where the feature of interest occurs on a network node, specification of intersecting links should be sufficient to locate the feature, assuming such links may be unambiguously determined on the client (such as required with TPEG's intersection locations). Linear referencing, as described in ISO 19133, allows a feature to be located along (or
offset at a specified distance from) a linear network element known to both client and server. AGORA-C is designed to address potential mismatch between client and server network information specifying a set of sequentially connected coordinate points, by using information such as road classifications, bearings, connection angles and directions, to provide a location reference with high accuracy and small message size.

2.5 Traffic Flow Data

Traffic flow data comprises sensor data like traffic volume, traffic density, velocity, travel times, delay times as well as derived quantities like LoS for points on the road network, roads, routes or the whole network. Data contains a validity period if forecasted or historic.

We also count the images outputs from traffic cameras among traffic flow data, because these are usually generated jointly with the sensor data.

2.5.1 DATEX 2

The main aim of DATEX 2 is to establish standardised links between Road Operators, TICs and Service Providers to allow the easy and efficient communication of traffic and traveller Information. An overview of the standard is available in the main document at subsection 3.1.2.

Feature Catalogue

The following description will only use Level A features from DATEX 2.

Information exchanged with DATEX 2 systems in the area of traffic messages is composed of different basic elements:

- Elaborated data (derived/computed data, e.g. travel times, traffic status) are normally derived on a periodic basis by the Traffic Centre from input data for specified locations like travel times, traffic status, traffic values and weather values

- Measured data (direct measurement data from equipments or outstations, e.g. traffic and weather measurements) are normally derived from direct inputs from outstations or equipment at specific measurement sites (e.g. loop detection sites or weather stations) which are received on a regular (normally frequent) basis like traffic values (flow, speed, concentration/density and individual vehicle data), weather values (temperature, wind …), travel times and traffic status

In addition Predefined Locations and Measurement Site Table information are also exchanged. These are not part of the basic elements, but are required if the information in the basic elements is to be understood by a client.
DATEX 2 distinguishes two types of publication\(^2\) of these basic elements, where the basic elements can be exchanged individually or grouped:

- Elaborated data publication: is used to provide information that is in some way elaborated or derived.
- Measured data publication is used to provide data from measurements (traffic data measurement or weather data measurement)

For eMOTION the “Elaborated data publication” is more interesting and will be described in the following section. The basic elements of “Traffic Status”, “Travel Times”, “Traffic Values” and “Weather Values” can be part of a “Elaborated data publication”.

TrafficStatusValue describes the current Level of Service information including a trend of development of the traffic condition. TrafficValue describes all kind of measured traffic data like speed, volume or density. The third type of traffic data is travel time data which contains information about current and forecasted travel time. All these types of traffic flow data are a specialisation of BasicDataValue. This entity contains general information about the provided traffic flow data like data quality. A component of BasicDataValue is an affected location in terms of GroupOfLocation (see chapter 2.4.1).

Weather data can also be provided in the ElaboratedDataPublication and the MeasuredDataPublication but will be described in chapter 2.10.4.

\(^2\) It is possible to communicate elaborated or measured data in a so called Traffic View publication. Because this publication type is not yet validated and it is not intended to use the Traffic View publication to establish a raw data feed, this publication type will not be described in this document.
Traffic Status

**TrafficStatusValue**

<table>
<thead>
<tr>
<th>Specialises</th>
<th>BasicDataValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>The status of traffic conditions on a specific section or at a specific point on the road network.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>trafficStatus</td>
<td>TrafficStatusEnum:</td>
<td>Status of traffic conditions on the identified section of road in the specified direction.</td>
</tr>
<tr>
<td></td>
<td>impossible, congested, heavy,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>freeFlow, unknown</td>
<td></td>
</tr>
<tr>
<td>trafficTrendType</td>
<td>TrafficTrendTypeEnum;</td>
<td>A characterisation of the trend in the traffic conditions at the specified location and direction.</td>
</tr>
<tr>
<td></td>
<td>trafficBuildingUp, trafficEasing, trafficStable, unknown</td>
<td></td>
</tr>
</tbody>
</table>

Travel Times

**TravelTimeValue**

<table>
<thead>
<tr>
<th>Specialises</th>
<th>BasicDataValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Derived/computed travel time information relating to a specific group of locations.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>freeFlowSpeed</td>
<td>KilometresPerHour</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>freeFlowTravelTime</td>
<td>Seconds</td>
</tr>
<tr>
<td>normallyExpected-</td>
<td>Seconds</td>
</tr>
<tr>
<td>TravelTime</td>
<td></td>
</tr>
<tr>
<td>travelTimeTrendType</td>
<td>TravelTimeTrendTypeEnum: decreasing, increasing, stable</td>
</tr>
<tr>
<td>vehicleType</td>
<td>VehicleTypeEnum: anyVehicle, articulatedVehicle, bus, car, carOrLightVehicle, carWithCaravan, carWithTrailer, four-WheelDrive, goodsVehicle, heavy-Lorry, heavyVehicle, highSidedVehicle, lightVehicle, lorry, motorcycle, twoWheeledVehicle, van, vehicleWithCatalyticConverter, vehicleWithoutCatalyticConverter, vehicleWithCaravan, vehicleWithTrailer, withEvenNumberedRegistrationPlates, withOddNumberedRegistrationPlates, other</td>
</tr>
</tbody>
</table>

**TrafficValues**

TrafficValues represent either averaged traffic flow data like headway, speed, flow (volume) and concentration (density) or single vehicle traffic flow data like speed, headway or detection time. A component of TrafficValues is VehicleCharacteristics. This means that every traffic data type can be assigned to a specific kind of vehicle depending on vehicle type, vehicle occupancy or fuel or load type.
Weather Values
See chapter 2.10.4.

Encodings
DATEX 2 has been modelled using UML 1.5. There is a demand to move to UML 2.0 in the future, but currently, it is important to understand that DATEX 2 is still based on UML 1.5.
The XML encoding of DATEX 2 has been generated from the UML representation and the
data dictionary by an automated conversion tool.

The following figure shows the workflow of the automated conversion process.

![Diagram: DATEX 2 - XML schema generation process]

*Figure 2-55: DATEX 2 - XML schema generation process*

The conversion tool is accessible in the Internet at http://www.datex2.eu.

**Portrayal Catalogue**

There is no portrayal associated with this standard.

**Service Reference**

There is a reference to DATEX 2 data service described in chapter 4.2.3.

**2.5.2 Journey Time Database (JTDB)**

The UK Highways Agency Journey Time Database provides journey time information for each 15-minute interval throughout the year for road links between junctions on the Highways Agency core network. An overview of the JTDB is available in the main document at subsection 3.1.10.

**Feature Catalogue**
Road Timetable is a data model for exchanging journey times on road links, for particular day types, with a mapping of day types to an unlimited number of specific days in the past and/or future. The `RoadTimeTableType` contains the administrative information about the data sets like period of validity or the creation date.

The class `TrafficDayType` describes the types of days for what the traffic data are valid. There are several categories of types consisting of a day type and a day-of-week like "Any: Christmas day", "Monday -- Friday: Bank holiday", "Saturday: Mid-August to mid-May (start and end of the football season)".

The class `RoadLinkType` contains the network information. Road links can be described by

---

3 Source: ITS Metadata Registry: http://www.itsregistry.org.uk/index.html
coordinates or OSGR data (Ordnance Survey grid reference) for the starting points and end points of road sections and by road names and numbers. In addition this class contains information about traffic attributes like free flow speed and existing speed limits.

The class of SpeedDataType contains speed data for each 15 minute time period (daily time series). Speed data are the average speed and the average travel time and the standard derivation of speed and travel time. The class also contains some administrative information like the data origin and the last update time.

**Encodings**

**Portrayal Catalogue**
Not associated.

**Service Reference**
unknown

### 2.5.3 Comparison and Evaluation

In the area of traffic data there is only DATEX 2 as international standardisation available. DATEX 2 covers all relevant data that are needed by traffic information services. Although DATEX 2 was originally developed for the communication between Traffic Information, and Traffic Management Centres with the focus on the Freeway network, Datex can also be used in the area of urban traffic. Because DATEX 2 used different referencing systems it can be used in a broad area of traffic data communication.

The Journey Time Database is a standardisation from UK and covers only a subsection of traffic data that also is covered by DATEX 2.

### 2.6 Traffic Messages

Traffic messages concern incidents and accidents, congestion, road works and road closures for specific points on the road, roads, routes or administrative areas. Traffic impact data, like detour recommendations, also fall into this category.

A validity period is attached for historic data and planned events like road works messages.

#### 2.6.1 DATEX 2

The main aim of DATEX 2 is to establish standardised links between Road Operators, TICs and Service Providers to allow the easy and efficient communication of traffic and traveller Information. An overview of the standard is available in the main document at subsection 3.1.2.
Feature Catalogue

The following description will only use Level A features from DATEX 2.

Information exchanged with DATEX 2 systems in the area of traffic messages is composed of different basic elements:

- Road and traffic related events (called `TrafficElements`) are all events which are not initiated by the traffic operator and force him to undertake (re)actions. They are classified in 6 main categories: (i) abnormal traffic such as long queues, stop and go; (ii) accidents; (iii) obstructions such as animal or vehicle presence, environmental or road equipment caused obstacles like fallen trees, rock falls or fallen power cables; (iv) activities held by authority such as police investigation, caused by disturbances such as an explosion or occurring with public events such as a bicycle race; (v) incidents on infrastructure equipment such as a variable message sign being out of order, tunnel ventilation not working, emergency telephone not working; and (vi) specific road or environment conditions related to weather such as ice, precipitation, wind or other reasons like oil on road surface.

- Operator actions classified in 4 main categories: (i) network management such as road closure, rerouting, traffic control measures; (ii) road works; (iii) road side assistance like vehicle repair, helicopter rescue; (iv) sign settings like VMS or matrix signs. Because this basic element is not so important for eMOTION it will not be described in the further sections.

- Advice such as speed limit, lane usage, winter driving.

- Impacts, in particular delays and traffic status (level of service). Because this basic element is not so important for eMOTION it will not be described in the further sections.

- Non road event information concern information about events which are not directly on the road (e.g. transit information, service disruption, car parks)

In addition `PredefinedLocations` and `MeasurementSiteTable` information is also exchanged. These are not part of the basic elements, but are required if the information in the basic elements is to be understood by a client.

DATEX 2 distinguishes two types of publication of these basic elements, where the basic elements can be exchanged individually or grouped:

- `SituationPublication` is used to provide data about an identifiable instance of a traffic situation comprising one or more traffic circumstances which are linked by one or more causal relationships.

- `TrafficViewPublication` is used to provide a snapshot of what happens on one itinerary, in one direction at a given time.

The following table presents what can be exchanged with each publication.
The most important DATEX 2 publications regarding to traffic messages are *SituationPublication* and *TrafficViewPublication*. A situation publication can contain several different situations. A situation represents a traffic/travel situation comprising one or more traffic/travel circumstances which are linked by one or more causal relationships and which apply to related locations. Each traffic/travel circumstance is represented by a *SituationRecord*.

A situation record is one element of a situation. It is characterised by values at a given time, defining one version of this element. When these values change, a new version is created.

---

A *Traffic View* is a snapshot of what happens on one itinerary, in one direction at a given time. A traffic view is autonomous: it contains all necessary information to understand what is happening on this itinerary at the specific point in time. There is no historical management. There is no need to manage a link failure and then recovery because, by definition, traffic view n° J cancels and replaces traffic view n° J-1. A traffic view is organised in "oriented road sections". Each section can contain Elaborated data (traffic status, travel time, delay), Road or traffic related events (if any) and Operators actions (if any). The Traffic View publication is not yet validated. The situation Publication is the publication to communicate traffic messages. For this reason the traffic view will not be described in this document.
The main component of a Situation is a SituationRecord. This SituationRecord is a generalisation of different kinds of situation like NonRoadEventInformation (Information about an event which is not on the road, but which may influence the behaviour of drivers and hence the characteristics of the traffic flow.), TrafficElement (An event which is not planned by the traffic operator, which is affecting, or has the potential to affect traffic flow.), OperatorAction (Actions that a traffic operator can decide to implement to prevent or help correct dangerous or poor driving conditions, including maintenance of the road infrastructure.)

Important properties of SituationRecord are the Location as reference of the situation and the Cause, where a NonManagedCause is mainly described by a enumeration (accident, traffic congestion, failure of roadside equipment, failure of road infrastructure, obstruction on the roadway, poor weather conditions, problems at the border crossing etc.) and a ManagedCause is a reference to another SituationRecord.

The most important components of SituationRecords are Impact and Advice. Impact describes the deduction of road capacity in combination with information about the number of closed or affected lanes and information about the resulting delays.
Advice covers all types of recommendation that can be given to the road user in case of poor traffic conditions. The different types of recommendation are described in several enumerations.
<table>
<thead>
<tr>
<th>Advice Enumerations</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiversionAdviceEnum</td>
<td>compulsoryDiversionInOperation, compulsoryDiversionInOperation, diversionIsNo LongerRecommended, doNotFollowDiversionSigns, followDiversionSigns, followLocalDiversion, followSigns, followSpecialDiversionMarkers, heavyLorriesAreRecommendedToAvoidTheArea, localDriversAreRecommendedToAvoidTheArea, noSuitableDiversionAvailable, other</td>
</tr>
<tr>
<td>InstructionsEnum</td>
<td>allowEmergencyVehiclesToPass, approachWithCare, clearALaneForEmergencyVehicles, clearALaneForSnowploughsAndGrittingVehicles, closeAllWindowsTurnOffHeaterAndVents, crossJunctionWithCare, doNotAllowUnnecessaryGaps, doNotDriveOnTheHardShoulder, doNotLeaveYourVehicle, doNotSlowdownUnnecessarily, doNotThrowOutAnyBurningObjects, driveCarefully, driveWithExtremeCaution, followTheVehicleInFrontSmoothly, increaseNormalFollowingDistance, inEmergencyWaitForPolicePatrol, keepYourDistance, leaveYourVehicleProceedToNextSafePlace, noNakedFlames, noOvertaking, noSmoking, observeSignals, observeSigns, onlyTravellIfAbsolutelyNecessary, overtakeWithCare, pleaseUseBusService, pleaseUseRailService, pleaseUseTramService, pleaseUseUndergroundService, pullOverToTheEdgeOfTheRoadway, stopAtNextSafePlace, stopAtNextServiceArea, switchOffEngine, switchOffMobilePhonesAndTwoWayRadios, testYourBrakes, useFogLights, useHardShoulderAsLane, useHardWardingLights, useHeadlights, waitForEscortVehicle, other</td>
</tr>
<tr>
<td>LaneUsageEnum</td>
<td>heavyVehiclesUseLeftLane, heavyVehiclesUseRightLane, keepToTheLeft, keepToTheRight, useBusLane, useHardShoulder, useHeavyVehicleLane, useLeftHandParallelCarriageway, useLeftLane, useLocalTrafficLanes, useRightHandParallelCarriageway, useRightLane, useThroughTrafficLanes, other</td>
</tr>
<tr>
<td>SpeedAdviceEnum</td>
<td>mandatorySpeedLimitInForce, observeRecommendedSpeed, observeSpeedLimits, policeSpeedChecksInOperation, reduceYour-</td>
</tr>
</tbody>
</table>

Figure 2-59: DATEX 2 - Advice : logical diagram
TrafficElement represents events that worsen traffic flow.

**TrafficElement** represents events that worsen traffic flow.

<table>
<thead>
<tr>
<th>TrafficElement</th>
<th>Conditions</th>
<th>AbnormalTraffic</th>
<th>AbnormalTraffic</th>
<th>AbnormalTraffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed, speedLimitInForceForHeavyVehicles, other</td>
<td>aquaplaningRisk, danger, dangerOfExplosion, dangerOfFire, emergencyVehiclesAtScene, extraPolicePatrolsInOperation, firemenDirectingTraffic, helicopterRescueInProgress, increasedRiskOfAccident, lookOutForFlagman, pilotCarInOperation, policeDirectingTraffic, policeInAttendance, radiationHazard, repairsInProgress, rescueAndRecoveryWorkInProgress, severalAccidentsHaveTakenPlace, skidRisk, slipperyPavements, surfaceWaterHazard, toxicLeak, trafficBeingDirectedAroundAccidentArea, trafficWardensDirectingTraffic, other</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WinterEquipmentAdviceEnum</td>
<td>SnowChainsOrTyresRecommended, snowChainsRecommended, snowTyresRecommended, studTyresMayBeUsed, winterEquipmentRecommended, other</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Conditions** contain information about any conditions which have the potential to degrade normal driving conditions. These conditions could be caused by road condition (due to poor weather situation – e.g. skid risk – or poor road infrastructure) and environment related conditions (such as poor weather conditions such as fog, extreme wind conditions or smog).

**AbnormalTraffic** gives information about the kind of traffic flow, i.e. specifically relating to the nature of the traffic movement. It is possible to add information about queue length, number of waiting vehicles and information about the future development of traffic movement. AbnormalTraffic can be interpreted as a more detailed description of level of service information.

---

© eMOTION Consortium
Page 119 of 354
PoorRoadInfrastructure is a collection of data providing information on reduced performance road infrastructure components (such as payment lanes at toll plazas not working, damage to bridges, emergency phone number not in operation, and lane control signs not working) that may be of interest or concern to travellers.

Accident describes cause (like driver distraction, driver illness, loss of vehicle control due to excessive vehicle speed) and type (common accidents, accidents involving hazardous materials, accidents involving busses or other special vehicle types, collision with animal) of any road accident.

Obstruction describes any stationary or moving obstacle of a physical nature (e.g. obstacles or vehicles from an earlier accident, shed loads on the carriageway, rock fall, abnormal or dangerous loads, or animals etc.) which could disrupt or endanger traffic. This covers animal presence on the road, obstruction caused by environmental conditions (like fallen trees on the road), obstruction caused by damage of road infrastructure or equipment and obstruction caused by vehicles.

OperatorAction

OperatorAction covers all actions that a traffic operator can decide to implement to prevent or help correct dangerous or poor driving conditions, including maintenance of the road infrastructure. One of the most important specialisation of OperatorAction is Roadworks. Others are SignSetting (Provides information on variable message and matrix signs and the information currently displayed.), NetworkManagement (Changes to the configuration or usability of the road network whether by legal order or by operational decisions. It includes road and lane closures, weight and dimensional limits, vehicle restrictions, contraflows and rerouting operations.) and RoadsideAssistance (Details of road side assistance e.g. in case of accident).
Specialises: OperatorAction

Description: Highway maintenance, installation and construction activities that may potentially affect traffic operations.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>effectOnRoadLayout</td>
<td>EffectOnRoadLayoutEnum: carriagewayClosures, contraflow, laneClosures, lanesDeviated, narrowLanes, newRoadworksLayout, obstacleSignalling, roadLayoutUnchanged, temporaryTrafficLights</td>
<td>The effects which the road works have or are expected to have on the road layout.</td>
</tr>
<tr>
<td>roadworksDuration</td>
<td>RoadworksDurationEnum: longTerm, mediumTerm, shortTerm</td>
<td>Indicates in general terms the expected duration of the road works.</td>
</tr>
<tr>
<td>roadworksScale</td>
<td>RoadworksScaleEnum: major, medium, minor</td>
<td>Indicates in general terms the scale of the road works.</td>
</tr>
<tr>
<td>underTraffic</td>
<td>Boolean</td>
<td>Indicates that the road section where the road works are located is trafficked or non-trafficked.</td>
</tr>
<tr>
<td>urgentRoadworks</td>
<td>Boolean</td>
<td>Indication of whether the road works are considered to be urgent. 'True' indicates they are urgent.</td>
</tr>
</tbody>
</table>

Figure 2-61: DATEX 2 - Roadworks: logical diagram
Encodings

Portrayal Catalogue
There is no portrayal associated with this standard.

Service Reference
There is a reference to DATEX 2 data service described in chapter 4.2.3.

2.6.2 ISO 18234-4 / ISO 24530-3: TPEG-RTM - Road Traffic Message Application

The Transport Protocol Experts Group (TPEG) developed the TPEG specifications for transmission of language independent multi-modal Traffic and Travel Information. An overview of these specifications is available in the main document at subsection 3.1.8.

The TPEG-RTM application is designed to allow the efficient and language independent delivery of road information directly from service provider to end-users. The information provided relates to event and some status information on the road network and on associated infrastructure affecting a road journey. For example, limited information about abnormal operation of links in the network may be included, such as ferries, lifting-bridges, etc.

Feature Catalogue

Each TPEG-RTM application message consists of three parts – the Message Management Container, the Application Event/Status Container and the Location Container (see picture below). Due to specific focus of ISO 18234-4 and ISO 24530-3 on the road traffic messages (placed in the app./event container), the content of Application Event/Status Container is described in a more detailed way than the other two.

Each message needs some event or information content and thus another container is assigned to this role – varying significantly according to application. This is recognized by the documentation of separate Parts for the varying applications (e.g. RTM, PTI etc.)
**Description of the event attributes**

Every event – independent from its logical class – has the same attributes. The diagram shows logical classes to demonstrate how events can be used for different kinds of information.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Code</td>
<td>In the original message this code is transmitted in binary form, in a decimal representation it can be up to 2048 as described before.</td>
</tr>
</tbody>
</table>
| Event Nature         | - Information  
- Forecast  
- Silent                                                                                                                                  |
| Optional Quantifier  | An optional field containing the reference numbers of quantifiers listed in the table of the event list. The quantifier contains more detailed information about the event. It describes the event’s duration (e.g. 20 minutes) or the affected distance (e.g. 5 miles or 20 minutes). |
| Duration Type        | - Dynamic  
- Long lasting  
- No duration shall be presented to the user                                                                                              |
| Default Directionality| - one direction                                                                                                                                  |
### Default terminal urgency
- Extremely urgent
- Urgent
- Normal event

### Event's update class
Number of the update class. The update class contains information for the receiving terminal on how to deal with updates of a message.

### Phrase code
A phrase code consists of 2 parts:
- Code Letter A – Z
- Code Number 1 – 999

It contains additional information for the TMC operators and helps them to give more precise information when translating the code into a verbal message. The codes are also defined in the event list.

E.g. code letter “D” indicates a dangerous event.

The ISO 18234-4 and 24530-3 deal with the road traffic messages data content definitions which are covered by the Application Event/Status Container. The other two parts of the TPEG message are the focus of other TPEG standards.

The following table gives a short overview of the components of a TPEG Message:

| Message management container | The TPEG Message consists amongst others of the type “Message Management Container”, which contains mandatory elements, the date and time elements (e.g. start time, stop time), the severity and reliability elements and coding of the message management container. |
| Application Event/Status Container | The component “Application Event/Status Container” consists of classes of Accidents, Obstructions, Activities, Road conditions, Network Performance, Network Conditions, Facilities Performance, Security Alert, Moving Hazard, Public Transport Information, Visibility, Weather, Diversion Advice |
| Location Container | The “Location Container” type provides both human and machine readable content in bilingual locations and includes more detailed area, network and node description. It is covered in more detail in subsection 2.4.6. |

The component “Application Event/Status Container” consists of the following classes:

<table>
<thead>
<tr>
<th>Class</th>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accidents</td>
<td>Position, animals specification, vehicles description, people data</td>
<td>Descriptions of situations in which road users (vehicles, animals and people) do not behave in a predictable or safe manner and either impact with each other or the roadside infrastructure and in some cases may leave the road.</td>
</tr>
<tr>
<td>Obstructions</td>
<td>Position, animals specification, vehicles description, people data, object specification</td>
<td>Descriptions of situations in which road users (vehicles, animals and people) or other causes (man-made or environmental) make it difficult or impossible for other road users to progress along that part of roadway.</td>
</tr>
<tr>
<td>Activities</td>
<td>Position, activity description, people data</td>
<td>Descriptions of events (particularly involving people) that can have an impact on the road traffic.</td>
</tr>
<tr>
<td>Road Conditions</td>
<td>Position, surface specification</td>
<td>Descriptions of changes to the properties of the</td>
</tr>
</tbody>
</table>
tion, adhesion description, marking

Network Performance
Performance, speed, delay, travel time

Network Conditions
Position, regulation, restriction, road-works

Facilities Performance
Traffic control, roadside assistance, roadside services

Moving Hazard
Position, animals specification, vehicles description, people data

Security Alert
Security alert specification

Public Transport Information
Public transport info description

Visibility
Obscurity, visual acuity, lighting, length affected

Weather
Precipitation, wind specification, temperature data

<table>
<thead>
<tr>
<th>Network Performance</th>
<th>Performance, speed, delay, travel time</th>
<th>Descriptions of the effect on road users (delay, flow or speed) that arise out of external events.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Conditions</td>
<td>Position, regulation, restriction, road-works</td>
<td>Descriptions of changes to network conditions planned, imposed or advised by the road network operator that affect the drivers, vehicles and routing.</td>
</tr>
<tr>
<td>Facilities Performance</td>
<td>Traffic control, roadside assistance, roadside services</td>
<td>Descriptions of the changes (for any reason) to the availability of control, assistance and roadside services.</td>
</tr>
<tr>
<td>Moving Hazard</td>
<td>Position, animals specification, vehicles description, people data</td>
<td>Descriptions of situations in which non-stationary road users (vehicles, animals and people) make it hazardous for others to use the road.</td>
</tr>
<tr>
<td>Security Alert</td>
<td>Security alert specification</td>
<td>Descriptions of situations which may have safety implications for the road user.</td>
</tr>
<tr>
<td>Public Transport Information</td>
<td>Public transport info description</td>
<td>Descriptions of abnormal operation of all forms of public transport which may affect the road user’s journey.</td>
</tr>
<tr>
<td>Visibility</td>
<td>Obscurity, visual acuity, lighting, length affected</td>
<td>Descriptions of changes to the normal obscurity or lighting conditions that may affect the road user’s ability to see the road or other road users ahead.</td>
</tr>
<tr>
<td>Weather</td>
<td>Precipitation, wind specification, temperature data</td>
<td>Descriptions of weather situations that are affecting or may affect the progress of the road user.</td>
</tr>
</tbody>
</table>

Encoding

There exist two encodings of a TPEG type RTM application: (i) tpegML, which is an XML implementation for use in editing systems and via the Internet and is used by ISO/TS 24530-3; and (ii) a binary TPEG description used for transmission over e.g. DVB and is used by ISO/TS 18234-4.

Portrayal Catalogue

There is no portrayal associated with this standard.

Service Reference

Data content defined and described in this standard is in liaison mainly with the dynamic navigation systems and traffic control centres.

2.6.3 ISO 14819 - RDS-TMC

RDS-TMC (Radio Data System – Traffic message Channel) is an Europe wide standard for transferring traffic relevant information into mobile devices. An overview of the standard is available in the main document at subsection 3.1.7.

This section mainly treats the message contents of RDS-TMC.
For a description of the location table, which defines the network the messages refer to, see 2.1.4, for the location referencing aspect see 2.4.2.

**RDS-TMC Message Structure**

A RDS-TMC message contains the following information:

- an event description
- location information
- direction and extent of the road segments which are effected by the event
- duration of the event
- diversion information for the user if it is available or recommendable
Feature Catalogue - Events

The ISO 14819-2 standard defines an event list which contains all possible events that can be used. There are simple events like “heavy snowfall” and “visibility reduced” both having their own event number. The combination of these events creates the new event “heavy snowfall. Visibility reduced” which is defined by a separate event number. Thus there is always one event describing the whole situation.

The events can logically be separated in 2 classes: real events and forecast events. The separation of the two classes is only done on a logical level. It does not influence the coding of the events. That’s why every event (independently from its class) has the same attributes.

Real events are these events which have already happened. Forecast events are expected events which inform the driver about expected conditions on the road.

Figure 2-64: RDS-TMC - UML Classes for RDS-TMC events

Real Events

The following UML diagram shows what kind of events are covered by the class “Real events” – events which have really happened. For brevity, the diagram does not cover all

---

5 System messages contain control information for the receiving terminal. They do not contain any relevant content for eMotion thus they are not further discussed in this document. More details about Extent, Direction and Lodation can be found in chapter 2.4.2. The events will be covered in this chapter.
logical subclasses. The complete list can be found below the diagram. This classification is only a logical classification for the reader to get an impression what is possible with RDS-TMC events. This classification is not represented in the real coding, thus it is not necessary to have the complete list in the UML diagram.

Real events can be logically classified as follows:

- Level of service (e.g. queuing traffic for 2 km) – 418 events
- Expected level of service (e.g. shed load(s), slow traffic expected) – 59 events
- Accidents (e.g. multi-vehicle accident) – 28 events
- Incidents (e.g. broken down vehicles) – 14 events
- Closures and lane restrictions (e.g. one lane closed) – 126 events
- Carriageway restrictions (e.g. parallel carriageway closed) – 19 events
- Exit restrictions (e.g. slip roads closed) – 12 events
- Entry restrictions (e.g. entry blocked) – 6 events
- Traffic restrictions (e.g. closed duet so sports meeting) – 56 events

---

6This logical classification does not exist in the standard. There are just events, without any classification it all. But the standard itself has introduced this logical classes to provides a better overview about the different kind of information which can be coded by RDS-TMC messages. The code numbers do not contain any information about this logical class.
• Carpool information (e.g. police directing traffic via the carpool lane) – 11 events
• Roadworks (e.g. major road works) – 30 events
• Obstruction hazards (e.g. fallen power cables) – 72 events
• Dangerous situations (e.g. large animals on roadway) – 23 events
• Road conditions (e.g. freezing rain expected) – 56 events
• Temperatures (e.g. current temperature) – 8 events
• Precipitation and visibility (e.g. heavy snowfall. Visibility reduced) – 58 events
• Wind and air quality (e.g. severe smog) – 13 events
• Activities (e.g. exhibition) – 47 events
• Security alerts ( e.g. police checkpoint) – 13 events
• Delays (e.g. resurfacing work. Long delays) – 147 events
• Cancellations (e.g. public transport strike) – 31 events
• Travel time information (e.g. expected trip time xy) – 9 events
• Dangerous vehicles (e.g. vehicles on wrong carriageway) – 16 events
• Exceptional loads / vehicles (e.g. wide loads) – 21 events
• Traffic equipment status (e.g. traffic light not working) – 30 events
• Size and weight limits (e.g. temporary axle load limit xy) – 11 events
• Parking restrictions (e.g. special parking restrictions in force) – 5 events
• Parking (e.g. no parking spaces available) – 29 events
• Reference to audio broadcast (e.g. urgent information will be given on normal program broadcasts) – 8 events
• Service messages (e.g. this service provides major road information) – 26 events
• Special messages (e.g. nothing to report) – 5 events

**Forecast Events**

Forecast events can also be logically divided into different groups. The following diagram shows all logical classes of the forecast events. These classes are not represented in the real coding. This representation is just for a better understanding and easier reading.

---

7 This logical classification does not exist in the standard. There are just events, without any classification it all. But the standard itself has introduced this logical classes to provides a better overview about the different kind of information which can be coded by RDS-TMC messages. The code numbers do not contain any information about this logical class.
The forecast events are divided in the following classes:

- Level of Service forecast (e.g. fair. Heavy traffic has to be expected) – 10 events
- Weather forecast (e.g. (Q probability of) winter storm)– 32 events
- Road conditions forecast (e.g. ice expected (above Q hundred meters))– 8 events
- Environment forecast (e.g. (Q probability of) smog) – 11 events
- Wind forecast (e.g. strong winds (Q)) - 12 events
- Temperature forecast (e.g. temperature rising (to Q)) – 22 events
- Delay forecast (e.g. delays of several hours have to be expected) – 12 events
- Cancellation forecast (e.g. park and ride service will not be operating (until Q) – 11 events

In summary the ISO standard defines 1520 events at the moment. The Alert-C coding can express 2048 different events, so the list still can grow a lot.

Every event has exactly one code which contains a lot of additional information. The structure of the event is shown in the UML diagram Figure 2-66 above for easier understanding.

---

8 The „Q“ represents the optional quantifier of the event which will be described in the next section. This qualifier gives more detailed information about the event in terms of duration or the concerned distance.
Encodings

The ISO 14918-1 standard describes the structure of a RDS-TMC message. These messages are based on the RDS system which is fully described in IEC 62106:2000. RDS-TMC uses RDS type 8A groups to carry the RDS-TMC messages. Such groups consist of a series of 4 data blocks (16 bits) each followed by a block of 10 parity bits. Events themselves are coded in 11 bits at the end of block C.

![Figure 2-67: RDS-TMC - Typical format of an RDS message according to IEC 62106:2000](image)

If one of these groups is not enough to transmit all information, it is possible to define at most 5 such groups to a continuous message. They are not nested, but sent one after the other containing information about the order.

RDS – TMC messages contain 2 types of information: basic and optional items. Basic items are used in every message, whereas optional information can be added to the messages if necessary.

The event list defines for each event the following attributes which are hard coded within the event code:

- Pre – assigned diversion advice (is it recommended to take another route)
- Direction (does the event effect only one or both directions)
- Duration type (is it a short term or long term event?)
- Urgency within the terminal (when shall the message be presented to the user)
- Nature (is it an existing event or is it a forecast)
- Update class (necessary for message management)
- Quantifier (speed limits, affected length, …)

If the definition in the event code does not match the current situation to describe, an additional message can be created to override this static definition.

Additionally it is possible to add more information to one message by linking more messages together. These additional messages can contain information like an additional event, a speed limit advice or an explicit stop time for the event.

Portrayal Catalogue

There are no such rules for events.

Service Reference

The content is provided by RDS-TMC service.
2.6.4 Scheduled Road Works (SRW)

Scheduled Road Work data are part of the pavement management system of the Highway Agency (HAPMS) in UK. An overview of the standard is available in the main document at subsection 3.1.16.

Feature Catalogue

The data model contains information about road works administrated in the pavement management system of the Highway Agency in the UK. Road work events are named as “Closures” in the data model. The aim of SRW is to generate a standardised closure_transfer_file with information about road works. There are two types of road work description. On one hand aggregated information about road works for the generation of traffic messages can be defined, while on the other hand more detailed information about the road work layout can be provided for traffic management and for traffic forecast service operators.

Scheduled Road Works (SRW) closure records (unpublished_description) come in two forms - Summary Closures and Detailed Closures. Both types of records have some common features and data attributes. All closure records, if correctly constructed, contain references to the HAPMS network sections which they affect. Summary Closures do not provide details of exact timings or lane configurations, but provide global indicative measures that help comprehension. Detailed closure provide a full lane-space model throughout the life of the closure.

The data model is shown in the following figure.
Figure 2-68: SRW - Data Model
The main object of the closure transfer file is *clos*, containing a detailed or summary description of road works. Common attributes of all closure data are information about the status, the publication date and flag.

**clos**

<table>
<thead>
<tr>
<th>Specialises</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Describes a partial or total road closure caused by scheduled road works.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>closure_key</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>closure_status</td>
<td>Closure_Status_Enum: Completed, Firm, Provisional</td>
<td>Completed indicates that the information held against the lane closure can be considered as an accurate record of the actual lane closure. Firm indicates that the lane closure are either expected to take place as described or are taking place. Provisional indicates that planned works are expected but the details are either liable to change or are incomplete.</td>
</tr>
<tr>
<td>publish_end_date</td>
<td>DateTime</td>
<td></td>
</tr>
<tr>
<td>publish_start_date</td>
<td>DateTime</td>
<td></td>
</tr>
<tr>
<td>publish_flag</td>
<td>Publish_Flag_Enum: N, X, Y</td>
<td>N = No, X = Cancelled, Y = Yes</td>
</tr>
</tbody>
</table>

There are two types of information about a closure. A *published_description* contains all information that are relevant to generate a road work message for traffic information services. *Unpublished_description* contains more details about the layout of the roadwork site only for internal use within the road administration.

**published_description**

<table>
<thead>
<tr>
<th>Specialises</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clos_end_date</td>
<td>DateTime</td>
<td>Time/date at which the closure overall ends</td>
</tr>
<tr>
<td>clos_start_date</td>
<td>DateTime</td>
<td>Time/date at which the closure overall starts</td>
</tr>
<tr>
<td>effect</td>
<td>Delay_Enum: Moderate (10 – 30 minutes), No Delay, Severe (more than 30 mins), Slight (less than 10 mins)</td>
<td>Expected delay caused by road work</td>
</tr>
<tr>
<td>length</td>
<td>numeric[12.3]</td>
<td>The length of the closure in metres.</td>
</tr>
<tr>
<td>narrowlanes</td>
<td>Yes_No_enum: Yes, No</td>
<td>Whether narrow lanes are in use?</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>publish</td>
<td>Yes_No_enum: Yes, No</td>
<td>Indicates whether the closure should be made available through the Highway Agency’s dissemination channels to the travelling public.</td>
</tr>
<tr>
<td>road_no</td>
<td>Character[20]</td>
<td>The reference road that the closure is considered to be on (for error checking). The list of permissible roads is taken from HAPMS.</td>
</tr>
<tr>
<td>tempspeedlimit</td>
<td>Temp_Speed_Limit_enum: 10mph, 20mph, 30mph, 40mph, 50mph, 60mph, Unchanged</td>
<td>Indicates the temporary speed limit</td>
</tr>
<tr>
<td>text_loc</td>
<td>character[75]</td>
<td>A human understandable free text description of location of the works, to be used in combination with road_number.</td>
</tr>
<tr>
<td>type</td>
<td>Closure_Type_enum: Cancelled, Emergency Works, Event, Incident, Planned Works</td>
<td>Indicates the status of the closure. Emergency Works refers to lane closures put in place by the Service Provider to make safe or make repairs to damage resulting from an incident on the network. Event indicates some form of event that may affect normal traffic flows. Events are used in conjunction with major traffic generating events (eg. major sporting events) or if the route concerned is part of an alternative or diversionary route. Incident refers to unplanned lane closures directly resulting from an incident on the network. Incidents normally involve road traffic collisions, disabled, damaged or abandoned vehicles, obstructions in the carriageway, or significant unplanned roadside events that are causing disruption to the normal flow of traffic Planned Works indicates traffic management modifying the normal flow of traffic in relation to works planned by the Service Provider to undertake their normal activities on the network. This includes all items of routine and non-routine maintenance, refurbishment and construction but does not relate to works being performed to rectify damage to infrastructure as the result of an incident although some more extensive repairs may be the subject of programmed activities.</td>
</tr>
</tbody>
</table>

Unpublished_description contains information about the reason and type of the road works and about the traffic management measures activated in the area of the work site. These information will be extended by a detailed or summary description of the road works.
<table>
<thead>
<tr>
<th>Description</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>contract_type</td>
<td>Contract_Type_enum</td>
<td>The type of contract for the closure.</td>
</tr>
<tr>
<td>reference</td>
<td>character[75]</td>
<td>Contract number/reference number given to the Highway Agency by the contractor</td>
</tr>
<tr>
<td>internal_notes</td>
<td>character[2000]</td>
<td>Internal notes relating to the closure - not for public use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The National Traffic Control Centre look in the Notes field for indications of whether an Advanced Warning Plan is requested.</td>
</tr>
<tr>
<td>login</td>
<td>character[10]</td>
<td>Username of the last editor of the closure record.</td>
</tr>
<tr>
<td>modified</td>
<td>DateTime</td>
<td>The date and time at which this record was last saved.</td>
</tr>
<tr>
<td>ref</td>
<td>character[75]</td>
<td>An internal reference number given to the closure</td>
</tr>
<tr>
<td>summary_or_detail</td>
<td>Summary_Detail_enum: Detail, Summary</td>
<td>Indicates (switches) the level of detail of the information to be entered about the closure</td>
</tr>
<tr>
<td>traf_man</td>
<td>Traffic_Management_enum: Carriageway_Closure, Contraflow, Convoy_Working, Height_Restriction, Lane_Closure, Lane_Closure_With_Switching, Mobile_Lane_Closure, Other, Speed_Restriction, Stop/Go_Boards, To_Be_Advised, Traffic_Signals, Weight_Restriction, Width_Restriction</td>
<td>Type of Traffic management for the closure</td>
</tr>
</tbody>
</table>
The **summary_closure** gives general information about the number of lanes that are closed / in operation during the road works and if the road works happen during night, peak or off-peak period.

<table>
<thead>
<tr>
<th>Summary_closure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specialises</td>
</tr>
<tr>
<td>Description</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hardshoulder_only</td>
<td>Yes_No_enum: Yes, No</td>
<td>Indicates whether the closure only affects the hardshoulder</td>
</tr>
<tr>
<td></td>
<td>Lanes_enum: Five_Lanes, Four_Lanes, Not_Specified, One_Lane, Six_Lanes, Three_Lanes, Two_Lanes</td>
<td>this is the number of lanes that been closed.</td>
</tr>
<tr>
<td></td>
<td>Lanes_enum: Five_Lanes, Four_Lanes, Not_Specified, One_Lane, Six_Lanes, Three_Lanes, Two_Lanes</td>
<td>this is the number of additional lanes that have been added over and above the normal running lanes. Opening the hard shoulder is included in this total.</td>
</tr>
<tr>
<td>Night period</td>
<td>Yes_No_enum: Yes, No</td>
<td>Indicates that the works are taking place in the nighttime period (7pm-7am).</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Offpeak period</td>
<td>Yes_No_enum: Yes, No</td>
<td>Indicates that the works are taking place in the off-peak period - between 10:00 to 16:00 weekdays and 07:00 to 19:00 at weekends</td>
</tr>
<tr>
<td>Peak period</td>
<td>Yes_No_enum: Yes, No</td>
<td>Indicates that the works are taking place in the peak period - between 07:00 to 10:00 and 16:00 to 19:00 on weekdays</td>
</tr>
</tbody>
</table>

Detailed closure gives more accurate information about the layout and the diary of the road work site. This information can be used to generate, for example, a congestion forecast for road works.

A layout represents a traffic management configuration that occurs at one specific time for a closure. Two layouts cannot occur at the same time. A layout may include elements from all the closures components. Layout Lane defines the status of a length of a lane for a particular Layout. The lanes are delimited by the bounds of the Components defined for the Closure.

All information in unpublish_description are referenced to a component, defined as a continuous length of a road within a closure. (for example, Central Reserve Works would typically have at least one component in each direction). All components consists of HAPMS road sections. Starting point and end points of the section are defined by the chainage.
Figure 2-69: HAPMS network - Data Model
Encodings
No information available.

Portrayal Catalogue
No information available.

Service Reference
No information available.

2.6.5 Comparison and Evaluation
In the area of traffic messages there are currently three international standardisation that are relevant for eMOTION. DATEX 2 covers all types of traffic messages and is at the moment used to communicate traffic messages from Traffic Management Centres. DATEX 2 is based on the data dictionary defined in DATEX 1 and in Alert-C. Also DATEX 2 uses the referencing systems of Alert-C and TPEG-loc.

The second important standard is RDS/TMC and Alert-C. These two standards are well established European standards in the area of traffic messages. Alert-C was not developed to support XML-based data communication but to broadcast messages via radio. Alert-C is based on the TMC location code list, that is currently only available in the freeway network and not in the urban road network.

A further development of Alert-C is TPEG. TPEG is the DAB/interent equivalent of RDS/TMC and it should be possible to make a mapping between TPEG and RDS/TMC data.

The SDEP and SRW standards are national data models from UK. Both cover data that are also available via DATEX 2 or RDS/TMC or TPEG data streams. Because they are not based on a international standardised location referencing system it should not be part of the eMOTION data model.

In the area of traffic messages the eMOTION data model should be based on DATEX 2, because DATEX 2 is XML-based and uses international location referencing systems like TPEG and the TMC location codes. DATEX 2 is able to communicate Alert-C events and could also be the data model for the back office data communication of a TPEG-DAB-service.

2.7 Parking
Parking information comprises static information like tariffs, opening times, etc and dynamic information about occupancy or vacancy for parking quarters, single parking facilities or all parking facilities near a specific address for a specific validity period.

2.7.1 DATEX 2
The main aim of DATEX 2 is to establish standardised links between Road Operators, TICs
and Service Providers to allow the easy and efficient communication of traffic and traveller information. An overview of the standard is available in the main document at subsection 3.1.2.

There is only small relevance of DATEX 2 in the area of parking information. It is possible to communicate some dynamic data regarding the occupancy of parking facilities with DATEX 2 in the *SituationPublication*.

**Feature Catalogue**

One part of *NonRoadEventInformation* is dynamic information about the occupancy of parking facilities.

**Figure 2-70: DATEX 2 - NonRoadEventInformation: composite structure diagram**
### carParkOccupancy
- **Type**: Percentage
- **Description**: The percentage value of car parking spaces occupied.

### carParkStatus
- **Type**: CarParkStatusEnum
- **Values**: CarParkClosed, allCarParksFull, carParkFacilityFaulty, carParkFull, carParkStatusUnknown, enoughSpacesAvailable, multiStoryCarParksFull, noMoreParkingSpacesAvailable, noParkAndRideInformation, noParkingAllowed, noParkingInformationAvailable, normalParkingRestrictionsLifted, onlyAFewSpacesAvailable, parkAndRideServiceNotOperating, parkAndRideServiceOperating, specialParkingRestrictionsInForce
- **Description**: Indicates the status of one or more specified car parks.

<table>
<thead>
<tr>
<th>exitRate</th>
<th>Type: VehiclesPerHour</th>
<th>Description: The rate at which vehicles are exiting the car park.</th>
</tr>
</thead>
<tbody>
<tr>
<td>fillRate</td>
<td>Type: VehiclesPerHour</td>
<td>Description: The rate at which vehicles are entering the car park.</td>
</tr>
<tr>
<td>numberOfVacantParkingSpaces</td>
<td>Type: NonNegativeInteger</td>
<td>Description: Indicates the number of vacant parking spaces available in a specified parking area.</td>
</tr>
<tr>
<td>occupiedSpaces</td>
<td>Type: NonNegativeInteger</td>
<td>Description: Number of currently occupied spaces.</td>
</tr>
<tr>
<td>queueingTime</td>
<td>Type: Time</td>
<td>Description: The current queuing time for entering the car park.</td>
</tr>
<tr>
<td>totalCapacity</td>
<td>Type: NonNegativeInteger</td>
<td>Description: Total number of car parking spaces.</td>
</tr>
</tbody>
</table>

### Encodings

### Portrayal Catalogue
There is no portrayal associated with this standard.

### Service Reference
There is a reference to DATEX 2 data service described in chapter 4.2.3.

#### 2.7.2 ISO 18234-7 / ISO 24530-5: TPEG-PKI - Parking Information
The Transport Protocol Experts Group (TPEG) developed the TPEG specifications for transmission of language independent multi-modal Traffic and Travel Information. An overview of these specifications is available in the main document at subsection 3.1.8.

TPEG-PKI is under development and due to its early stage it is not available at the moment. After the standard approval it should be incorporated into the eMOTION service technical
specification.

**Feature Catalogue**

Not available.

### 2.7.3 UTMC Standard - TS004: Car Park Monitor MIB

The Urban Traffic Management and Control (UTMC) programme is the UK Department for Transport (DfT) main initiative for the development of a more open approach to Intelligent Transport Systems (ITS) in urban areas. An overview of these specifications is available in the main document at subsection 3.1.22.

**Feature Catalogue**

UML schema shown below presents the structure of management information bases (MIB) as these are defined within the UTMC standard.

There have been seven MIBs defined within this standard so far and it is possible to come with new MIB proposals. After the approval process which is also given by UTMC standard the new MIB can be added into this list and it is ready for use by anyone who wants to be in UTMC compliance. Parking facilities have already been covered by UTMC through the Car Park Monitor MIB.

Car Park Monitor MIB contains several classes which are composed from sub-classes containing the data objects defined using ASN.1. For more details see section Encodings.
class: Traps
This class contains attributes CarParkStateTrap, faultAlarm and CarParkFaultRTC which represent the traps. The purpose of these attributes is to indicate that a fault has occurred and to return the appropriate value to the object.

class: General and Identification objects
This class contains the attributes which include important data for user login to systems, general information about network such as IP address. These data are carParkSoftwareVer, carParkPassword, carParkLogOff, carParkTime, carParkIPAddress, carParkPort, carParkMIBVer, carParkSetPassword, carParkRTC.
The function of these attributes is to indicate the version of software, access and authentication information, log off and re-entered access, current time, hold the IP Address and Port Number to which traps are returned, indicate Car Park MIB Version, set the password to be used to log on, set or return the current time.

class: Zone Configuration and data
This class contains the attributes which include data about carParkCapacity, carParkZoneID, carParkOpening/ClosingTime, carParkIncreasing/DecreasingOccupancy, carParkEntrance-Full, carParkTrapTrigger, carParkOccPercent, carParkFillRate, carParkExitRate, carParkQueue, carParkCounterTable, carParkOpeningTable, carParkOpeningEntry, carParkOpeningDay, carParkClose/OpeningOpen.
The function of these attributes is to indicate capacity, opening/closing time of car-park/zone, occupancy and indicate whether its occupancy is increasing or decreasing, occupancy trigger trap/percentage, fill/exit rate, queue, and set openingvalidity time of car-park/zone.

class: Counter Configuration
This class contains the attributes which include data for carParkCounterTable, carParkCounterEntry, carParkCounterNum, carParkCounterEntryExit, carParkCounterStartLoop, carParkCounterLoopType, carParkNumCounters.
The function of these attributes is to configure table unit and its values with respect to monitored counters, indicate states if the counter is an entry or exit counter, number of counters on unit, and detector connected to counter.

class: Faults
This class contains the attributes which include data for carParkFaultTable, carParkFaultEntry, carParkFaultNo, carParkFaultID, carParkFaultType, carParkFaultDate, carParkFaultRTC, and carParkFaultDesc.
The function of these attributes is to provide a table of instances of faults, return the index and identifier, sub-type for a fault entry, type and return the time & date for the fault and return the user string with extended fault information.

class: Historical Data
This class contains the attributes which include data for carParDatakStartTime, carPark-
DataPeriod, carParkDataOccupancy, carParkDataOccPercent, carParkDataFillRate, carParkExitRate, carParkDataOccTrend, carParkDataQueue, carParkDataCounters, CarParkStartRTC, CarParkCounterSetUp, CarParkIndividualCounts.

The function of these attributes is set the start time, period for download of data, average of the occupancy, fill rate, exit rate, estimate of car park/zone queuing, counts over the specified start time and period, set or return the fault time, return data for over the specified start date and period and for the individual loopCounters too.

Encoding

Preferred encoding of data objects to be transmitted may be described from two points of view:

- Transmission between two UTMC centres
- Transmission between UTMC centre and the field equipment

In the first case are communications expected to use the CORBA (Common Object Request Architecture) or XML (eXtensible Markup Language) approach, and should exploit the technical guidance provided by the Travel Information Highway where appropriate.

The General Inter-ORB (Object Request Broker) Protocol (GIOP) specifies a set of message formats and Common Data Representation (CDR). The CDR (and BER) takes care of inter-platform issues such as byte ordering and memory alignment, etc.

The Internet Inter-ORB Protocol (IIOP - Internet Inter-ORB Protocol) specifies how GIOP messages are exchanged over a TCP/IP network.

Data Objects, and their exchange processes, shall be specified in a suitable standard specification language. Acceptable languages include:

- XML (particularly for Internet objects)
- IDL (for additional CORBA services)
- Entity-relationship diagrams (particularly for Database Objects)
- Abstract Syntax Notation 1 ASN.1 (particularly for MIBs). Systems using ASN.1 should use Basic Encoding Rules (BER) for transmitting data in accordance with ISO 8825 and CCITT’s X.209.
- UML

Car Park Monitor management information base contains the following ASN.1 defined objects from which can be formed car park facilities related UTMC messages:

- carParkSoftwareVer
- carParkPassword
- carParkLogOff
- carParkTime
- carParkAFdecreasing
- carParkFincreasing
- carParkFdecreasing
- carParkEntranceFull
- carParkIPAddress
- carParkPort
- carParkMIBVer
- carParkSetPassword
- carParkRTC
- carParkZoneID
- carParkCapacity
- carParkOpeningTime
- carParkClosingTime
- carParkAFincreasing
- carParkTrapTrigger
- carParkState
- carParkOccupancy
- carParkOccPercent
- carParkFillRate
- carParkExitRate
- carParkOccTrend
- carParkQueue
- carParkAutoReset
- etc.

**Portrayal Catalogue**

There is no portrayal associated with this standard.

**Service Reference**

UTMC’s service reference can be connected with almost all types of services dealing with the traffic information transmission.

### 2.7.4 Comparison and Evaluation

In the area of parking information there are three standardisations with relevance for eMOTION. DATEX 2 is mainly used in the area of traffic data and traffic messages but there is only a minor section of the data model containing dynamic data about the current availability of parking places. It is not possible to communicate static information about the parking facilities (e.g. fees and opening time). One advantage of DATEX 2 is the use of other standardised solutions like the referencing system of TPEG and Alert-C.

The TPEG standard will be an important standard in the area of traffic information service. It will also cover parking information but the standard is not yet defined. In future TPEG-pklML should be considered by the eMOTION framework.

At the moment the UTMC standard from the UK provides the most comprehensive data modelling for static and dynamic parking data.

### 2.8 Public Transport Service Data

Static public transport data comprises public transport network data as well as typical service data like timetable information for stop points, public transport lines for specific directions and time period or date.

Dynamic public transport data concern delay times, detours or cancellations.
2.8.1 ENV 12896: Transmodel

Transmodel defines an extensive data model for relevant Public Transport related information. An overview of Transmodel is available in the main document at subsection 3.2.14.

Besides the description of network data, already given in section 2.2.2, here a major relevance is given to passenger information and to Multi-modal Operation / Multiple operator Environments. A description concerning routing can be found in section 2.12.1.

Feature Catalogue

Items included in the feature catalogue takes into account the structuring principles of the TRANSMODEL abstract specification with a hierarchical, hereditary organization of data items that can be subtypes of and inherit characteristics from other data items.

Timetable information

Published in many different forms (booklets, electronic displays etc.), a timetable is produced as a result of the scheduling process. It gives the scheduled passing times (in general departure times) for all VEHICLE JOURNEYs, on all STOP POINTs, for one or possibly several DAY TYPES.

The process of planning describes how timetable information (in particular, passing times) is computed.

Timetabled Passing Times

A timetable has data derived from the scheduling process. Hence a ‘timetable’ entity is not specifically described on the data model. In a real-world implementation this data is typically calculated on demand and cached due to performance reasons. The entity PASSING TIME represents the properties of any passing time for every point in the network. This entity has several subtypes.

TIMETABLED PASSING TIMEs result from the scheduling process, DATED PASSING TIME are PASSING TIMEs that are calculated on a given OPERATING DAY. They can be, in turn: TARGET PASSING TIME (most recent official plan for a DATED VEHICLE JOURNEY, on a POINT IN JOURNEY PATTERN) ESTIMATED PASSING TIME (forecasted MONITORED VEHICLE JOURNEY, on a POINT IN JOURNEY PATTERN) and OBSERVED PASSING TIME (recording of a MONITORED VEHICLE JOURNEY, on a POINT)

For the production of the timetable, TIMETABLED PASSING TIMEs are the values that are taken into account. They are normally generated before the period of operation and remain valid for a long period.

Possible extensions to PASSING TIME may be implemented with attributes such as arrival time, departure time, waiting time.
Figure 2-72: Transmodel – Timetabled passing times

The SERVICE JOURNEY INTERCHANGE entity specifies a planned interchange between two SERVICE JOURNEYs and corresponding STOP POINTs. INTERCHANGE STATUS can be used to store data on the status of actual SERVICE JOURNEY INTERCHANGEs. An
INTERCHANGE STATUS is related to a SERVICE JOURNEY INTERCHANGE and to a
specific OPERATING DAY.

Where the same area is served by different operators, an integrated information system will
typically be provided and maintained by the authority to inform the public on services from
various PT operators. Merged timetables are often produced with no reference to the oper-
ator's scheduling data but using the features of the TIMETABLED PASSING TIME entity.

**Journey Patterns**

Planning activities and data (used not only for operational purposes but also for passenger
information purposes) are supported by the concept of JOURNEY PATTERNs that can be
defined to associate both service and timing patterns. The provided information can then
include the SERVICE PATTERN description and the passing times planned for every STOP
POINT.

The reference data model describes two main alternative practices to deal with the working
pattern definition: a JOURNEY PATTERN defined as a common sequence of STOP POINTs
and TIMING POINTs, or a combination of a SERVICE PATTERN and a TIMING PATTERN,
defined and managed in a separate way.

Here, an ordered sequence of STOP POINTs and an ordered sequence of TIMING POINTs
(at which timing information is scheduled) is considered. TIMING POINTs may be STOP
POINTs or other POINTs.

The working pattern must be related to a ROUTE that is covered. Several JOURNEY PAT-
TERNs may use the same ROUTE.

<table>
<thead>
<tr>
<th>JOURNEY PATTERN</th>
</tr>
</thead>
</table>

**Specialises**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The working pattern of a planned journey. It consists of:</td>
</tr>
<tr>
<td>- an ordered sequence of STOP POINTs to be served on a given ROUTE;</td>
</tr>
</tbody>
</table>
| - an ordered sequence of TIMING POINTs at which timing information is sched-
|uled. These TIMING POINTs may be STOP POINTs or other POINTs. |

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| ID       | A unique ID | Semi-dynamic data; updated at each ser-
|          |      | vice planning phase - In case of DRT ser-
|          |      | vices, planning occurs at a higher fre-
|          |      | quency |
| Name     | A unique 'name' (char string) | |
| Classified as | a ROUTE | The route the JOURNEY PATTERN is re-
|            |      | ferring to |
| Made up by | A set of POINTs IN JOURNEY PATTERN | The actual definition (points) of the JOURNEY PATTERN |
| Made of  | a TIMING PATTERN | |
Figure 2.56: TRANSMODEL - Definition of JOURNEY PATTERNS

**Encodings**

Transmodel is a concept data model and does not mandate any particular implementation at
the logical or physical level. Other Standards are based on Transmodel and uses specific encodings. For example TransXchange is XML based.

**Portrayal Catalogue**

None

**Service Reference**

Data content defined and described in this standard is related with public transport information services.

### 2.8.2 Service Interface for Real Time Information (SIRI)

The Service Interface for Real Time Information (SIRI) specifies a European interface standard for exchanging information about the planned, current or projected performance of real-time public transport operations between different computer systems. An overview of SIRI is available in the main document at subsection 3.2.13.

The content related parts of the SIRI-standard are described in the following compendium. It is divided into two logical parts. The first feature catalogue gives an overview about the different types of timetables mentioned in the SIRI standard. The second part consists of exchangeable monitoring-content and information relevant in the eMotion context.

**Feature Catalogue Timetable**

This Section describes the four types of timetable-deliveries representing the main part of the relevant exchangeable content:

- ProductionTimetable
- EstimatedTimetable
- StopTimetable
- ConnectionTimetable

In view of efficient space-allocation by the UML-diagrams, following reoccurring Elements have been appended below the timetable descriptions:

- DisruptionGroup
- OperationalInfoGroup
- VehicleJourneyInfoGroup

**ProductionTimetable**

The Production Timetable Delivery is a result of the SIRI Production Timetable Service which transmits daily timetables including any planned updates already known at the time of transmission. It is typically used to exchange content during the communication between Schedule-
ing systems and Automated Vehicle Management Systems (AVMS), and also between AVMS systems and intelligent clients of the AVMS system.
This element is also used for the transmission of the planned interchanges between jour-
neys, including information about the linking of vehicle parts through the interchange, such as whether passengers are able to remain seated in the vehicle.

The provision of known updates results in a more accurate data set of journeys for the SIRI Estimated Timetable Service, allowing a more efficient real-time exchange of content.

Following additional descriptions for the elements are mentioned in the prCEN/TS 15531-3 Final Draft from June 2006:

Each *ProductionTimetableDelivery* is made up of *DatedTimetableVersionFrame* elements. There will be status messages for any request that could not be returned.

Each production timetable is returned as a *DatedTimetableVersionFrame* element. Each *DatedTimetableVersionFrame* contains a version of the timetable for a line and direction, and comprises one or more *DatedVehicleJourney* elements.

Each *DatedVehicleJourney* contains an ordered list of *DatedCall* elements representing the sequence of stop calls, as well as other properties that apply to the journey as a whole.

Each *DatedCall* describes the times at a stop. A journey must contain at least two calls.

Each *TargetedInterchange* describes the connections that may be made at a stop to another onwards distributor journey.

**EstimatedTimetable**

The Estimated Timetable is used by the Automated Vehicle Management System (AVMS) to inform interested schedule information systems of the current status of all known vehicle journeys. This enables the schedule information system to provide information for short-term journey planning.

Message content can include:

- Schedule deviations with predictions for future route sections.
- Failure of a vehicle journey.
- Additional journeys.
- Change to the vehicle capacity (passenger load).
- Change to vehicle type/vehicle equipment.
- Platform changes.
- Stop obstruction.
- Path changes.
Following additional descriptions for the elements are mentioned in the prCEN/TS 15531-3 Final Draft from June 2006:

Each **EstimatedTimetableDelivery** is made up of **EstimatedVehicleJourney** elements. Status messages exist for any requests that could not be fulfilled.

Each production timetable is returned as an **EstimatedTimetableVersionFrame** element. Each **EstimatedTimetableVersionFrame** comprises one or more **EstimatedVehicleJourney** elements.
ney elements grouped for a version of the timetable.

Each EstimatedVehicleJourney contains an ordered list of EstimatedCall elements, as well as other properties.

There are three different ways to match the journeys described in the real-time data to the planned schedule. In the diagram above they are displayed as DatedVehicleJourneyRef, DatedVehicleIndirectRef and EstimatedVehicleJourneyCode. For a detailed description of the differences, please see the SIRI standard documentation.

Finally each EstimatedCall describes the times at a stop. A journey must contain at least two calls.

StopTimetable

The Stop Timetable represents a stop-centric view of timetabled vehicle arrivals and departures at a designated stop and provides a data feed of the static timetables.
The StopTimetableDelivery returns timetabled Stop Visits for a stop, with scheduled times for each arrival or departure from the stop.

A StopTimetableDelivery is made up of TimetabledStopVisit instances, each representing a call at the stop by a vehicle.

Each TimetabledStopVisit contains a set of subelements describing the vehicle’s visit to the stop, including scheduled times; information about the journey may optionally be included in a VehicleJourneyInfo Group.

Each TimetabledStopVisit describes a call at a stop point by a vehicle making a timetabled vehicle journey, including scheduled passing times.
Each *TargetedVehicleJourney* describes a vehicle journey timetable according to either the original or production timetable, including call details.

Each *TargetedCall* describes a call at a monitoring reference point by a vehicle making a timetabled vehicle journey, including scheduled passing times.

A *TimetabledStopVisitCancellation* describes a cancellation of a previous *TimetabledStopVisit*.

**ConnectionTimetable**

The *ConnectionTimetable* represents the feeder arrivals to a specified connection link.
A **ConnectionTimetableDelivery** is made up of zero, one or many **TimetabledFeederArrival** elements, each representing a distinct planned interchange between vehicle journeys over which passengers may transfer.

Each **TimetabledFeederArrival** describes an inbound connection, together making up a connection timetable for the **ConnectionLink**.

Each **FeederJourney** describes information about the connecting journey.

Each **TimetabledFeederArrivalCancellation** describes the cancellation of a planned feeder connection.
DisruptionGroup

The DisruptionGroup provides optional data about real time disruptions of a vehicle journey.

Figure 2-77: SIRI - DisruptionGroup

OperationalInfoGroup

The **OperationalInfoGroup** provides optional operational entities associated with a vehicle making a journey. It includes an **OperationalBlockGroup**.

Figure 2-78: SIRI - OperationalInfoGroup
VehicleJourneyInfoGroup

The **VehicleJourneyInfoGroup** provides optional information about a vehicle journey. It includes a **ServiceInfoGroup**.

![VehicleJourneyInfoGroup Diagram](image)

**Figure 2-79: SIRI - VehicleJourneyInfoGroup**

Feature Catalogue Monitoring

As mentioned above, SIRI provides three different types of monitoring services:

- StopMonitoring
- VehicleMonitoring
- ConnectionMonitoring

Due to the scope of intermodal routing and the emphasis of the public transport to individual traffic intersections, only the StopMonitoring seems to be relevant in the eMotion context.

StopMonitoring

The SIRI Stop Monitoring Service provides a stop-centric view of vehicle arrivals and departures at a designated stop. It can be used by displays and other presentation services in conjunction with the SIRI Stop Timetable service.
Following additional descriptions for the elements are mentioned in the prCEN/TS 15531-3 Final Draft from June 2006:
The **StopMonitoringDelivery** returns Stop Visits for a stop, with scheduled and predicted times for each arrival or departure from the stop. It is made up of **MonitoredStopVisit** instances, each representing a call at the stop by a vehicle, and/or **MonitoredStopVisitCancellation** instances to remove **MonitoredStopVisit** instances.

If the Vehicle Journey is a Train made up of several parts that may merge or join, there should be a separate **MonitoredStopVisit** for each train part. Each **MonitoredStopVisit** contains a set of subelements describing the vehicle’s visit to the stop, including scheduled and/or real-time times, and also a **MonitoredVehicleJourney** element, which describes the vehicle’s route and associations with the **DatedVehicleJourney** that it follows, and other entities. The **Note** element allows one or more arbitrary text strings to be associated with the whole stop, a vehicle journey or an individual journey pattern.

Each **MonitoredStopVisit** describes a call at a stop point by a vehicle making a monitored vehicle journey, including scheduled and/or predicted real-time times. It has a **MonitoredVehicleJourney** element, which associates the **MonitoredStopVisit** with the **MonitoredVehicleJourney** being made by the Vehicle arriving at the stop.

The **PreviousCall** element describes a call which has already been made in the **MonitoredVehicleJourney**.

The **MonitoredCall** element describes the call be made at a Stop point of the **MonitoredVehicleJourney**.

The **OnwardCall** element describes a call which has still to be made in the **MonitoredVehicleJourney**.

Passengers do not expect vehicles that have already arrived at the stop to continue to be shown on a display. The **MonitoredStopVisitCancellation** element is used to clear down an earlier **MonitoredStopVisit**, i.e. to indicate that it can be removed from a display. This can be useful if only updates are being exchanged.

Each **StopLineNotice** describes a text notice relevant to a stop point about a line.

The **StopLineNoticeCancellation** element is used to clear down an earlier **StopLineNotice**, i.e. to indicate that a notice is no longer current and can be removed from a display.

**Encodings**

The SIRI Service Interfaces are built by XML-messages for data exchange.

**Portrayal catalogue**

None

**Service reference**

SIRI itself is defined as a service interface, see its description in 4.2.4.

Service instances of the standard have been deployed in Berlin, Hamburg, West Yorkshire,
2.8.3 Transport Exchange (TransXChange)

TransXChange is the UK national de facto standard for the exchange of bus route and timetable information. An overview of TransXChange is available in the main document at sub-section 3.2.15.

Feature Catalogue

The essential features to describe service information will be described in this section.

Within NaPTAN stops are defined using the NaPTAN Stop Model.

Standard Service

Standard Bus services are described using three distinct layers: Route, Journey Pattern and Vehicle Journey.

The Route is described as a sequence of RouteLinks connecting individual StopPoints in NaPTAN space. RouteLinks may be grouped into RouteSections to allow reuse in other routes. The RouteLinks may be projected onto geospatial coordinate systems using Track elements.

Journey Pattern is the path over the Route made up of Journey Pattern Timing Links, each with timing information (all relative times) and optional operational information. Journey Pattern Timing Links ends may have stop activity and other service information assigned using a Journey Pattern Stop Usage element. Journey Pattern Sections allow for the reuse of timing links in other Journey Patterns. A Journey Pattern must traverse the complete or a contiguous subset of links in the associated Route.
Vehicle Journey is a traversal of a specific Journey Pattern at a specific time, using Vehicle Journey Timing Links and Vehicle Journey Stop Usage elements. Each Vehicle Journey is assigned an absolute departure time, which is combined with timing information from links to define actual passing times. Vehicle Journeys are publicly identified by a Line. One or more Lines may be associated with a Service. The link sequence of the Vehicle Journey must exactly correspond to the underlying Journey Pattern.

Flexible Service

TransXChange may also represent flexibly routed services, which operate on flexible routes with fixed stops and/or spatial zones (NaPTAN Stop Points of type FlexibleZone) which may be visited in any order by the Service. A Service may contain both FlexibleService and StandardService components. Interchange elements defined to describe the transition between the flexible and fixed service.
A FlexibleService is described by one or more FlexibleJourneyPatterns, which describe stop usages and Booking Arrangements. FlexibleVehicleJourneys assigned to the FlexibleJourneyPattern contain additional information on operating times.

**Interchanges**

The Interchange model is used to specify the connection between vehicle journeys. Connection between JourneyPattern instances is specified by a JourneyPatternInterchange. This interchange between inbound and outbound journey may be different NaPTAN stops representing a transfer, with the mode of the transfer being indicated. VehicleJourneyInterchange, which projects onto a JourneyPatternInterchange, specifies connection between two specific VehicleJourneys. Certain VehicleJourneyInterchange properties are inherited from the assigned JourneyPatternInterchange.
Encodings
TransXChange describes an XML schema, suitable for exchange of TransxChange data in XML format.

Portrayal Catalogue
No portrayal associated.

Service Reference
TransXChange make use of the NaPTAN Stop model, and is used for the exchange of data used within the JourneyWeb protocol. TransXChange is based on TransModel.

2.8.4 Railway Markup Language (RailML®)
RailML.org is a development partnership of independent businesses and institutions. One goal of railML is to designate the totality of all XML schemas that have been written for railway data, including of all the individual formats. An overview of railML and detailed information about railML Group is available in the main document at subsection 3.2.10.

Feature Catalogue
The actual complete RailML Specification can be downloaded from the RailML Website: http://www.railml.org.
Timetable

The diagram below gives an overview to railML timetable. To keep it readable the not relevant parts (categories, compositions) are not included in detail.

Figure 2-84: railML - Timetable

Timetable information - focussed on traveller information services - exchanged with railML systems can be structured as follows:

- Train
• Timetable periods
• Operating periods

### Train

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>trainId</td>
<td>string</td>
<td>identification of train course</td>
</tr>
<tr>
<td>trainGroupID</td>
<td>string</td>
<td>grouping different trains, belonging to one (published) train number for the course</td>
</tr>
<tr>
<td>lineID</td>
<td>string</td>
<td>identification of the traveled line</td>
</tr>
<tr>
<td>intervalGroupID</td>
<td>string</td>
<td>grouping different trains belonging to a cycle of trains during the day</td>
</tr>
<tr>
<td>trainNumber</td>
<td>string</td>
<td>train number</td>
</tr>
<tr>
<td>kind</td>
<td>string</td>
<td>train kind</td>
</tr>
<tr>
<td>type</td>
<td>enumeration</td>
<td>timetable kind</td>
</tr>
<tr>
<td>trainStatus</td>
<td>string</td>
<td>requested, planned, canceled,…</td>
</tr>
<tr>
<td>debitcode</td>
<td>integer</td>
<td>customer reference number</td>
</tr>
<tr>
<td>remarks</td>
<td>string</td>
<td>individual information</td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>train description</td>
</tr>
<tr>
<td>seatCapacity</td>
<td>integer</td>
<td>number of seats or passengers</td>
</tr>
<tr>
<td>kindNo</td>
<td>string</td>
<td>train kind number used by (German) DB</td>
</tr>
<tr>
<td>timetablePeriodID</td>
<td>string</td>
<td>indicating the timetable period</td>
</tr>
<tr>
<td>categoryID</td>
<td>string</td>
<td>indicating the train category (priority,…</td>
</tr>
<tr>
<td>dataSource</td>
<td>string</td>
<td>source</td>
</tr>
<tr>
<td>dataDateTime</td>
<td>dateTime</td>
<td>former field “date”</td>
</tr>
<tr>
<td>dataStatus</td>
<td>string</td>
<td>real, planned, unreasonable,…</td>
</tr>
</tbody>
</table>

### TimetablePeriod

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timetablePeriodID</td>
<td>string</td>
<td>timetable period identification</td>
</tr>
</tbody>
</table>
OperatingPeriods

| Description | The element “operatingPeriods” summarizes all regular (daily, Mo-Fr,...) or irregular (Sundays and holidays, every day but not on xxx,...) days of operation, called operating periods or services. |
| Property | Type | Description |
| service | Complex | all kind of operating services |
| holidays | Complex | holidays, to be referenced in some of the services |

Rolling stock

From the eMOTION point of view only “passenger” as subtype of wagon (vehicle) is important. All other features e.g. engine or brake were out of the scope.

The RailML rolling stock specification describes no details about the multiplicities and therefore the UML diagram below does not contain this information.
Figure 2-85: railML - Rollingstock

<table>
<thead>
<tr>
<th>passenger</th>
<th>wagon</th>
</tr>
</thead>
</table>

**Specialises**

**Description**

Data for passenger cars, description of any additional service provided per vehicle.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>class1</td>
<td>int</td>
<td>number of first class seats per vehicle</td>
</tr>
<tr>
<td>class2</td>
<td>int</td>
<td>number of standard class seats per vehicle</td>
</tr>
<tr>
<td>lowFloorPortion</td>
<td>decimal</td>
<td>percentage of low floor portion per vehicle</td>
</tr>
<tr>
<td>catering</td>
<td>boolean</td>
<td>flag, whether the vehicle comprises catering service</td>
</tr>
<tr>
<td>baggage</td>
<td>boolean</td>
<td>flag, whether the vehicle comprises a baggage compartment</td>
</tr>
<tr>
<td>deck</td>
<td>string</td>
<td>number of decks per vehicle</td>
</tr>
<tr>
<td>controltrailer</td>
<td>boolean</td>
<td>flag, whether the vehicle comprises driving cab</td>
</tr>
<tr>
<td>tilting</td>
<td>boolean</td>
<td>flag, whether the vehicle comprises tilting capabilities</td>
</tr>
</tbody>
</table>

**Encodings**

The standard is based on XML schemes and uses XML documents for data exchange.
Portrayal catalogue
None

Service reference
unknown

2.8.5 Rail Journey Information System (RJIS)
The RJIS Datafeeds Interface Specification for Timetable Information describes the extraction format for exchange of data with the UK Rail Journey Information System. An overview of RJIS is available in the main document at subsection 3.2.9.

Feature Catalogue

BASIC TIMETABLE DETAIL
The Basic Timetable Detail File contains: Schedule Records contain sets of train schedules; Association records document the link between a pair of trains (for example where a train splits to form 2 services, or crosses a national border); and UK domestic TIPLOC (Timing Point Location) codes.

![Figure 2-86: RJIS - Basic Timetable Detail Data Record](image-url)
TRAIN SCHEDULE

The Train Schedule section of the file contains: a Basic Schedule which describes general operating details of the service; Extra Details including additional data for compatibility with external systems; Origin; Intermediate; and Terminating Station for the service, and; Changes En Route, which describes scheduled changes to the operating details of the service. Relevant locations are described using TIPLOC codes.

**Figure 2-87: RJIS - Train Schedule Record**

**Encodings**

Data is exchanged in CIF (Common Interchange Format) format, an ASCII file format of fixed length records.

**Portrayal catalogue**

There is no portrayal information associated with this standard.

**Service reference**

There is no service associated with this standard.

**2.8.6 ISO 18234-5 / ISO 24530-4: TPEG-PTI - Public Transport Information**

The Transport Protocol Experts Group (TPEG) developed the TPEG specifications for transmission of language independent multi-modal Traffic and Travel Information. An overview of these specifications is available in the main document at subsection 3.1.8.
The TPEG-PTI application is designed to transfer information from the service provider’s database to an end-user’s equipment. Public Transport Information Application is intended to cover all modes of public transport (i.e. collective) as well as inter-urban and intra-urban travel. It is designed to allow the efficient and language independent delivery of public transport information directly from service provider to end-users.

**Feature Catalogue**

![Figure 2-88: TPEG-PTI - Application Event/Status PTI Container](image)

This diagram presents the general structure of TPEG messages. The Message management container and Location container are described in chapter 2.6.2. The Application Event/Status PTI container is covered by ISO 18234-5 and ISO 24530-4 and deals with public transport information data content definitions.

Level one classes in TPEG-PTI, such as transport_mode, message_report_type and service_information, denote principle categories which group relevant content descriptions. They are intended to present public transport information to end-user with an overview of the message. In most cases level one classes have sub-divisions allowing for further refinement of the relevant descriptions. This methodology enables a service provider to deliver information to a degree to which details are available and are judged useful or relevant. At the other end of the delivery chain, a TPEG-decoder is equally able to present available information to the level required.
This concept of a message using several levels may be illustrated by considering this example of a message:

“A landslide is blocking a railway line”

Route Description

<table>
<thead>
<tr>
<th>Level 1</th>
<th>Railway Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 2</td>
<td>Cancelled</td>
</tr>
<tr>
<td>Level 3</td>
<td>Environment Event</td>
</tr>
<tr>
<td>Level 4</td>
<td>landslide</td>
</tr>
</tbody>
</table>

The minimal element in the example would tell a TPEG-decoder that there was a railway route message. The next level would then add the fact that services were cancelled. An additional detail such as the problem was environmental and in fact a landslide could be discovered by looking at further refining levels.

Suppliers of TPEG-decoders have the liberty to present information to whatever level they feel appropriate. A service provider, similarly, can choose the amount of detail to send. The hierarchical approach is useful for distribution on bearers of different capacity. High capacity ones may carry all levels of the message, whilst lower ones may carry fewer levels, but still retain the essential (top level) elements of a message.

The hierarchical coding structure is future-proofed, since it allows a new message component to be added and used by a service provider. Existing TPEG-decoders, although not able to decode it, can represent the message component fully to one level higher.

The method also results in efficient coding where many of the event tables can be used under several level-one class messages. This reduces the number of reference tables that need to be stored in the TPEG-decoder.

Note

All following UML diagrams in this section (Figure 2-89 to Figure 2-94) are copied from [http://www.itsregistry.org.uk/](http://www.itsregistry.org.uk/).
The `public_transport_information` class represents public transport information from TPEG PTI. It is intended to convey information to public transport users. The information provided relates to the event and some status information on the public transport network and on the associated infrastructure affecting any "leg" of a public transport journey. PTI has a fairly flat structure, with only some hierarchical elements to allow for the creation of messages from a set of PTI tables, which are essentially word-oriented and cover most needs.

The attributes of the `public_transport_information` class give a general description of the `public_transport_information` as the `version_number`, `the_start_time` and `end_time`, etc. The class has five aggregated classes which are the `message_report_type`, the `location_container`, the `service_information`, the `transport_mode` and `additional_information`. Most of these classes have further associated classes which complete the TPEG PTI model.

The `message_report_type` is mainly for presentation purposes. It allows the indication of the type of presentation the service provider had in mind when the message was generated. Possible values of the `message_report_type` are `unknown`, `incident`, `station-terminal_view`, `route`, `individual_service_view` or `undefined`.

The `location_container` holds all information pertaining to the location of an event, be that a point, an area, a stretch, or some other kind of location. It is a complete description of location and all tpegML applications describing a location make use of this container.

The `additional_information` class is used to provide further details to the end-user, including perhaps such things as "hypertext links".

The `service_information` element is used to describe the service in further detail as well as what is happening. It contains `transport_service_identification`, `service_condition`, `facilities`,...
booking_status, ticket_restrictions, severity, event_reason, timetable_type and route_description. The service_condition element provides information such as delayed, cancelled and replacement transport. The facilities element enhances the information by describing a service more fully and by providing more amenities, for example it is possible to indicate whether a service can be used by disabled people or travellers with bicycles. The booking_status element indicates whether a service is fully booked or if there is space available. Ticket_restrictions indicate permitted ticket types for a particular service.

Most these classes are described by an enumeration which lists all possible values of the individual class types.

The transport_service_identifier as a associated class from service_information is used to identify a particular service. It contains the attributes transport_service_id and transport_information_type. These elements allow the use of either a Service Number, such as a flight number or a Service Name, such as the name of a vessel, or a particular route. They can also be used in combination.
The `transport_service_identifier` owns three associated classes which are the `operator_name`, the `operator_subsidary_name` and the `transport_service_name`.

The `operator_name` is one way of identifying who is running a service. This element identifies the provider/operator of a service. The information is provided as short strings, which should reflect what users are likely to see on local signage. The `operator_subsidary_name` identifies a subsidiary of a provider/operator in the case of code sharing, or where a parent company provides different services through different subsidiaries. The `transport_service_name` gives the name of the service.
The event_reason as another associated class from service_information gives a broad reason why something is happening. There are four types of event_reasons which are further specified in the event_reason_subtype. These four types of event_reasons are general problems, personnel problems, equipment problems and environment problems. Each of these event_reasons is specified in an enumeration which lists all aspects of these categories. The detailed list of event_reasons is shown in Figure 2-92 above.

**Figure 2-92: class diagramm “event_reason” with descriptive enumerations**
The transport_mode element is used to indicate which mode of transport is affected by the message. It contains transport_mode and transport_submode. These elements allow the description of the mode of transport. The mode can be refined by use of the submode descriptor. Possible transport_submodes are unknown, railway service, coach service, suburban railway service, urban railway service, metro service, underground service, bus service, trolleybus service, tram service, water transport service, air service, ferry service, telecabin service, funicular service, taxi service, etc.

![transport_submode diagram](image)

**Figure 2-93: transport_submode diagram**

The class route_description as part of the service_information is used to describe the route
of a service. The route description will allow a user to filter on messages based on any number of locations where a service might or might not be available. For greater granularity, the `service_delivery_point` permits differentiation between a scheduled or predicted delivery point. The route description is specified by `route_description_type` which has access to additional information as `time` and `timetable` information.

![route_description diagram](image)

**Figure 2-94: route_description diagram**

**Location referencing**

Messages relate to real-world objects that are called *Locations*. Each message requires a description of the location, to which a message relates. TPEG-decoders without navigational systems or digital maps will require the location to be expressed as character strings, using familiar place names such as town names, rail terminals or airport names to be presented either as text or speech to the end user. Intelligent systems, such as digital mapped-based TPEG-decoders, require the location information to be expressed in a machine-readable coded form. Within the *Location Coordinates* elements, TPEG-Loc combines both requirements in a way that permits machine interpretation while simultaneously ensuring useful,
human understandable, analogues are available. More detailed information about Location referencing (TPEG-Loc) can be found in chapter 2.4.6.

**Encoding**

TPEG messages are exchanged in two formats. The binary transmission format is described in the CEN / ISO TS 18234 series, while the XML transmission format is described in the CEN / ISO TS 24530-5.

**Portrayal Catalogue**

There is no portrayal associated with this standard.

**Service Reference**

TPEG-PTI is designed for use within TPEG services, as defined in ISO TS 18234 and ISO TS 24530.

**2.8.7 OpenTravel Alliance (OTA) Standard and Ferry XML**

The standardisation of air traffic on an international level is organised by the OTA Consortium, which is responsible for these topics. OTA standards are service standards – the content is provided as part of the response of a service interface. The content interesting for eMOTION covers mainly the description of public transport plans. For a description of the OTA service standard see 4.5.3.

Similarly, based on a service model Ferry XML covers the means of ferry traffic on a European level. For a description of Ferry XML see 4.5.4.

An overview of the OTA standard is available in the main document at subsection 3.5.11, Ferry XML can be found in the main document at subsection 3.5.3.

**2.8.8 Comparison and Evaluation**

First we describe data relating to local transport, such as buses, trams, metro and local rail. For traditional scheduled services these include:

- Service including line number, mode of transport, operator, days of operation (which are expressed by different methods) and validity period.
- Origin, destination and intermediate Stops (projected onto Stop Points within the public transport network model) linked by a Route.
- Route Links (which make up the route) projected onto Tracks
- Activities at each Stop - Pick Up, Set Down or Pass
- Relative Timing information between Stops/Timing Points, including departure and/or arrival or pass through times
- Absolute Times of vehicles, expressed as explicit vehicle departures, or service fre-
From the end user point of view, such information is traditionally presented as a timetable and route map. Ability to render such material on request may be complimented by the ability to link to pre-rendered information. OTA, RailML, RJIS, SIRI, Transmodel and TransXChange all contain the information required to assemble static timetables, although their underlying models differ. Transmodel describes a layered model to describe such information. Both TransXChange and SIRI are based on Transmodel, albeit with different purposes, the former designed for the efficient registration of bus services, the latter designed for exchange of real time information. Due to SIRI's emphasis on Vehicle Journeys, no explicit reference is made to Transmodel's Line, Route and JourneyPattern. Models instead place more emphasis on describing deviations from expected Vehicle Journey behaviour, based on features provided by Detection and Monitoring models within Transmodel.

Features provided this range of models include: ability to describe different types of scheduled routes; ability to describe variations in operation (short running, express services) or provide annotations; ability to describe hail and ride operations; ability to describe flexible services (where vehicles stop points and/or timings are not fixed); ability to link different service ends for service interchange and; ability to describe availability of dynamic information relating to particular services, in the form of both estimated and actual arrival times.

Within TransXChange the current omission of Fares information should be noted, although Fare Stages may be associated with stops. SIRI also lacks explicit structures relating to such information. This contrasts with information for specific transport modes which might also require booking or reservation, such as OTA (Air and Car Hire), FerryXML (Ferry) and RJIS (Rail), where fare information models exist (although not explicitly covered in this analysis), and where with typically higher costs, such information will be more valuable to the end user. Within the abstract Transmodel, fare information is included under the Fare Collection data model. Nevertheless, fares currently fall outside the scope of eMotion.

TPEG-PTI is designed for delivery of changes to public transport services, relating to a particular Station/Terminal/Interchanges, Route, Service or Area, in a language independent way. Messages can describe: service facilities (toilets, announcements, no mobile phones, first class); service conditions (cancelled, delayed, additional service); service ticket restrictions and event reasons of different types (misc, personnel, equipment, environmental). The SIRI standard makes reference to TPEG types for definition of Transport Modes and Service Facility classifications, but includes a richer set of features (eg. specific mobility features such as low access buses).

Both RailML and RJIS describe similar types of rail service information, essentially relating to scheduled timetables, but including information on train types and facilities. With RJIS, these details may change during the journey.

OTA includes information on scheduled flights (for planning and reservation), as well as information on dynamic status of flights. FerryXML is essentially concerned with retrieval of schedule information and booking functionality.
If real time data is provided by the operator, the data model is similar to that for static data. The service standard SIRI covers dynamic Timetable information

- Depending on the operators policy the connection timetable may be passenger information related, or is simply operational.
- stop monitoring services only relevant to service section are all covered by SIRI

2.9 POI and Other Directories

POIs typically contain venue descriptions, sometimes coupled with specific details, e.g. opening hours or event information. Connection to public transport may be explicit, or implicit through spatial location.

Basically, POIs are the contents of various directories, like Yellow Pages, White Pages, or specialised directories like lists of restaurants, hotels or tourist features.

Other important directories may describe topographic names, and their linkage to public transport.

2.9.1 ISO 14825: Geographic Data Files (GDF) - Feature Theme “Services” (POIs)

ISO14825:2004 / GDF describes a data model for representing real-world geographic phenomena on a conceptual level, semantic level, and physical level. An overview of GDF is available in the main document at subsection 3.5.7.

**Feature Catalogue**

The POI model is specified in different parts in ISO14825, namely the feature catalogue, attribute catalogue, and relationship catalogue. The following UML diagrams are taken from the XGDF project in ISO, depicting the most relevant feature classes, attributes types and relationship types.
Figure 2-95: GDF - ServiceFeature

Feature classes for POIs are not normative (in response to a variety of proprietary classification schemes applied in the market, often conditional to the targeted scope of applications and services), while the set of attributes types and relationship types contained in the GDF...
standard is normative.

GDF Service features are sub-types of the ServiceFeature Class, inheriting a number of attributes valid for the super-type, as well as local attributes. Wherever more than one Service feature owns a certain semantic attribute (such as brandName which for instance applies to Restaurant, Hotel, and Bank), the identical attribute semantics is applicable, being specified in the Attribute Catalogue.

Figure 2-96: GDF - Service Relationships

GDF Services (POIs) can be involved in association relationships with other features, such
as Road Elements, administrative areas, other GDF Services, to name a few (some of which are depicted in above diagram). Additionally, one or multiple Entry Points can be modelled providing for the point of access for travellers.

Encodings
See more extensive section on ISO14825 GDF road network data (section 2.1.1).

Portrayal Catalogue
See more extensive section on ISO14825 GDF road network data (section 2.1.1).

Service Reference
See more extensive section on ISO14825 GDF road network data (section 2.1.1).

2.9.2 OGC OpenLS - Core Services, Abstract Data Types (ADT)

The OGC OpenLS standard defines Location Based Services for a “GeoMobility Server” platform. It defines the interfaces for a couple of core service types, namely Directory Service, Gateway Service, Location Utility Service, Presentation Service, and Route Service. An overview of OGC OpenLS is available in the main document at subsection 3.4.7.

This subsection describes Abstract Data Types (ADT) definition of the standard, especially the definition of an object model for POIs.

Feature Catalogue
The following is “backward-engineered” from the XML Schema definitions of the ADT definition of the standard.

According to the standard a POI (Point-of-Interest) can be employed as one form of Location reference, but it constitutes also a feature of its own right, and therefore contains a mandatory identification id. Optionally there is also a description, a poiName and a phoneNumber.

The location of the POI can be described by a Point (from GML and thus from ISO19107) or by and Address, which is a data item described in 2.4.5.

The POI can be characterised by two different multiple items, called ReferenceSystem and POIInfoList.

ReferenceSystem identifies the directories the POI is described in. A few U.S. directories (NACE, NAICS, SIC) are predefined in the standard, however other NamedReferenceSystems may be added.

Information regarding the specific POI is given by a list of name-value-pairs grouped in the POIInfo object.
Figure 2-97: OGC OpenLS - POI

Encodings

Portrayal Catalogue
None associated.

Service Reference
This is part of the OpenGIS® OpenLS service definition, see 4.7.2.

2.9.3 Directory Services Standards (DSML) v1.0
The Directory Services Markup Language v1.0 (DSMLv1), provides a means for representing
directory structural information as an XML document. An overview of DSML is available in the main document at subsection 3.4.1.

As soon as Internet-based services start sharing information among each other or with client devices there is the need for an interchangeable and structured data exchange. The solution is to loosely connect as many data sources available using a standardized protocol rather than having one big monolithic database.

In order to lookup information in a vast variety and amount of different directories an extensible data structure and corresponding operations is required. Before you can access traffic information, point of interests, transit tables, etc. first you have to know where this type of information is located and how you can access it. Especially when dealing with region content, which traffic data certainly represents, directory services are required to ensure a virtual area-wide coverage.

**Feature Catalogue**

A DSML document describes either directory entries, a directory schema or both of them.

Each directory entry has a universally unique name called its distinguished name. A directory entry has a number of property-value pairs called directory attributes. Every directory entry is a member of a number of object classes. An entry's object classes constrain the directory attributes the entry may take. Such constraints are described in a directory schema which may be included in the same DSML document or may be in a separate document.

![Figure 2-98: DSML - DSML document](image)

**Directory Services Markup Language v1**

© eMOTION Consortium

Page 189 of 354
DSMLv1 is intended to be a simple XML schema definition that will enable directories to publish basic profile information in the form of an XML document so that it can be easily shared via Internet protocols. A DSML document describes either directory entries, a directory schema or both.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directory entries</td>
<td>entry element</td>
<td>The <em>entry</em> element contains elements representing the entry's directory attributes</td>
</tr>
<tr>
<td></td>
<td>entry object class</td>
<td>The object classes of an entry are represented by <em>oc-value</em> (URI Reference) child elements of an <em>objectclass</em> element. The content of each element indicates an object class to which the entry belongs.</td>
</tr>
<tr>
<td></td>
<td>directory attributes</td>
<td>Directory attribute is element that has a mandatory XML attribute name which indicates a name of the directory attribute.</td>
</tr>
<tr>
<td></td>
<td>multi-valued attributes</td>
<td>Multi-valued attribute is used in case of when an entry has multiple values for a particular attribute.</td>
</tr>
<tr>
<td></td>
<td>binary data</td>
<td>Binary data are encoded using an encoding scheme identified by an XML.</td>
</tr>
<tr>
<td>Directory schema</td>
<td>object classes</td>
<td>Object classes for each directory entry are indicated by elements of the type <em>objectclass</em> - defined with a class element in a directory-schema.</td>
</tr>
<tr>
<td></td>
<td>attribute type definitions</td>
<td>Directory attribute types - XML attributes are chosen in those cases where the information provides unique identification, is an enumeration or is a reference.</td>
</tr>
</tbody>
</table>

**Encodings**

DSMLv1 represents the state of a directory and allows XML-based enterprise applications to leverage profile and resource information from a directory in their native environment. It allows XML and directories to work together and provides a common ground for all XML-based applications to make better use of directories. Directory attributes containing binary data are encoded using an encoding scheme identified by an XML attribute encoding on the value element. At present, DSML supports only base64 as a value, but the encoding XML attribute is included in order to enable support for other encoding schemes in the future.

Encoding base64 allows for whitespace characters which are to be ignored by any decoding software. Base64 encoding does not introduce < or & characters and therefore no additional encoding is necessary to include base64 in XML character data.

**Portrayal Catalogue**

There is no portrayal associated with this standard.
Service Reference

From the general meaning point of view, specification defined within this standard is related with the directory services which can be through it reachable for the mobile end-users’ devices. For more details see chapter 4.7.

2.9.4 Point Of Interest eXchange Language Specification (POIX)

POIX (Point Of Interest eXchange Language Specification) is a location-related information descriptive language prepared with the aim of exchanging location-related information over the Internet, and is designed with XML 1.0 (Extensible Markup Language [W3C Recommendation]). An overview of POIX is available in the main document at subsection 3.4.8.

Feature Catalogue

![POIX - Features](image)

POIX represents a pretty simply defined standard. It focuses only on data attributes that are essential for a point of interest.
POI base object

The base element holds pretty obvious attributes: name of the POI and contact information like phone number or email address. A hypertext link allows to reference dynamic data somewhere on the Internet.

move (Moving POIs)

A POI can be represented as fixed location or a moving object. If the POI is modeled as a moving object the instance holds additional action attributes like velocity, travel direction, vehicle (car, ship, aircraft) type, locus or geographic coordinates (longitude, latitude, altitude, quality attributes). Currently, moving POIs are not commonly used at all.

access (Access Points)

Since a POI object may have multiple access points (entrance, exit, etc.) this object may have multiple instances to ipoint and tpoint objects described below.

ipoint (Introductory Point)

In order to describe the location of a POI in a “humanized” way user’s need better information that geographic coordinates. The ipoint data element add descriptive and human readable and understandable attributes to the POI object. Among a location class
ipoint provides also the (official) name of the location as well. Reverse geocoding methods don't necessarily provide this quality of information.

Today, map providers ship their maps with thousands of POI objects for almost every country coverage and region. Unfortunately, the geographic coding is done very poorly. A single geographic position for POI object is not very helpful for advanced navigation systems and location based services. Next to the virtual center position user's need the accurate location of entry points, exits, etc. Imagine being guided to a huge parking garage or football stadium. The attribute tpoint was exactly defined for this purpose.

The element point indicates the representative location of a POI objects. In the real world this means basically the virtual center point of a building or land usage are. The exact location of this point again is defined by regular geographic coordinates plus optional error indicators.

In the POIX standard we're still missing a very valuable piece of information all previos data formats missed as well: the access link. Since vehicles do move on road segments (links) we need information which link leads to the POI object. For this, an external reference id to the underlying digital road network is extremely helpful.

Since there are hundreds of different coordinate transformation methods this element defined which one was used when creating this object. Usually, WGS84 and a geo-decimal format is used – analog to GPS data. The field author defines the data source which is useful.
when doing quality checks and merging dataset.

The POIX standard is an approach in the right direction, but does not support today's requirements for modelling and geographically referencing objects on a map. However, since POIX is an XML based data structure extensions can be proposed and realised easily.

For the eMotion projects we're missing a clean and comprehensive for categorizing POI classes so that they may be exchanged, merged or compared with each other. At least for the domain of travel and traffic POI should be categorized using extensible structures based on XML schemas.

When dealing with names and classes we're totally missing rudimentary internationalization and localization aspects. For example, the monument “Kölner Dom” is called “Cologne Cathedral” in English, but the data model does not support multiple languages at all.

As far as the standardization of representing geographical and point of interest information, is concerned, relevant work by the Dublin Core and the IETF should also be considered.

**Encodings**

POIX is an XML format specified by a DTD.

POIX file type: For using POIX via the Internet and the other data processing systems, POIX file extension and media type must be defined.

POIX file extension: POIX file extension is .poi or .poix.

POIX media type: Media Type of POIX file is follow. application/x-po

**Portrayal Catalogue**

There is no portrayal associated with this standard.

**Service Reference**

Not associated to a service.
2.9.5 Tourism Markup Language (TourML)

Tourism Markup Language (TourML) is an XML encoding for the transport and storage of tourism information, including both the spatial and non-spatial properties of tourist objects, based on GML and developed 2003 by the Open Tourism Consortium. An overview of TourML is available in the main document at subsection 3.4.9.

Feature Catalogue

TourML provides an open and vendor-neutral framework for the definition of tourism application schemas and objects. It supports the description of tourism application schemas for the full range of tourism objects and activities. TourML defines abstract objects, events of interest and concrete tourist schema objects such as hotels, restaurants, movie theatres, banks, sights, etc.

TourML promises to enable and speedup the creation and maintenance of distributed tourism applications and datasets. By defining a standard TourML increases the ability of tourism organizations to share application schemas and the info they describe.
An Object Of Interest (OOI) or Event Of Interest (EOI) – this is how “point of interests” are called in the tourism sector – basically is a POI that focuses more on tourism attributes than on classical properties relevant in navigation systems. This said, we can consider OOI as POI with extended attributes like business hours, hotel category, restaurant type, region name, etc.

Obviously there are two emerging standards that somehow conflict with each other rather than sharing a common XML schema base. Since TourML is closely related to TourStyle, TourCMS, TourImplement, etc. this standard covers a large domain within a rapidly growing market. This standard is developed from the view of end user service providers who want to attract more visitors and customers to their regions and locations and exchange tourism offers through open interfaces.

From our analysis results we think that tourism databases will grow very rapidly while conventional POI databases that are embedded inside conventional digital maps will remain unattractive. TourML is driven by business models and mature markets that work properly.

Companies like Wcities or iPublish currently explore an emerging market for dynamic, worldwide and high quality tourism and travel information. These companies also plan to integrate their well investigated and updated content into navigation systems and online travel guides. Classical POI databases will waste away due to their poor attribute set and very bad up-to-dateness.

**Encodings**

XML encoding.

**Portrayal Catalogue**

There is no portrayal associated with this standard.

**Service Reference**

Not associated to a service.

### 2.9.6 Identification of Fixed Objects in Public Transport (IFOPT)

IFOPT defines a model and identification principles for the main fixed objects related to public access to Public Transport (e.g. stop points, stop areas, stations, connection links, entrances, etc.). An overview of the standard is available in the main document at subsection 3.2.4.

In this subsection is described the optional submodel *Point of Interest Model*.
The IFOPT Point of Interest model describes the structure of Points of Interest (POI), including points of access to these. The model includes a classification hierarchy which supports journey planning functions. A POI may be part of a parent POI, and one or more entrances may be specified for each POI. A POI may also have one or more POI Classifications assigned. The Classification Hierarchy supports searches for types of POI. Both POIs and POI Entrances are Places, therefore they may possess Zone Projection and/or Point Projection, and may be associated with an address.

Figure 2-101 : IFOPT - Point of Interest Model

GAZETTEER / TOPOGRAPHICAL MODEL

IFOPT also contains a specialised Directory of settlements, which is linked to the StopPoint, StopArea, and StopPlace objects of the IFOPT model.

This special Directory is presented in 2.12.2 Identification of Fixed Objects in Public Transport (IFOPT) because it is so special and linked to the journey planning needs.
Encodings
The fixed object model may be represented as an XML schema.

Portrayal Catalogue
There is no portrayal associated with this standard.

Service Reference
Not associated to a service.

2.9.7 National Public Transport Gazetteer (NPTG)
The National Public Transport Gazetteer (NPTG) provides a topographic database of towns and settlements in the UK. An overview of NPTG together with NaPTAN is available in the main document at subsection 3.2.6.

Feature Catalogue

NPTG MODEL AND DISCOVERY MODEL
The NPTG forms a specialised hierarchical Directory of towns and settlements in the UK. It is linked to a model of public transport StopPoints and StopAreas.

This special Directory is presented in 2.12.3 National Public Transport Gazetteer (NPTG) because it is so special and linked to the journey planning needs.

Encodings
Both NPTG and NaPTAN data are exchanged using XML documents complying with the NPTG and NaPTAN Schema, or alternatively using a NPTG and NaPTAN CSV formats respectively.

Portrayal Catalogue
N/A

Service Reference
JourneyWeb is built upon NaPTAN and NPTG standards. TransXChange also shares common elements.

2.9.8 Comparison and Evaluation
In the previous sections we have seen that the existing and emerging standards for modelling and storage of POI data mainly is driven by market demands and proven business plans.
Each standard still focuses on data elements and attributes which apply for a certain user group or market.

While the GDF standard offers probably the most advanced and sophisticated method for POIs in a geographic context, particularly concerning the referencing POIs to the road network, it lacks when dealing with content properties and attributes.

The OGC OpenLS standard provides us with a very generic description of POIs, which is suited for probably any directory, but which accordingly carries little semantics.

The POIX standard represents an also geographic-centric modelling method, which is a kind of superset of the GDF standard, even if both cannot be compared actually. Even that POIX has a richer attribute set, it still lacks information of POI objects when using in the tourism context.

TourML was designed as a B2B interchange protocol in order to share well formatted and extensible tourism data. While having a very rich and detailed attribute set, TourML lacks a proper geographic reference for navigation devices.

However, since all mentioned standards base on XML and inherently are extensible structures service providers may build and operate middleware components merging and transforming from one data source to an other. Using XML schema definitions and XML style sheet processors third parties can utilize open standards to build decentralized and distributed databases.

The IFOPT and NPTG directories described are specialised towards the needs of public transportation. They are also hierarchically structured, which distinguishes them from the flat structures simply describing POIs. The eMOTION design should therefore also consider gazetteers models (e.g. ISO 19112) to be able to map data source like IFOPT and NPTG entirely. This analysis has treated gazetteers under the heading of “services”, see section 4.8.

### 2.10 Road Weather

Road weather information is given for specific roads, routes or administrative areas and specific validity periods.

#### 2.10.1 TLS Standard - Environmental Data (FG3)

TLS is a national standard for Road Weather and Environmental Data Acquisition on Road Site Remote Stations (Technical Delivery Conditions for road side Controllers). An overview of TLS is available in the main document at subsection 3.3.2.

**Feature Catalogue**

TLS is divided in function groups for example for Traffic Data, Control of Traffic Signes etc.- One function group covers the field of acquiring of Road Weather or Environmental Data.
This feature covers a lot of traffic relevant data collected by road weather information systems.

The following table shows the defined function groups within TLS:

<table>
<thead>
<tr>
<th>Application Layer: Predefined Function Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FG No</strong></td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10 … 15</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>17 … 127</td>
</tr>
<tr>
<td>128 … 253</td>
</tr>
<tr>
<td>254</td>
</tr>
<tr>
<td>255</td>
</tr>
</tbody>
</table>

FG 3 is relevant in this chapter.

**Geographical Reference**

There is no absolute unambiguous geographical reference model used in TLS standard. The geographical reference is a node address, which is unique within a certain network. Function Group contain also a Data-Type which define some properties of the location, such as:

- Identifier for Country
- Type of Road
- Number of Road
- Route Kilometer of Location

The following Table shows the predefined Data-Types (DE-Typ) of the relevant function group:
Function Group: Road Weather and Environmental Data

<table>
<thead>
<tr>
<th>Description</th>
<th>Predefined DE-Types</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property</strong></td>
<td><strong>Type of DE-Value</strong></td>
</tr>
<tr>
<td>Air Temperature</td>
<td>16 Bit</td>
</tr>
<tr>
<td>Surface Temperature</td>
<td>16 Bit</td>
</tr>
<tr>
<td>Chemical Factor</td>
<td>8 Bit</td>
</tr>
<tr>
<td>Intensity of Precipitation</td>
<td>16 Bit</td>
</tr>
<tr>
<td>Air Pressure</td>
<td>16 Bit</td>
</tr>
<tr>
<td>Relative Humidity</td>
<td>8 Bit</td>
</tr>
<tr>
<td>Wind Direction</td>
<td>16 Bit</td>
</tr>
<tr>
<td>Winds Speed</td>
<td>16 Bit</td>
</tr>
<tr>
<td>Snow Height</td>
<td>8 Bit</td>
</tr>
<tr>
<td>Visibility</td>
<td>16 Bit</td>
</tr>
<tr>
<td>Brightness</td>
<td>16 Bit</td>
</tr>
<tr>
<td>Wind Speed Gust</td>
<td>16 Bit</td>
</tr>
<tr>
<td>Freezing Temperature</td>
<td>16 Bit</td>
</tr>
<tr>
<td>Dew Point Temperature</td>
<td>16 Bit</td>
</tr>
<tr>
<td>Subsurface Temperature TTx</td>
<td>16 Bit</td>
</tr>
<tr>
<td>Condition of Road Surface</td>
<td>8 Bit</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Type of Precipitation</td>
<td>8 Bit</td>
</tr>
</tbody>
</table>
Function Group: Road Weather and Environmental Data

<table>
<thead>
<tr>
<th>Description</th>
<th>Predefined DE-Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>Type of DE-Value</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Waterfilm Depth</td>
<td>16 Bit</td>
</tr>
</tbody>
</table>

**Encodings**

The physical layer normally is a Modem Line communication. A TLS Message contains typically the Data link Layer (OSI 2) and the network layer (OSI 3) and finally the application layer (OSI 7). The transport, session and presentation layer are not used.

The relevant application layer Data-Format is as follows:

```plaintext
class Application Message
    General_Message_Header
        + Node_Address: long

    Single_Message_Header
        + Application_Identifier: byte
        + function_group_No: byte
        + Jobnumber: byte

    DE_Block
        - DE-Channel: byte {readOnly}
        - DE-Type: byte {readOnly}
        - DE-Value: int
```

**Figure 2-102: TLS - Data Model of Application Layer**

Node Address is the location reference for the road side station. The “Application Identifier” specifies is used for specifying the purpose of the Message and the Direction of communication (request or replay). The job number is used to identify the message through all instances of the network. A replay uses the same job number as the corresponding request.

**Portrayal Catalogue**

none
Service Reference

none

2.10.2 Binary Universal Form for the Representation of meteorological data (BUFR) / Character form for the Representation and EXchange of data (CREX)

BUFR and CREX are standards for the international exchange of observational data. An overview of BUFR and CREX is available in the main document at subsection 3.3.1.

The figure below shows the typical architecture of the information chain using BUFR and CREX standard, actual in operation in the German RWIS (Road Weather Information System) and also the distribution of information to content operators – for example micKS MSR GmbH, who provides Content for Service Provider like T-Traffic.

Figure 2-103 typical architecture of information chain

Feature Catalogue

BUFR support the following Data Categories, used internationally in the whole field of meteorological information.
<table>
<thead>
<tr>
<th>ID</th>
<th>Figure Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Surface data – land</td>
</tr>
<tr>
<td>1</td>
<td>Surface data – sea</td>
</tr>
<tr>
<td>2</td>
<td>Vertical soundings (other than satellite)</td>
</tr>
<tr>
<td>3</td>
<td>Vertical soundings (satellite)</td>
</tr>
<tr>
<td>4</td>
<td>Single level upper-air data (other than satellite)</td>
</tr>
<tr>
<td>5</td>
<td>Single level upper-air data (satellite)</td>
</tr>
<tr>
<td>6</td>
<td>Radar data</td>
</tr>
<tr>
<td>7</td>
<td>Synoptic features</td>
</tr>
<tr>
<td>8</td>
<td>Physical/chemical constituents</td>
</tr>
<tr>
<td>9</td>
<td>Dispersal and transport</td>
</tr>
<tr>
<td>10</td>
<td>Radiological data</td>
</tr>
<tr>
<td>11</td>
<td>BUFR tables, complete replacement or update</td>
</tr>
<tr>
<td>12</td>
<td>Surface data (satellite)</td>
</tr>
<tr>
<td>13 – 19</td>
<td>Reserved</td>
</tr>
<tr>
<td>20</td>
<td>Status information</td>
</tr>
<tr>
<td>21</td>
<td>Radiances (satellite measured)</td>
</tr>
<tr>
<td>22 – 30</td>
<td>Reserved</td>
</tr>
<tr>
<td>31</td>
<td>Oceanographic data</td>
</tr>
<tr>
<td>32 – 100</td>
<td>Reserved</td>
</tr>
<tr>
<td>101</td>
<td>Image data</td>
</tr>
<tr>
<td>102 – 239</td>
<td>Reserved</td>
</tr>
<tr>
<td>240 – 254</td>
<td>For experimental use</td>
</tr>
<tr>
<td>255</td>
<td>Indicator for local use, with sub-category</td>
</tr>
</tbody>
</table>

Surface-Data (Land) and Radar-Data are currently used for exchange of source-Data for content-operator purposes.

Each category of Data is divided into different classes of Elements. The following table shows the classes used currently in exchange of Road Weather station data.

<table>
<thead>
<tr>
<th>Class ID</th>
<th>Class Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Identification</td>
</tr>
<tr>
<td>02</td>
<td>Instrumentation</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>03</td>
<td>Reserved</td>
</tr>
<tr>
<td>04</td>
<td>Location (time)</td>
</tr>
<tr>
<td>05</td>
<td>Location (latitude)</td>
</tr>
<tr>
<td>06</td>
<td>Location (longitude)</td>
</tr>
<tr>
<td>07</td>
<td>Location (vertical)</td>
</tr>
<tr>
<td>08</td>
<td>Significance qualifiers</td>
</tr>
<tr>
<td>09...10</td>
<td>Reserved</td>
</tr>
<tr>
<td>11</td>
<td>Average Wind Data</td>
</tr>
<tr>
<td>12</td>
<td>Temperature Data</td>
</tr>
<tr>
<td>13</td>
<td>Precipitation and Humidity Data</td>
</tr>
<tr>
<td>14 ... 19</td>
<td>Reserved</td>
</tr>
<tr>
<td>20</td>
<td>Phenomena and Visibility Data</td>
</tr>
</tbody>
</table>

**Geographical Referencing**

The Location Referencing is done by means of WGS84 coordinates and also optional the altitude is specified.

The following UML Diagram shows the Data Model structure.
Figure 2-104: BUFR / CREX - Data Model for Road Weather Information
Encodings

BUFR is usually transported in a file via FTP protocol. BUFR is encoded in a binary file, while CREX is encoded in ASCII.

The structures of the BUFR and CREX code forms are the following:

- **BUFR**
  - SECTION 0 Indicator Section
  - SECTION 1 Identification Section
  - SECTION 2 (Optional Section)
  - SECTION 3 Data Description Section
  - SECTION 4 Data Section
  - SECTION 5 End Section

- **CREX**
  - SECTION 0 Indicator Section
  - SECTION 1 Data Description Section
  - SECTION 2 Data Section
  - SECTION 3 (Optional Section)
  - SECTION 4 End Section

The Indicator Sections and the BUFR Identification Section are short sections, which identify the message. The list of descriptors, pointing towards elements in predefined and internationally agreed tables that are stored in the official WMO Manual on Codes (described previously), are contained in the Data Description Section. These descriptors describe the type of data contained in the Data Section and the order in which the data appear there. The Optional Section can be used to transmit any information or parameters for national purpose. The End Section contains the four alphanumeric characters “7777” to denote the end of the BUFR or CREX message.

Since the data in a CREX message are laid out one after the other, and since the data values of the parameters in a CREX message are transmitted in a set of characters, it is very simple to read a CREX message. While the order of the data contained in a BUFR message is likewise described by the BUFR Data Description Section, the data values of the parameters in a BUFR message are translated in a set of bits in BUFR. Consequently, a BUFR message is not human readable, or extremely difficult to decipher without the help of a computer program. CREX can be looked upon as the image in characters of BUFR bit fields. When there is a requirement for transmission of new parameters or new data types, new elements are simply added to the WMO BUFR and CREX tables, after approval by the CBS. Since table driven code forms can thus describe any new parameter by the simple addition of a new entry to the appropriate code table, table driven code forms possess the flexibility to transmit an infinite variety of information. Therefore, definition of new «code forms» is no longer necessary. Furthermore, procedures and regulations are fixed. A new edition number is assigned every time the BUFR or CREX code structure is changed. Although these edition changes require an update to BUFR or CREX encoding or decoding software, such changes
are infrequent (the BUFR Edition Number has changed only twice since 1988). Likewise, a new version number is assigned every time additions are made to BUFR or CREX code tables. Although version number changes are more frequent than edition number changes, they do not require modifications to the processing software.

**Portrayal Catalogue**

none

**Service Reference**

None

**2.10.3 ISO 14819: RDS-TMC - Road Weather related ALERT-C Codes**

RDS-TMC (Radio Data System – Traffic message Channel) is a Europe wide standard for transferring traffic relevant information into mobile devices. An overview of the standard is available in the main document at subsection 3.1.7.

This subsection contains only additional information regarding Road Weather related events. ALERT-C event definitions are also used by the road weather content provider micKS in relationship to the traffic information service providers. The corresponding Standard for mobile devices is EN ISO 14819-1:2003 (‘Coding protocol for RDS-TMC using ALERT-C’).

**Feature Catalogue**

The general structure of RDS-TMC messages is described in subsection 2.6.3.

**ROAD WEATHER**

The following list shows a short selection of the road weather related predefined ALERT-C events which are currently supported by the road weather content platform of micKS.

<table>
<thead>
<tr>
<th>Description</th>
<th>ALERT-C</th>
<th>Description</th>
<th>ALERT-C</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>16. PRECIPITATION AND VISIBILITY</strong></td>
<td></td>
<td><strong>17. WIND AND AIR QUALITY</strong></td>
<td></td>
</tr>
<tr>
<td>damaging hail (visibility reduced to Q)</td>
<td>1132</td>
<td>tornadoes</td>
<td>1201</td>
</tr>
<tr>
<td>damaging hail (with visibility reduced to Q)</td>
<td>1174</td>
<td>hurricane force winds (Q)</td>
<td>1202</td>
</tr>
<tr>
<td>hail (visibility reduced to Q)</td>
<td>1106</td>
<td>gales (Q)</td>
<td>1203</td>
</tr>
<tr>
<td>sleet (visibility reduced to Q)</td>
<td>1107</td>
<td>storm force winds (Q)</td>
<td>1204</td>
</tr>
<tr>
<td>thunderstorms (visibility reduced to Q)</td>
<td>1108</td>
<td>strong winds (Q)</td>
<td>1205</td>
</tr>
<tr>
<td>winter storm (visibility reduced to Q)</td>
<td>1128</td>
<td>gusty winds (Q)</td>
<td>1209</td>
</tr>
<tr>
<td>blizzard (visibility reduced to Q)</td>
<td>1130</td>
<td>crosswinds (Q)</td>
<td>1210</td>
</tr>
<tr>
<td>blizzard (with visibility reduced to Q) expected</td>
<td>1173</td>
<td>strong winds (Q) affecting high-sided vehicles</td>
<td>1211</td>
</tr>
<tr>
<td>heavy snowfall (Q)</td>
<td>1101</td>
<td>severe smog</td>
<td>1190</td>
</tr>
<tr>
<td>heavy snowfall (Q) expected</td>
<td>1170</td>
<td>severe exhaust pollution</td>
<td>1191</td>
</tr>
<tr>
<td>heavy snowfall. Visibility reduced (to Q)</td>
<td>1134</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Description

<table>
<thead>
<tr>
<th>Description</th>
<th>ALERT-C</th>
<th>Description</th>
<th>ALERT-C</th>
</tr>
</thead>
<tbody>
<tr>
<td>heavy snowfall (Q). Visibility reduced to &lt;30 m</td>
<td>1102</td>
<td>tornado warning ended</td>
<td>1217</td>
</tr>
<tr>
<td>heavy snowfall (Q). Visibility reduced to &lt;50 m</td>
<td>1103</td>
<td>strong winds easing</td>
<td>1213</td>
</tr>
<tr>
<td>snowfall (Q)</td>
<td>1104</td>
<td></td>
<td></td>
</tr>
<tr>
<td>snowfall. Visibility reduced (to Q)</td>
<td>1135</td>
<td>message cancelled</td>
<td>1214</td>
</tr>
<tr>
<td>snowfall (Q). Visibility reduced to &lt;100 m</td>
<td>1105</td>
<td></td>
<td></td>
</tr>
<tr>
<td>heavy rain (Q)</td>
<td>1109</td>
<td></td>
<td></td>
</tr>
<tr>
<td>heavy rain (Q) expected</td>
<td>1171</td>
<td></td>
<td></td>
</tr>
<tr>
<td>heavy rain. Visibility reduced (to Q)</td>
<td>1136</td>
<td></td>
<td></td>
</tr>
<tr>
<td>heavy rain (Q). Visibility reduced to &lt;30 m</td>
<td>1110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>heavy rain (Q). Visibility reduced to &lt;50 m</td>
<td>1111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Encodings

The Encoding is described in the TMC Compendium – ALERT-C Coding Handbook.

### Portrayal Catalogue

none

### Service Reference

None

#### 2.10.4 DATEX 2

The main aim of DATEX 2 is to establish standardised links between Road Operators, TICs and Service Providers to allow the easy and efficient communication of traffic and traveller Information. An overview of the standard is available in the main document at subsection 3.1.2.

Weather data are part of the *MeasuredDataPublication* of DATEX 2. In addition it is possible to give information about weather conditions and recommendation for the behaviour of the road user related to environmental and weather conditions.

### Feature Catalogue

One specialisation of *BasicDateValue* is *WeatherValue*. This feature covers a lot of traffic relevant data collected by road weather information systems or provided by weather services.
Visibility

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>visibilityValue</td>
<td>MetresAsNonNegativeInteger</td>
<td>The distance, measured or estimated, beyond which drivers may be unable to clearly see a vehicle or an obstacle.</td>
</tr>
</tbody>
</table>

PrecipitationDetail

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>depositionDepth</td>
<td>MetresAsFloat</td>
<td>The equivalent depth of the water layer resulting from precipitation or deposition on a non-porous horizontal surface. Non liquid precipitation are considered as melted in water.</td>
</tr>
<tr>
<td>precipitationIntensity</td>
<td>MillimetresPerHour</td>
<td>The height of the precipitation received per unit time.</td>
</tr>
<tr>
<td>precipitationType</td>
<td>PrecipitationTypeEnum: drizzle, hail, rain, sleet, snow</td>
<td>The type of precipitation which is affecting the driving conditions.</td>
</tr>
</tbody>
</table>
### RoadSurfaceConditionMeasurements

<table>
<thead>
<tr>
<th>Description</th>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurements of the road surface condition which relate specifically to the weather.</td>
<td>delIcingApplication-Rate</td>
<td>IntensityKilogrammesPerSquare-Metre</td>
<td>Indicates the rate at which de-icing agents have been applied to the specified road.</td>
</tr>
<tr>
<td></td>
<td>delIcingConcentration</td>
<td>ConcentrationGramsPerCubicCentimetre</td>
<td>Indicates the concentration of de-icing agent present in surface water on the specified road.</td>
</tr>
<tr>
<td></td>
<td>depthOfSnow</td>
<td>MetresAsFloat</td>
<td>The measured depth of snow recorded on the road surface.</td>
</tr>
<tr>
<td></td>
<td>protectionTemperature</td>
<td>TemperatureCelsius</td>
<td>The road surface temperature down to which the surface is protected from freezing.</td>
</tr>
<tr>
<td></td>
<td>roadSurfaceTemperature</td>
<td>TemperatureCelsius</td>
<td>The temperature measured on the road surface.</td>
</tr>
<tr>
<td></td>
<td>waterFilmThickness</td>
<td>MetresAsFloat</td>
<td>Indicates the depth of standing water to be found on the road surface.</td>
</tr>
</tbody>
</table>

---

### Temperature

<table>
<thead>
<tr>
<th>Description</th>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Details of atmospheric temperature.</td>
<td>airTemperature</td>
<td>TemperatureCelsius</td>
<td>The air temperature measured in the shade between 1.5 and 2 metres above ground level.</td>
</tr>
<tr>
<td></td>
<td>dewPointTemperature</td>
<td>TemperatureCelsius</td>
<td>The temperature to which the air would have to cool (at constant pressure and water vapour content) in order to reach saturation.</td>
</tr>
<tr>
<td></td>
<td>maximumTemperature</td>
<td>TemperatureCelsius</td>
<td>The expected maximum temperature during the forecast period.</td>
</tr>
<tr>
<td></td>
<td>minimumTemperature</td>
<td>TemperatureCelsius</td>
<td>The expected minimum temperature during the forecast period.</td>
</tr>
</tbody>
</table>

---

### Wind

<table>
<thead>
<tr>
<th>Description</th>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wind conditions on the road.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© eMOTION Consortium
Page 211 of 354
<table>
<thead>
<tr>
<th>maximumWindSpeed</th>
<th>KilometresPerHour</th>
<th>The maximum wind speed in a measurement period of 10 minutes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>windDirectionBearing</td>
<td>NonNegativeInteger</td>
<td>The average direction from which the wind blows, in terms of a bearing measured in degrees (0 - 359).</td>
</tr>
<tr>
<td>windDirectionCompass</td>
<td>DirectionCompassEnum: east, eastNorthEast, east-SouthEast, north, Northeast, northNorthEast, northNorthWest, northwest, south, southeast, southSouthEast, southSouthWest, southwest, west, westNorthWest, west-SouthWest</td>
<td>The average direction from which the wind blows, in terms of points of the compass.</td>
</tr>
<tr>
<td>windMeasurementHeight</td>
<td>MetresAsNonNegativeInteger</td>
<td>The height in metres above the road surface at which the wind is measured.</td>
</tr>
<tr>
<td>windSpeed</td>
<td>KilometresPerHour</td>
<td>The wind speed averaged over at least 10 minutes, measured at a default height of 10 metres (meteo standard) above the road surface, unless measurement height is specified.</td>
</tr>
</tbody>
</table>

There are some further items in DATEX 2 with respect to weather conditions:

- One specification of **Advice** is **Warning** with the possibility to give following weather-related warnings:
  - Water lying on the road surface is producing aquaplaning conditions.
  - Increased risk of skidding.
  - Increased risk of pedestrians slipping on the pavements.
  - Water is resting on the roadway which provides an increased hazard to vehicles.

- One specification of **Advice** is **WinterEquipment** with the possibility to give following weather-related advices:
  - Snow chains or tyres are recommended.
  - Stud tyres may be used.
  - Carrying of winter equipment (snow chains and/or snow tyres) is recommended.

- One specialisation of **TrafficElement** is **Conditions** with the possibility to describe weather-related road conditions (surface conditions) and poor environmental conditions which are also weather related.
Encodings


Portrayal Catalogue

There is no portrayal associated with this standard.

Service Reference

There is a reference to DATEX 2 data service described in chapter 4.2.3.

2.10.5 Comparison and Evaluation

The above described standard for Road Weather Information TLS and BUFR/CREX are considered as rough-data or “sensor-data” communication standards which are used as inputs for content operators. The encoding covers only measured data – no elaborated data or added values information are included. DATEX2 is based on XML standards and can be used for exchange of elaborated data and information as well as for measured data. Therefore this standard is much more relevant for intercommunication between eMOTION partners. Regarding the traffic related weather information, DATEX2 provides already predefined data types. But this specifications should be improved and enhanced in terms of eMOTION Services.

ALERT-C in the framework of RDS-TMC standard covers solely elaborated data, which are already service related data, but also used to provide value added information between content operators and service providers. The ALERT-C can directly be decoded by mobile devices of the endusers.

The geographical reference in TLS is based on the position of the road and is not compatible with other in eMOTION relevant geographical reference standards.

While in BUFR / CREX the geographical referencing is based on WGS84 coordinates which can be in principle be matched to other relevant geographical models. ALERT-C in the framework of RDS-TMC uses predefined TMC Locator for location referencing. For TMC locator there are mapping tools available into Link-ID’s of other common used road map models e.g. TeleAtlas or Navtech. In DATEX2 there are also ALERT-C location references defined.

Regarding the content and data types, the BUFR covers a subset of data types which are directly represented also in the environmental Data (FG3) function group of the TLS standard. For example “Surface Temperature”, “FreezePoint Temperature”, “Waterfilm Depth” and “Intensity of Precipitation” are data types represented in TLS and BUFR. While the ALERT-C Message “slippery road” (Code: 1003) is an elaborated information which requires a quite complex fusion of the above mentioned parameter resp. measured data. This fusion also has to take in account the geographical position of the source data and the timeframe of the acquisition of them – and so on. While DATEX2 is intended to cover both, the measured data as well as value added information. Seen from this point of view, DATEX2 seem to be
the best choice for the services between the eMOTION partners. At least regarding the traffic weather data the data model within DATEX2 had to be extended and updated to cover all the necessary information and data types needed for sufficient providing of appropriate content.

2.11 Data for Routing

Routing needs static road network data as explained above. The road network need to be a topologic network containing the necessary information for turn-by-turn-navigation and inhibited manoeuvres. The static network standards are explained above and are not repeated here.

The result of the routing process is a ‘route’, which is a data entity of its own importance and is transmitted from services to clients. It has therefore been depicted in this section.

Other data, which would fall under the category, for example the network ‘cost model’, has not been presented here, because it is usually hidden in a routing service.

Routing may also need access to additional data

- dynamic traffic data, both traffic flow and messages (to compute traffic condition triggered detours)
- parking (as possible route destination)
- POIs (as possible route destination)

These additional access requirements shall be fulfilled by appropriate data services belonging to other content domains.

2.11.1 ISO 19133 - Chapter 7 Navigation and Chapter 9 Network

ISO 19133:2005 describes the data types, and operations associated with those types, for the implementation of tracking and navigation services. An overview of the standard is available in the main document at subsection 3.4.5.


Feature Catalogue

The NT_Route type is a construct, which describes a route though the network in terms of and in reference to a network. The construct is therefore strongly coupled to the types of the ISO19133 Network, which was explained in section 2.1.2.

NT_Route basically defines a path though a NT_Network (see 2.1.2), adds a geometry for that path (a GM_CompositeCurve) and gives summary information regarding the route, which among other things treats the special case at the start of the route and at the end, where the route is usually bounded by positions, which cannot directly described by network elements.

The main network reference of an NT_Route is given by a sequence of references to
NT_Maneuver objects (see 2.1.2), which encompass all turns and links in the network making up the route.

The \textit{NT\_RouteSummary} object makes statements about total time, distance and extent of the route and additionally replicates the \textit{NT\_WayPointList}, which was originally given in the routing request.

Start and end of the route are defined by \textit{NT\_NetworkPosition} objects, which either are represented by NT\_Junctions (if start or end is in a junction) or by positions on a link, which is are given by referencing that NT\_Link and and additional linear referencing quantities \textit{linkMeasure} and \textit{marker}.

We now come to the more extensive representation of a route, which is part of the response to a routing request, which is named \textit{NS\_InstructionList}.

A \textit{NS\_InstructionList} stands for an ordered list of instructions. A route can be navigated by executing the sequence of instructions. A \textit{NS\_InstructionList} also contains a \textit{NT\_Route} object as described above.
Each *NS_Instruction* describes a single route instruction or advisory. It contains information about the *cost* of the instruction, which is the cost for the maneuvers and the following links, and optionally clear-text navigation *action* descriptions. An instruction can cover multiple maneuvers (NT_Maneuver objects) as in “turn right and then immediately left”. Instructions are required for each maneuver that does not execute the main-road rule.

*NT_Advisory* object owns an advisory *category*, which states the subject of the advisory, like highway-merge, road-merge, landmark, toll-booth, etc. There may be also a clear text *description* attached. There is a set of *NT_AdvisoryElements* bound to a *NT_Advisory*, which can come in up to four different flavours.

*NT_AdvisoryDistance* objects describe the distance at which the item mentioned in the descriptive text of the *NT_Advisory* occurs.

*NT_AdvisoryDirection* stands for the direction of the item mentioned in the descriptive text using terminology like straight, keepLeft, slightLeft, left, sharpLeft, uTurn, etc.

*NT_AdvisorySpatialRelation* describes the spatial relationship between the route and the item mentioned in the descriptive text. Possible codes are leftOfRoute, rightOfRoute, overRoute.
NT_ExitAssociation provides a link to an exit or a pair of exits at a junction. The attribute exit1 is the count from the entry point of the route into the junction to the primary exit. exit2 allows to specify another count for the same junction.

**Encodings**
There is no encoding associated.

**Portrayal Catalogue**
There are no portrayal rules associated.

**Service Reference**
The content described is used by the definition of the ISO 19133 Navigation Service, see 4.4.1.

### 2.11.2 OGC OpenLS - Core Services, Route Service and Abstract Data Types (ADT)

The OGC OpenLS standard defines Location Based Services for a “GeoMobility Server” platform. It defines the interfaces for a couple of core service types, namely Directory Service, Gateway Service, Location Utility Service, Presentation Service, and Route Service. An overview of OGC OpenLS is available in the main document at subsection 3.4.7.

This description refers to Part 5 of this standard, which is named “Route Service” and to the definition of Abstract Data Types (ADT). The OGC Route content definition is very close to the ISO 19133 route structures explained above.

**Feature Catalogue**
The following is “backward-engineered” from the XML Schema definitions of the ADT definition and the Route Service definition of the standard.

The DetermineRouteResponse object is the container, for different representations of a route which actually belongs to the OpenLS Service interface. It contains a RouteHandle attribute, which refers to a representation of the route on service, which is a stateful service.
The first representation of the route is a mandatory `RouteSummary`, consisting of `totalTime`, `totalDistance` and a `boundingBox`.

The `RouteGeometry` is simply a geometry object, a `LineString`.

The `RouteInstructionList` is a container for one or more `RouteInstructions`, each being a turn-by-turn route instruction or advisory formatted for presentation. Attributes in the `RouteInstructionList` select a `format` (a MIME type) for the specification of the instructions and a `language`.

Each `RouteInstruction` carries a `duration` and a `distance` for the described turns, `description` and `instruction` being free-text of the selected `format` and an optional `boundingBox`. The `RouteInstruction` may also carry a geometry.

Another way of specifying a route is by a map, a `RouteMap`. However, we will not make this point out exactly.

**Encodings**


**Portrayal Catalogue**

None associated.
Service Reference
This is part of the OpenGIS® OpenLS service definition, see 4.4.2 and 4.7.2.

2.11.3 Comparison and Evaluation
Both standards, ISO 19133 and OGC OpenLS Route Service have been analysed regarding the structures they offer for representing the results of routing or navigation requests, i.e. for the representation of routes.

As it turns out, both models are quite similar, though the OGC OpenLS model does not list ISO 19133 as a contributing input reference.

Both models describe a route by
- a geometry
- a route summary object, stating overall measures for route line time and distance
- a detailed instruction list giving instructions and advisories on a turn-by-turn basis.

Generally, the OpenLS modelling is simpler than the ISO 19133 model and better suited for transporting the information in a web service environment, whereas the ISO model is entangled with the underlying network model and would need an interface, which is more fine-grained than the OpenLS interface.

For eMOTION this will mean that it will probably make sense to start from the OpenLS definition and add necessary data by building an appropriate profile of the definition.

2.12 Data for Public Transport Journey Planning
Public transport journey planning needs static public transport network data as described above, time tables and additionally information on fares, connection footpaths and appropriate maps.

Transport Journey Planning may also need access to
- dynamic public transport data (to adapt to disturbances)
- POIs (as possible route destination)

These additional access requirements shall be fulfilled by appropriate data services belonging to other content domains.

The data requirements for transport journey planning seem to coincide with the data requirements from the Public Transport Service Data section above.

2.12.1 ENV 12896: Transmodel
Transmodel defines an extensive data model for relevant Public Transport related information. An overview of Transmodel is available in the main document at subsection 3.2.14.
Feature Catalogue

A trip planning can be calculated in response to TRIP OPTIMIZATION QUERies. This task uses information on places of origin and destination and may offer several solutions.

Several criteria and preferences can be taken into account: minimal trip duration, minimal number of interchanges, low cost etc. A “trip pattern” is proposed as the result of such a search and optimization task with rides on PT vehicles and if necessary, access walks or connection walks for interchanges. Once a trip pattern proposal is known, it is possible to make further calculations and optimizations on duration, fares, constraints and so on.

PLACES and Access Walks

It may be useful or necessary to have an origin and destination definition not strictly associated to STOP POINTs. A simple “generic place” concept may be more appropriate for many real cases. The data model defines an abstract PLACE entity.

Generic places can be 0 dimension: sites, stop points, 1 dimension: road sections (streets), 2 dimensions: sections of town (zones).

Examples of “real-world” start and stop places for a trip may be a street, an address etc.

The entity PLACE provides a description of geographical elements that defines the origin and the destination of a passenger trip.

A TRIP search can use the PLACE entity to represent start and stop parameters for the traveller’s trip. ACCESS LINK defines the walking paths to access the public transport network from a PLACE origin, or to leave the network in a PLACE destination. A default duration is an attribute of ACCESS LINK expressing the duration to cover this link. A ‘distance’ may also be stored.

A PLACE may be defined as a

- POINT, (STOP POINT or other type)
- STOP AREA,
- ZONE
- SITE (whose name is well known, in the neighbourhood of the network)

SITEs, can be classified by TYPE OF SITE. Such as SERVICE SITE (relevant for the operator) and PUBLIC SITE (relevant for passengers). A SITE may be in the vicinity of a STOP AREA.

Trip Pattern Search

The entity TRIP PATTERN describes the spatial movement of a passenger from a PLACE origin to a PLACE destination. A RIDE is the part of a passenger trip taken on a single PT vehicle, from one STOP POINT to another. A RIDE is associated to only one JOURNEY PATTERN.
A TRIP OPTIMIZATION QUERY gives the TRIP PATTERN as the result of the search. The algorithm that gives response to the TRIP OPTIMIZATION QUERY will find the optimal proposal using several optimization criteria that involves the above entities. An OPTIMIZATION MODE may specify particular optimization conditions.

The TRIP PATTERN may be composed of one or more RIDEs, walks on ACCESS LINKs and (in presence of several RIDEs) the necessary walks on CONNECTION LINKs, that is an entity of relevance in the inter-modal transport network data context and hence already introduced in sect. 2.3.

One objective is to minimize (or have null) length/duration of ACCESS LINK and CONNECTION LINKs.

A RIDE is generally associated to a SERVICE JOURNEY PATTERN. A TRIP PATTERN may be formed by one PT TRIP and walks on the corresponding ACCESS LINKs. RIDE IN PT TRIP is an expression of the order of RIDEs in the PT TRIP. PT TRIP stands for the part of TRIP PATTERN operated on the public transport network.
Figure 2-109: Transmodel - Passenger Trips

Encodings

Transmodel is a concept data model and does not mandate any particular implementation at
the logical or physical level. Other Standards are based on transmodel and uses specific encodings. For example TransXchange is XML based.

**Portrayal Catalogue**

None

**Service Reference**

Data content defined and described in this standard is related with public transport information services.

### 2.12.2 Identification of Fixed Objects in Public Transport (IFOPT)

IFOPT defines a model and identification principles for the main fixed objects related to public access to Public Transport (e.g. stop points, stop areas, stations, connection links, entrances, etc.). An overview of the standard is available in the main document at subsection 3.2.4.

In this subsection describes the optional submodel Topographical Model.

**Feature Catalogue**

The model contains a topographical representation of settlements.

It is useful to associate Stop Places and Points of Interest with topographic features to support functions such as journey planning.
The model supports hierarchical arrangement of geographical localities of types (e.g. city, town, village, suburb, etc.) as with a regular Gazetteer, but allows explicit links to public transport entities (namely IFOPT Stop Places and TransModel Stop Points and Stop Areas).

**Encodings**

The fixed object model may be represented as an XML schema.

**Portrayal Catalogue**

None

**Service Reference**

None associated

**2.12.3 National Public Transport Gazetteer (NPTG)**

The National Public Transport Gazetteer (NPTG) provides a topographic database of towns and settlements in the UK. An overview of NPTG together with NaPTAN is available in the
main document at subsection 3.2.6.

Feature Catalogue

NPTG MODEL

The NPTGLocality represents the fundamental entity of the NPTG, and represents a city, suburb, town, district, village or other settlement. Relevant elements of locality are detailed in the table below.

<table>
<thead>
<tr>
<th>NPTG Locality</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Specialises</strong></td>
</tr>
<tr>
<td><strong>Description</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locality Code</td>
<td>A unique ID</td>
<td>Unique Identifier of the Locality</td>
</tr>
<tr>
<td>Location</td>
<td>OS Grid Reference</td>
<td>Central point of the locality, expressed as Ordnance Survey Grid Reference</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>The NPTG locality name</td>
</tr>
</tbody>
</table>

*Figure 2-111: NPTG - Overview Class Diagram (excluding administrative elements)*
**Short Name** | **String** | **A short name for the locality**
--- | --- | ---
**Alternative Descriptor** | **Alternative Descriptor List** | **Alternative name for locality. May also be used for other languages**
**Qualifier Name** | **String** | **Additional name which might also be specified to disambiguate between localities with the same name**
**Qualifier Reference** | **Reference to Unique ID** | **Reference to qualifier locality or district used for qualifier name**
**Parent Locality** | **NPTG Locality Reference** | **Reference to Unique ID of parent**
**Classification** | **Locality Type Enumeration** | **Used to classify the locality, for example, city, town, suburb**

### NPTG DISCOVERY MODEL

AdjacentRegionExchangePoints define NaPTAN StopPoints on the boundaries of neighbouring journey planners areas which JourneyWeb distributed journey planners need to recognise.

A TrunkLocality is a geographical grouping of Stop Points and Stop Areas which may be used by external journey planners to find trunk access points for a locality.

---

© eMOTION Consortium
Encodings
Both NPTG and NaPTAN data are exchanged using XML documents complying with the NPTG and NaPTAN Schema, or alternatively using a NPTG and NaPTAN CSV formats respectively.

Portrayal Catalogue
N/A

Service Reference
JourneyWeb is built upon NaPTAN and NPTG standards. TransXChange also shares common elements.

2.12.4 Comparison and Evaluation
Public Transport Journey Planning necessarily involves data from public transport network and public transport service data. Road network and routing data will also be valuable for the user to navigate by foot. This section has included standards for data necessary to glue this information together and provide a journey plan.

Standards covered in the section fall into two categories:

1. Data the allows location of appropriate access points to the network (at both origin and destination)
2. Data that allows distributed journey planning

For the first of these, origin and destination may be:

- station/airport/PT: in NaPTAN or IFOPT Stop Place model used to locate stops in proximity.
- address: coordinate lookup and then locate closest stop points.
- town/city/suburb: stop points lookup to locate main access points to locations such as for NaPTAN combined with NPTG, or IFOPT linked Stop Place and Topographical/Gazetteer model
- POI: coordinate lookup to locate access points in proximity; or explicitly linked as with Access Path Links between POI and Stop Place Model in IFOPT
- Generic place: TransModel's Access Links allow a Stop Point to be linked from/to any Place (including Zones, Roads, Addresses and POIs).

Where an explicit Access Link is stored or calculated, a minimum time to traverse the link is specified, and possibly a distance.

The final data standard relates to distributed journey planning, where the NPTG Discovery
Schema provides information on suitable access points for:

- trunk journeys - identification of trunk localities where journey planner regions may not be neighbours.
- adjacent region journeys - identification of adjacent region points where journey planners cover neighbouring areas.

2.13 Data for Inter-Modal Journey Planning

Inter-modal journey planning needs static network data, both road and public transport as described above. It may also require additional data, which is specific for the special needs of multi- and inter-modality.

2.13.1 ISO 19134 – Multimodal Routing and Navigation

ISO 19134:2007 describes a model for multi-modal (in the sense of inter-modal) routing and navigation. An overview of the standard is available in the main document at subsection 3.4.6.

As in 2.11.1 we will concentrate on the Route construct, which constitutes the primary output of a routing operation. The object model pointed out at this place extends the one presented in 2.11.1 and creates a multi-modal Route, which is the equivalent to an inter-modal journey plan. The definitions rest on the extended network definitions given in 2.3.1.

Feature Catalogue

We will first describe the extended (multi-modal) Route construct.
The extension is called **MM_Route** and is a specialisation of NT_Route. It constitutes a specialised (linear) MM_MultimodalNetwork in the same sense that NT_Route is a NT_Network. There are only few and minor extensions in MM_Route compared to NT_Route.

A new item introduced at the level of multi-modality is the **MM_RouteSegment** object. It is used to delineate any route segments between two transfer nodes (MM_TransferNode, see 2.3.1) or between a transfer node and either the start or end of the parent MM_Route. The type **MM_RouteSegment** is a subtype of both MM_Route and MM_SingleModeNetwork. It may have two subtypes, one (MM_FixedScheduleRouteSegment) for fixed route segments such as transit route segments and another (MM_FlexibleScheduleRouteSegment) for non-fixed route segments such as taxi route segments.

**MM_TripScheme** is a specialisation of NT_Maneuver and has essentially the same role in MM_Route as NT_Maneuver has in NT_Route. It describes a trip for a traveller on the multimodal network. A trip is a sequence of actions given by sequence of either turns or transfers.

The **MM_RouteSummary** type adds a numberOfTransfers property to NT_RouteSummary, while the rest of the summary structure remains identical to the one of NT_Route.

Regarding the instruction list (see below), there is only the specialisation of NS_Instruction to **MM_Instruction** to allow referencing the new MM_TripScheme object.
Figure 2-114: ISO 19134 - Multi-modal Trip Scheme

**Encodings**
No encodings associated.

**Portrayal Catalogue**
No portrayal associated.

**Service Reference**
The definition is part of the definition of an ISO 19134 “multi-modal routing service”.

**2.13.2 Comparison and Evaluation**
Evaluation of the multi-modal (inter-modal) feature model of the Route construct against the single-mode one reveals only small changes.

Concerning public transport the considerations of the last section 2.12 have to be included.

In constructing a general Route construct for eMOTION, possibly extending an existing service interface definition, these additional items can be easily incorporated.

**2.14 Data for Freight Traffic**
Freight Traffic needs special information on weight and height limitations of vehicles, messages regarding disturbances in customs clearing, information on ferries, weather conditions critical for freight vehicles.

General traffic flow and messages as treated above are also necessary.
There are no specific standards in the area of freight traffic relevant for the eMOTION project because of the following facts:

- the eMOTION project is only dealing with information services that provide special information for freight traffic with the focus on road freight traffic,
- the eMOTION project is not dealing with fleet management and goods distribution systems, and
- the eMOTION project is not dealing with logistics application for road and rail freight traffic operators, or the communication between different operators in freight traffic like the carrier and the railway operator or the carrier (organisation of intermodal freight traffic) and logistic companies at harbours (shipment).

Because of this, the content standards especially in the area of traffic messages and weather information are also relevant for freight traffic information. This means that the necessary contents will be realised as extensions to other types of information.

2.15 General Metadata

This analysis has not discovered any specific metadata standard from the ITS domain, which would be suitable to appropriately describe content offerings.

Such metadata is, however, required to set up centralized service resources which would allow ITS data and services to be detected and evaluated for their usability.

The eMOTION system specification will therefore have to define this metadata from scratch or have to fall back on “general purpose” metadata standards. Clearly, the metadata standard from the ISO19100 series of international standard lends itself to be employed, because the content and services model of eMOTION will be based on these standards anyway.

2.15.1 ISO 19115 – Metadata

ISO 19115 defines a schema required for describing geographic information and services. It provides information about the identification, the extent, the quality, the spatial and temporal schema, spatial reference, and distribution of digital geographic data.

An overview of the standard is available in the main document at subsection 4.2.1.

Feature Catalogue⁹

The metadata for geographic data, defined in ISO 19115, is presented in packages, where each package stands for one specific aspect of metadata. Each package usually consists of

⁹ Note that the word “feature”, the use of which in this place is due to a uniform structuring approach in this document, is not appropriate. “meta-feature” would be more suitable.
several classes, which describe the metadata concerned with that specific aspect.

In this analysis only the package “Metadata entity set information” will be shown as an example for the various packages modelled in ISO 19115. Please refer to the standard to see the complete model.

**Metadata entity set information**

The package shows containment relationships with the other metadata classes, which, in aggregate, define metadata for geographic data. The idea behind the central class MD_Metadata is to describe (with the help of the associated specialised classes) a geographic resource as a whole and in all its aspects.

![Figure 2-115: ISO 19115 - Package “metadata entity set information”](image)

A list and a short description of all other packages follow.

**Identification Information**

The classes in this package (main class MD_Identifier) are required to identify a resource. It also defines separate specialisation sub-classes for identifying data and services.

**Constraint information**

This package (main class MD_Constraints) contains the metadata required for managing rights on information including restrictions on access and use.

**Data quality information**

© eMOTION Consortium
This one (main class DQ_DataQuality) defines the metadata required to give a general assessment of the quality of a resource. This contains lineage information, which is metadata required to describe the sources and production processes used in producing a dataset, and information about completeness, consistency, and accuracy.

**Maintenance Information**

This package (main class MD_MaintenanceInformation) contains the metadata necessary to describe maintenance and update practices for information.

**Spatial representation information**

The package (abstract main class MD_SpatialRepresentation) defines metadata to describe the mechanism used to represent spatial information.

**Reference system information**

This package (main class MD_ReferenceSystem) defines metadata required to describe the spatial and temporal reference system used.

**Content information**

The package (main class MD_ContentInformation) contains metadata about the content of a coverage and the feature catalogue(s) used to define features.

**Portrayal catalogue information**

This package (main class MDMPortrayalCatalogueReference) defines metadata about the portrayal catalogue(s) used to display data.

**Distribution information**

This package (main class MD_Distribution) defines metadata required for accessing a resource (metadata such as distributor, ordering process, medium, formats).

**Metadata extension information**

This package (main class MD_MetadataExtensionInformation) defines information about extensions to ISO 19115 definitions.

**Application schema information**

This package (main class MD_ApplicationSchemaInformation) defines the application schema used.

**Encodings**

An encoding should follow ISO 19118 and ISO 19139.

**Portrayal Catalogue**

No portrayal associated.
Service Reference
There is an OGC Catalogue Service definition, which are tailored to use ISO 19115 metadata. See 4.10.4.

2.15.2 Comparison and Evaluation
ISO 19115 is not the only existing metadata standard for geospatial data, let alone for simple, non-spatial data. Most other standards are less complex than ISO 19115.

It is, however, the only one in the ISO 19100 series of international standards, and is therefore probably a good initial choice for eMOTION.

Since no real requirements except the necessity for find and evaluation metadata have been found, eMOTION should make use of the standard in its current form and not provide a special profile. This can be done as soon as some experience regarding its employment is on hand.
3. Analysis of Encoding Standards

Encodings describe specialized vocabularies for the transfer of specific kinds of data as messages between application clients and services, and between services.

Encoding standards are therefore auxiliary standards for the definition of service interfaces. They are usually part of the request / response service definitions, often, however, they are important and general (and most often complex) enough to be worth a definition of their own. Most encodings can be used by more than one service definition.

Most encodings are of course XML-based, however there are also well-known binary and text encodings in use, for example image formats like GIF or PNG.

This chapter is not at all complete regarding encodings relevant for eMOTION, because in this analysis they are of minor importance and they are therefore in most cases treated in the context of the interface definition of the service in which they are employed.

The abilities and structure of encodings are not in the focus of eMOTION modelling, because they will be used exactly in the form they are defined and there is no need for an elaborate survey of encodings for harmonisation or to decide which to use. Most encodings (like HTML) are actually so well-known, that they need not be explained.

An exception to this rule is Geography Markup Language (GML), because it is the chosen encoding for the exchange of eMOTION content over eMOTION services.

3.1 Content Encodings

Actually, there is more than one encoding standard capable of capturing content and transporting this content between services, and from services to clients. Most modelling standards and models also define an exchange standard, which allows the serialisation and transport of content modelled according to the standards.

For example the GDF definition possesses an exchange standard based on a fixed character format, which is still in use. (Clearly, this was defined in the pre-XML era.)

The encoding standard for data modelled in the framework of ISO/TC211 (the ISO 191xx standard) is ISO 19136, also called Geography Markup Language, GML. We will focus on this single content encoding standard, because it will play a central role in eMOTION.

3.1.1 Geography Markup Language (GML)

Geography Markup Language (GML) is an XML encoding for the transport and storage of geographic information, including both the spatial and non-spatial properties of geographic features. An overview of the standard is available in the main document at subsection 4.1.1.

To get an impression about the GML encoding, please refer to the following example from another European FP6 project, called RISE, with the focus on water, elevation and land-use data themes. RISE is short for Reference Information Specifications for Europe.
See [http://www.eu-rise.org/](http://www.eu-rise.org/) for more information on RISE.

RISE is based on the ISO 191xx series of standards and uses the model-driven architecture approach defined there.

The following diagram shows the conceptual UML model of RISE. From this model a GML application schema is generated by means of an automated process. The GML schema describes the format of RISE data exchange.

![GML - RISE UML model](image)

**Figure 3-1: GML - RISE UML model**

The following GML instance file shows for a selected feature type (the one marked by an arrow in the UML diagram) a corresponding feature instance. The output was obtained from a RISE WFS.

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
  <gml:boundedBy>
    <gml:Envelope srsName="urn:ogc:def:crs:EPSG:6.7:25833">
      <gml:lowerCorner>296853.142818 6499100.771801</gml:lowerCorner>
      <gml:upperCorner>300624.658848 6501528.349657</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <RISE:RiverWaterBody gml:id="RiverWaterBody.650391-125371">
    <RISE:eu_waterbodyCode>SE650391-125371</RISE:eu_waterbodyCode>
    <RISE:insertedBy>unknown</RISE:insertedBy>
  </RISE:RiverWaterBody>
</wfs:FeatureCollection>
```
You see a `wfs:FeatureCollection` object (because the piece of GML comes from a WFS invo- 
cation), containing two properties, namely

- an instance the `gml:boundedBy` property and
- a `gml:featureMember` property.

`gml:boundedBy` contains a `gml:Envelope` object, which gives information on the geographical 
extent of the total contents. The property is mandatory part of all FeatureCollection objects 
and gives a client the opportunity to accommodate for the spatial extent represented in the 
rest of the file.

The `gml:featureMember` (typically there many more than just one contained in a FeatureCol-
lection) contains an instance of a `RISE:RiverWaterBody` feature. This feature is uniquely 
identified by the attribute `gml:id` (the identifier is `RiverWaterBody.650391-125371`) and owns 
a series of properties. Most of the properties are inherited from feature type `RISE:WaterBody` 
– these can be traced by following the line of generalisations in the UML diagram.

The first of the inherited properties is `RISE:eu_waterbodyCode`, the last one is 
`RISE:waterbodyCode`. See how the attributes in the UML class correspond to the properties 
in the GML representation. The next property comes from `RISE:RiverWaterBody` itself and is
a geometric one named $RISE$:shape. It can be seen that the UML type GM_CompositeCurve is translated into the appropriate GML type $gml$:CompositeCurve.

The remaining properties with name $RISE$:hasSegment are special, because they realise the association to $RISE$:RiverSegment features. In the UML diagram hasSegment is the associated role name of the association. As can be seen, this property has multiplicity 1..*, and in fact there are three association instances expressed in the feature instance. The referencing itself is done by xlink:href syntax.

### 3.2 Presentation Encodings

Presentation encoding standards are an important building block in serving the task of making content available to people.

Those standards let content sources encode their content in a standardised way, which is understood by software specific to the content encoding standard. The software then transforms the encoding to a presentation directly available to human senses.

One well known example of presentation encodings is HTML, which encodes nearly all content visible by web browsers in the World Wide Web. A HTML document contains both content and formatting which are displayed by HTML interpreters (usually being part of web browsers).

Although HTML will certainly play an important role in eMOTION, it will not be examined in detail, because HTML is so well known a standard.

The presentation encodings we are describing here are not as well known as HTML. They have however some special relevance regarding the thematic coverage of eMOTION.

SMIL (Synchronized Multimedia Integration Language) is an authoring language for interactive audiovisual presentations and constitutes a good way to represent multimedia content.

VoiceXML is designed for creating audio dialogs that feature synthesized speech, digitised audio, recognition of spoken and DTMF key input, recording of spoken input, telephony, and mixed initiative conversations. Its major goal is to bring the advantages of Web-based development and content delivery to interactive voice response applications.

The Speech Synthesis Markup Language (SSML) is designed to provide a rich, XML-based markup language for assisting the generation of synthetic speech in Web and other applications.

SAMPA, Speech Assessment Methods Phonetic Alphabet and X-SAMPA are computer-readable phonetic scripts based on the International Phonetic Alphabet (IPA).

The Pronunciation Lexicon Specification (PLS) is designed to enable interoperable specification of pronunciation information for both Automatic Speech Recognition (ASR) and Text-To-Speech (TTS) engines within voice browsing applications.
3.2.1 Synchronized Multimedia Integration Language (SMIL)

The Synchronized Multimedia Integration Language (SMIL) enables simple authoring of interactive audiovisual presentations. An overview of the standard is available in the main document at subsection 4.5.10.

SMIL contains collections of modules with profiles. The most interesting one is the mobile profile, described below.

The SMIL 2.1 Mobile Profile is a collection of SMIL 2.1 modules that provide support for the SMIL 2.1 language within the context of a mobile device. Such a device is expected to have sufficient display, memory, and processor capabilities to render basic interactive multimedia presentations in SMIL. The SMIL 2.1 Mobile Profile is a super-set of the SMIL 2.1 Basic Profile and a sub-set of the SMIL 2.1 Extended Mobile Profile. The SMIL 2.1 Mobile Profile is largely compatibility with the SMIL profile that the Third Generation Partnership Program (3GPP) has defined for the multimedia messaging (MMS) and the enhanced packed switched streaming (e-PSS) mobile services in its own specification ([3GPP26.246R6]).

The functionality of the SMIL 2.1 Mobile Profile may be further extended by using the SMIL 2.1 Scalability Framework. When extending the functionality of this profile, it is highly recommended to include functionality from the SMIL 2.1 Extended Mobile Profile first.

The collection names contained in the following table define the SMIL 2.1 Language Profile vocabulary.

<table>
<thead>
<tr>
<th>Collection Name</th>
<th>Elements in Collection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Animation</td>
<td>animate, set, animateMotion, animateColor</td>
</tr>
<tr>
<td>ContentControl</td>
<td>switch, prefetch</td>
</tr>
<tr>
<td>Layout</td>
<td>region, root-layout, layout, regPoint, topLayout</td>
</tr>
<tr>
<td>LinkAnchor</td>
<td>a, area (anchor)</td>
</tr>
<tr>
<td>MediaContent</td>
<td>text, img, audio, video, ref, animation, textstream, brush, param</td>
</tr>
<tr>
<td>Metainformation</td>
<td>meta, metadata</td>
</tr>
<tr>
<td>Structure</td>
<td>smil, head, body</td>
</tr>
<tr>
<td>Schedule</td>
<td>par, seq, excl</td>
</tr>
<tr>
<td>Transition</td>
<td>transition</td>
</tr>
<tr>
<td>Other</td>
<td>customAttributes, customTest, paramGroup, priorityClass</td>
</tr>
</tbody>
</table>

The collection names contained in the following table define the Mobile Profile vocabulary.

<table>
<thead>
<tr>
<th>Collection Name</th>
<th>Elements in Collection</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMIL 2.1 Mobile Profile</td>
<td></td>
</tr>
</tbody>
</table>
SMIL 2.1 Mobile Profile

<table>
<thead>
<tr>
<th>Collection Name</th>
<th>Elements in Collection</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContentControl</td>
<td>switch, prefetch</td>
</tr>
<tr>
<td>Layout</td>
<td>region, root-layout, layout, regPoint</td>
</tr>
<tr>
<td>LinkAnchor</td>
<td>a, area</td>
</tr>
<tr>
<td>MediaContent</td>
<td>text, img, audio, video, ref, textstream, param, paramGroup</td>
</tr>
<tr>
<td>Metainformation</td>
<td>meta, metadata</td>
</tr>
<tr>
<td>Structure</td>
<td>smil, head, body</td>
</tr>
<tr>
<td>Schedule</td>
<td>par, seq</td>
</tr>
<tr>
<td>Transition</td>
<td>transition</td>
</tr>
</tbody>
</table>

3.2.2 Voice Extensible Markup Language (VoiceXML)

VoiceXML is a standard format from W3C. It is designed for creating audio dialogs that feature synthesized speech, digitised audio, recognition of spoken and DTMF key input, recording of spoken input, telephony, and mixed initiative conversations. An overview of the standard is available in the main document at subsection 4.5.12.

An example of a possible application of VoiceXML is a request for pre trip planning with maps, which can be done by typing in the target location or by speech. The web service recognizes the command, i.e. “give me the fastest route to Berlin, Potsdamer Platz, please”, compares the current position and generates a route or replies like “Is city of Berlin, location Potsdamer Platz correct?”.

The user can input requests either by voice or by DTMF (means input via typing numbers).

Connected to VoiceXML and SMIL is SSML (Speech Synthesis Markup Language), currently with a working draft (see 3.2.3).

In terms of quality for audio/voice services this might become interesting since SSML can set emphasis and pronunciation to sentences which makes automated replies more realistic and voice input probably more precise to process.

The following link shows a German Initiative using VoiceXML.
http://www.xmlcityberlin.de/index.php/m/tixml/t/0/c/0/i/102.html

VoiceXML integrates some important features like grammar based on W3C Speech Recognition Grammar Specification [SRGS; http://www.w3.org/TR/voicexml20/#ref_SRGS]. You can describe synonyms like “AMEX” and “American Express” in a separate file or inline.

The following figure shows you an example using DTMF and/or speech.
VoiceXML can handle some simple conditional logic like if – else and decide where to continue. This way you can design entire menu structures for voice commands, enabling the parallel use of graphical user interfaces as well as voice control. Additional to explaining text from the GUI, which is obviously hard to read during road trips as driver, VoiceXML offers a help element. So if you are in some submenu you can just say “help” to get a detailed voice description which explains the possible commands. This feature is quite comfortable for navigations systems and an improvement in terms of security as mentioned in the feedback for eSafety.

Examples of elements available in VoiceXML are:

- `<vxml>` - Top-level element in each VoiceXML document
- `<prompt>` - Queue speech synthesis and audio output to the user. The content of the `<prompt>` element is modelled on the W3C Speech Synthesis Markup Language 1.0 [SSML; http://www.w3.org/TR/voicexml20/#ref_SSML].

---

10 Diagram from: [http://www.w3.org/TR/voicexml20/](http://www.w3.org/TR/voicexml20/)
3.2.3 Speech Synthesis Markup Language specification (SSML)

The Speech Synthesis Markup Language (SSML) is designed to provide a rich, XML-based markup language for assisting in the generation of synthetic speech in Web and other applications. An overview of the standard is available in the main document at subsection 4.5.11.

SSML is part of a larger set of markup specifications for voice browsers developed through the open processes of the W3C. It is designed to provide a rich, XML-based markup language for assisting the generation of synthetic speech in Web and other applications. The essential role of the markup language is to give authors of synthesizable content a standard way to control aspects of speech output such as pronunciation, volume, pitch, rate, etc. across different synthesis-capable platforms.

The intended use of SSML is to improve the quality of synthesized content.

One very simple example is as follows:

```xml
<speak version="1.0" xml:lang="en-US">
  <voice name="Dave">
    Hello, world; my name is Dave.
  </voice>
</speak>
```

In the example the voice named "Dave" is to pronounce the following sentence: "Hello, world; my name is Dave."

This example is contained in the following document, where also further details and examples can be found:


3.2.4 Speech Assessment Methods Phonetic Alphabet (SAMPA / X-SAMPA)

Speech Assessment Methods Phonetic Alphabet (SAMPA) and X-SAMPA are computer-readable phonetic scripts based on the International Phonetic Alphabet (IPA). An overview of the standards is available in the main document at subsection 4.5.9.

SAMPA was devised as a hack to work around the inability of text encodings to represent IPA symbols. However, as Unicode support for IPA symbols becomes more widespread, the necessity for a separate, computer-readable system for representing the IPA in ASCII decreases.

Examples of encodings using X-SAMPA:

- English father ["fA:D@((r)]
- Polish żrebak ["z\'rEbak]
3.2.5 Pronunciation Lexicon Specification (PLS)

The Pronunciation Lexicon Specification (PLS) is designed to enable interoperable specification of pronunciation information for both Automatic Speech Recognition (ASR) and Text-To-Speech (TTS) engines within voice browsing applications. An overview of the standards is available in the main document at subsection 4.5.7.

The language allows one or more pronunciations for a word or phrase to be specified using a standard pronunciation alphabet or if necessary using vendor specific alphabets. Pronunciations are grouped together into a PLS document which may be referenced from other markup languages, such as the Speech Recognition Grammar Specification [SRGS] and the Speech Synthesis Markup Language [SSML].

Pronunciation lexicons in general are widely used to customize the pronunciations on specific domains, for instance to correctly pronounce proper nouns, addresses, acronyms, etc.

PLS 1.0 will become a standard format for defining pronunciation lexicons to increase interoperability among different TTS engines.
4. Analysis of Service Standards

In this chapter service standards relevant for eMOTION are explored. Since these are distinguished mainly by their computational properties, the following issues are treated:

- Functionality
- Service interfaces
- Related data and encodings
- Service related metadata

The distinction between service related standards and content related ones (which are treated in the previous chapter) is not always strict and clear. As a rule, data belonging to some special service is treated as a content standard, if it has general significance. For example, a route description object is the significant output of a routing service and is therefore ‘content’. A response format description is only important to a service and is described at the ‘service’ description.

The chapter will mainly discuss Web Services, because eMOTION is being built as a Web Services Infrastructure. There may be exceptions if appropriate Web Service Definitions do not yet exist.

“Comparison and Evaluation” considerations for the individual service types have not been carried out as have been done in the “content chapter”, because services usually do not have the need to be harmonised like data models. To be compliant to other uses of the service definition, these have to be used “as is”, or in the sense of a “profile”, which means a more restricted interpretation of the standard. An extensive comparison of services therefore seemed unreasonable and too laborious.

4.1 Application Services

Application Services serve as the interactive interfaces of computational environments to humans.

Output documents from Work Package 1 (WP1) among other service classes have identified and documented a considerable quantity of “End User Services”. Most constitute specialised Internet applications, which are connected to a distributed network of various road and traffic related eMOTION services.

End User Services, or Application Services as they are more generally called, are targeted towards the special needs they are serving, which is why there is no Application Service per se, and hence no Application Server technical standard.

There are several standards regarding the subject of Web Accessibility, which targets towards methods and ways to make websites more accessible, particularly for physically disabled people. Recommendations and legislation from European and non-European bodies regulate the way public authorities have to arrange their own web appearances to disabled
users.

There is also a W3C standard on this topic: “Web Content Accessibility Guidelines 1.0” (WCAG).

Since eMOTION does not focus on End User Services, the Web Accessibility topic is not researched as far as the issue of website accessibility is concerned.\footnote{11 Other recommendations and legislation regarding the support of physically disabled people have a wider scope than just “Web Content Accessibility”. These are indeed may be relevant for eMOTION because they influence information modelling decisions.}

One technical standard supporting the creation of generic Web Mapping Applications (which allow to overlay web map sources from different servers and treat them as whole in a GIS-like manner) is the OpenGIS® Web Map Context (WMC) Implementation Specification. The standard describes how to save a map view comprised of many different layers from different Web Map Servers. A 'context' can be encoded and saved so that Web maps created by users can be automatically reconstructed and augmented by the authoring user or other users in the future.

4.2 Data Service

The main characteristic of Data Services is their ability to provide data in response to a data demanding request, usually named a query. There may be also the ability to insert, update or delete data. So, Data Services constitute the Web Service equivalent of data bases.

Data Services may be ‘general’ and provide data for any application schema, however there may also be special data services, which are only suited for data specially related to these services.

Data Services are very important because they form the original source of the service-supported flow of information in eMOTION.

Besides the data services from the Open Geospatial Consortium (OGC), this section also describes two services from the domains traffic messages (DATEX 2) and public transport (SIRI).

Please note that only parts of the functionality of these two services constitute “data services” in the sense of this section. Being complete domain specific services fulfilling well-defined tasks, they also contain functionality which in our taxonomy appears elsewhere. Both services, for example, offer interfaces for registration and event-driven call-back, which we treat in 4.11 Event Notification Service.

4.2.1 OGC Web Feature Service (WFS)

The Web Feature Service defines a general model for accessing data over the web, without being specialised on certain information domains. It provides a “web database interface”. An overview of the standard is available in the main document at subsection 4.1.3.
Functionality

The WFS specification defines interfaces for data access and manipulation operations on geographic features, using HTTP as the distributed computing platform. Via these interfaces, a Web user or service can combine, use and manage geodata from different sources. A WFS-client can retrieve and update geospatial data encoded in Geography Markup Language (GML) from and on multiple Web Feature Services.

The following WFS operations are available to manage and query geographic features and elements:

- Get or query features based on spatial and non-spatial constraints
- Create a new feature instance
- Delete selected feature instances
- Update selected feature instances
- Lock selected feature instances to protect them from being changed

WFS uses GML as its native encoding format, see 3.1.1.

A WFS request consists of a description of the query (expressed in Filter Expression language) or of data transformation operations that are to be applied to one or more features. The request is generated on the client and is posted to the WFS using HTTP.

A Basic WFS only includes the operations GetCapabilities, DescribeFeatureType, and GetFeature – it constitutes a "read-only" version of a WFS.

An Xlink WFS is a Basic WFS, which additionally is able to interpret associations between features in a well-defined and automatic way. It additionally has to support the operation GetGmlObject.

Finally, a Transaction WFS is a Basic WFS additionally supporting the Transaction operation. A Transaction WFS can update data in the database.

Interface

The service is defined as a Web Service and uses the HTTP protocol with the GET and POST methods. All operations are available for both methods, however full support for Transaction is only available for POST. There is also a SOAP binding which transports the interface message documents contained wrapped in SOAP envelopes (see 5.1.11).

We will describe the interface in an abstract way, which is independent from protocol and
method.

**Operation: GetCapabilities**

A GetCapabilities request is answered by a response document, which contains service metadata.

This comprises information about the service itself, its capabilities (which operations do exist and how are they invoked?), the list of supported feature types including available coordinate reference systems, the operations allowed to be performed on the features, and finally the query capabilities of the service. The latter point out in detail the subset of the general filter encoding language the WFS supports.

The GetCapabilities operation is mandatory for all WFS implementations.

**Operation: DescribeFeatureType**

For each feature type name specified in the request, the service returns the corresponding GML application schema. The response document is an XML Schema document.

By means of the DescribeFeatureType operation a client can determine the data model of the feature types advertised in the GetCapabilities response.

This is a mandatory operation and must be implemented for all WFS implementations.

**Operation: GetFeature**

A GetFeature request submits a query to a WFS, which subsequently returns a reply containing a GML document of the features, which qualify against the query.

The query contains a logical expression according to the Filter Expression standard. It combines scalar comparisons (like SQL) with spatial predicates (close to a point, contained in an area, close to a line, intersecting a line, etc.).

A typical such query would be:

```
Get all parking facility features which are contained in the area given by a polygon and which have at least 100 free spaces.
```

Several options allow the query to be modified. For example you can limit the number of features returned, omit feature properties you do not need or just query the number of qualifying features instead of the features themselves.

An Xlink WFS can be requested to follow associations from the queried features to other features to a certain depth and return the so-determined associated features along with the original ones.

GetFeature is a mandatory operation.

**Operation: GetGmlObject**
The GetGmlObject operation allows retrieval of features and elements by ID from a WFS. In response to a GetGmlObject request containing the ID of a feature or some other element like a geometry, an XML document fragment containing the result set is returned. The operation provides the interface through which a WFS can be asked to traverse and resolve Xlinks to the features or other elements it serves.

GetGmlObject is mandatory for XLink WFS implementations.

**Operation: GetFeatureWithLock**

This works as GetFeature, but additionally takes a lock on the features returned. The lock-id returned can subsequently be employed to safely alter those features in isolation.

Transactional WFS implementations are free to implement this operation.

**Operation: LockFeature**

LockFeature takes locks on selected features without returning the features themselves. The operation returns a status response. The lock-id returned can subsequently be employed to safely alter those features in isolation.

Transactional WFS implementations are free to implement this operation.

**Operation: Transaction**

A Transaction request submits a bundle of Insert, Update and/or Delete instructions on features and their properties.

A Transaction request is executed as an atomic unit. References to existing features can be constrained to locked ones. The locks are subsequently released. The operation returns a status response.

The operation is mandatory for Transactional WFS implementations.

### 4.2.2 OGC Web Coverage Service (WCS)

The Web Coverage Service (WCS) supports electronic retrieval of geospatial data as "coverages" – that is, digital geospatial information representing space-varying phenomena. An overview of the standard is available in the main document at subsection 4.1.2.

**Functionality**

A WCS provides access to potentially detailed and rich sets of geospatial information, in forms that are useful for client-side rendering, multi-valued coverages, and input into scientific models and other clients. The WCS may be compared to the OGC Web Map Service (WMS) and the Web Feature Service (WFS); like them it allows clients to choose portions of a server’s information holdings based on spatial constraints and other criteria.

Unlike WFS (see last section), which returns discrete geospatial features, the Web Coverage Service returns coverages representing space-varying phenomena that relate a spatio-
temporal domain to a (possibly multidimensional) range of properties.

The Web Coverage Service provides three operations: GetCapabilities, DescribeCoverage, and GetCoverage.

**Interface**

The service is defined as a Web Service and is using the HTTP protocol with the GET and POST methods. All operations are available for both methods. There is also a SOAP binding which transports the interface message documents wrapped in SOAP envelopes.

**Operation: GetCapabilities**

A GetCapabilities request is answered by a response document, which contains service metadata. The response is an XML document describing the service and brief descriptions of the coverages that clients may request. Clients would generally run the GetCapabilities operation and cache its result for use throughout a session, or reuse it for multiple sessions.

The GetCapabilities operation is mandatory for all WCS implementations.

**Operation: DescribeCoverage**

The DescribeCoverage operation lets clients request a full description of one or more coverages served by a particular WCS server. The server responds with an XML document that fully describes the identified coverages.

This is a mandatory operation and must be implemented for all WCS implementations.

**Operation: GetCoverage**

The GetCoverage operation is normally run after GetCapabilities and DescribeCoverage operation responses have shown what requests are allowed and what data are available. The GetCoverage operation returns a coverage (that is, values or properties of a set of geographic locations), encoded in a well-known coverage format.

Its syntax and semantics bear some resemblance to the WMS GetMap and WFS GetFeature requests, but several extensions support the retrieval of coverages rather than static maps or discrete features.

### 4.2.3 Datex 2 Data Service

The main aim of DATEX 2 is to establish standardised links between Road Operators, TICs and Service Providers to allow the easy and efficient communication of traffic and traveller Information. An overview of the standard is available in the main document at subsection 3.1.2.

DATEX 2 addresses variety of needs with different data exchange profiles: The Regular Profile offers rich functionality based on the Web Services family of protocol standards, whereas
the Low Cost Profile establishes a minimum interface being capable of carrying the full DATEX content while at the same time requiring minimum Internet access features.

**Functionality**

The DATEX 2 Standard contains mechanisms and defines objects for the exchange of data between suppliers and customers (clients). To fully specify delivery of traffic and travel related information a set of exchange metadata is defined, illustrated by the UML class diagram below.

![Figure 4-1: DATEX 2 - Exchange Metadata](image)

The attributes of the `Exchange` class are defined as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>changedFlag</td>
<td>Indicates that either a filter or a catalogue has been changed.</td>
</tr>
<tr>
<td>clientIdentification</td>
<td>In a data exchange process, an identifier of the organisation or group of organisations which receives information from the DATEX 2 supplier system.</td>
</tr>
<tr>
<td>deliveryBreak</td>
<td>Indicates that a data delivery is stopped for unplanned reasons, i.e. excluding the end of the order validity or the case when the filter expression is not met.</td>
</tr>
<tr>
<td>denyReason</td>
<td>Indicates the reason for the refusal of the requested exchange.</td>
</tr>
<tr>
<td>keepAlive</td>
<td>Indicator that this exchange is due to &quot;keep alive&quot; functionality.</td>
</tr>
<tr>
<td>requestType</td>
<td>The types of request that has been made by the client on the supplier.</td>
</tr>
<tr>
<td>response</td>
<td>The type of the response that the supplier is returning to the requesting client.</td>
</tr>
</tbody>
</table>
A subscription is a necessary pre-condition for the exchange of data. Without a subscription delivery of data is not possible.

DATEX 2 exchange operations are distinguished between four modes. Three modes are for data delivery. An Operating Mode is a set of rules and conditions that stipulate the physical transmission of data between supplier and client(s) and is defined within the subscription.

- Operating Mode 0 – Subscription Management Mechanism (a specialized Operating Mode to handle subscriptions)
- Operating Mode 1 – Publisher Push on occurrence (data delivery initiated by the publisher every time data is changed)
- Operating Mode 2 – Publisher Push periodic (data delivery initiated by the publisher on a cyclic time basis)
- Operating Mode 3 – Client Pull (data delivery initiated by the Client, where data is returned as a response).

While not handling subscriptions by an offline component Operating Mode 0 is necessary to be done with success of creating a suitable subscription before any other operation can take place. Beside of creating subscriptions this mode is used to edit and delete subscriptions.

Differing amounts of data can be delivered within a Publication. Dependent on the type of Publication (Operation Mode) the following Update Methods are possible, not all for every Operating mode:

- singleElementUpdate: If an atomic part of the data has been changed this atomic part, and only this atomic part, will be exchanged.
- allElementUpdate: If an atomic part has been changed the data complex associated with the atomic part will be exchanged.
- Snapshot: A snapshot contains all information that is available for a subscription.

**Interface**

**Operating Mode 0 – Subscription Management Mechanism**

A session is initialised by the client through opening a physical connection to the Supplier. The Supplier acknowledges the connection by using the "initSession" operation. This establishes a logical session. Both the Client and the Supplier are able to stop the Session at any time.
Figure 4-2: DATEX 2 - subscription: operating mode 0
In case of creating a new (empty) subscription the Client sends an `exchangeSubscriptionRequest` message to the Supplier which should be approved by sending a subscription identification or otherwise would be not approved transmitting a `DeniedReason`. After a subscription was created successfully the client has to fill in necessary details within the `exchangeData` message. The mandatory elements and the range of values depend from the selected Operating Mode. As an example for editing a subscription of Operating Mode 3 the mandatory elements are shown in the figure below.

![Diagram](image)

Figure 4-3: DATEX 2 - Exchange Data Reference: operating mode 3

In response the DATEX 2 System either approves a subscription by sending the `exchangeAcknowledge` message or denies the subscription by using the `exchangeDenyReason` message. A service commences immediately after the `subscriptionStartTime` and runs until the optional `subscriptionStopTime`.

When a catalogue service is used Clients must ensure they hold the same version of catalogues held by the Supplier. For this the Client must first request what versions of the catalogues are used by the Supplier. With this information he is able to assess whether it is necessary to request an update of catalogues, which are defined in the `CataloguePublication` Package. By sending the relevant catalogue reference, the Client can then request details of individual catalogue entries.

The possibility to publish filters is optional. Where used the Client sends an `exchangeFilterRequest` message. It is also possible to send customer defined filters from the client system to the DATEX 2 System.
Operating Mode 1 – Publisher Push on occurrence

In this mode, for situation publications (events, operator actions, ...), situation and situation records lifecycles have to be managed (update, cancel, and end). The supplier initialises the session. The amount of data is defined by the subscription, i.e. catalogue, filter, update method. Without use of catalogues and filters it is up to the supplier to define what is sent.

If a link failure between the Supplier and his Client(s) occurs when using Operating Mode 1, recovery mechanisms have to be built. Therefore the session management can be enhanced by the use of keep-alive functionality, monitoring and preserving the link between DATEX 2 System and the Client. If no data has been available for transmission to the client for a stipulated period and thus none has been sent to the Client, a keep-alive message will be generated and sent. On the Client side an optional exchangeAcknowledge can be sent in order to enhance the monitoring of the connection.

![Diagram of SupplierPushOnOccurrence](image)

**Figure 4-4: DATEX 2 - SupplierPushOnOccurrence: operating mode 1**

Operating Mode 2 – Publisher Push periodic

In this mode the supplier initialises the session and delivers data periodically. The amount is defined by the subscription, i.e. catalogue, filter, update method. Without use of catalogues and filters it is up to the supplier to define what is sent.

An optional exchangeAcknowledge can be sent by the Client to enhance the monitoring of the connection.
Operating Mode 3 – Client Pull

In this mode the client initialises the session and requests data as a snapshot. The kind of data can be defined in the subscription. Without use of catalogues and filters it is up to the supplier to define what is sent.
For further information about Data Exchange in the context of DATEX 2 refer to DATEX II – Exchange Mechanism\textsuperscript{12}.

4.2.4 Service Interface for Real Time Information (SIRI)

The Service Interface for Real Time Information (SIRI) specifies a European interface standard for exchanging information about the planned, current or projected performance of real-time public transport operations between different computer systems. An overview of SIRI is available in the main document at subsection 3.2.13.

Functionality

SIRI emphasizes extensibility, the addition of additional services is expected. The following list shall give an overview over the mentioned SIRI-Services:

- The \textbf{Production Timetable Service} exchanges information about the expected operation of a transport network for a specified day in the near future. It is typically requested a few hours or days before the desired day and includes all changes to the timetables known at that stage. This information can also be used for e.g. smart devices with base timetables.

- The \textbf{Estimated Timetable Service} provides operation-details of the transport network for a period within the current day. It covers information about real time deviations from the timetables and control actions affecting the Timetable (cancellations, additional Journeys and Detours). Apart from Automated Vehicle Location (AVL) systems this service is also suitable for smart devices with real-time timetables.

- The \textbf{Stop Timetable and Stop Monitoring services} deliver information about actual vehicle arrivals and departures at a stop or Monitoring Point. Typically this service is used for departures within the next 20 - 60 minutes for display to the public, but is also suitable for providing departure boards on all forms of device.

- The \textbf{Vehicle Monitoring service} provides information about the current location and expected activities of a particular vehicle, and is suitable for onboard displays, visualisation of vehicle movement, and for exchanging information on roaming vehicles between different control systems. The deliveries of this service are out of scope of eMotion and are not mentioned in the forthcoming feature-description.

- The \textbf{Connection Protection Services (Connection Timetable and Connection Monitoring)} allow transport operators to exchange information about the real-time management of interchanges between feeder and distributor vehicles arriving and departing at a connection point. This service is suited for providing e.g. passenger-information in case

\textsuperscript{12} [EC, 2005] DATEX II - EXCHANGE MECHANISM, Deliverable 4.1, Document Number D2_4.1v1.01, European Commission DG TREN, 08th February 2005
of delayed trains to let them know that a local bus service will wait for them. Because of
the public transport internal kind of information-context, this service seems to be out of
scope and will not be described in detail.

**Interface**

SIRI intentionally tries not to restrict the usage of the Standard to special implementations,
but discusses different realizations to leave the decision for the preferred implementation to
the developers. The normal way of exchanging SIRI-Messages is via HTTP using the POST
method with an XML-document containing the encoded message-content as simple attach-
ment. Another way of transmission is the usage of SOAP-envelopes containing the SIRI-
messages. The XML-structure is modularized into separate packages and uses the ISO/IEC
8859-15 character set exclusively for encoding.

Endpoint addressing is allowed at two different capability-levels. The implicit addressing ex-
changes no endpoint addresses and leaves the addressing to the transport layer against
which explicit addressing may be specified as return within the request-structure which al-

dows protocol independent access to the endpoint properties. The Functional Service ad-

Endpoint addressing is allowed at two different capability-levels. The implicit addressing ex-
changes no endpoint addresses and leaves the addressing to the transport layer against
which explicit addressing may be specified as return within the request-structure which al-

Several different mappings into a Server and Client WSDL binding of the SIRI schema are
described as examples. They include different variants of SOAP usage such as SOAP/RPC
and SOAP/Document, where the simpler SOAP/RPC is recommended. The SIRI WSDL may
be built directly from the types defined within the main XSD-file and is defined as one file for
the Producer Server side and one for the Subscriber and Consumer side. While the imple-
mentation of SIRI as a SOAP Web Service is not mandatory, it must be compliant with the
WSDL-files provided by SIRI if it is provided.

SIRI uses two main complementary interaction-patterns for data exchange. An implementa-
tion may support both or chose the most efficient pattern according to the nature of their im-
plementation – the Request/Response and the Publish/Subcribe pattern.

The Request/Response-Pattern is the basic interaction-pattern for the fulfillment of a client’s
ServiceRequest to a server that provides a specific SIRI functional service by the response
of the requested ServiceDelivery. They are also used for the interactions that make up other
patterns, such as Publish/Subcribe (see subscription-operation). The following Requests
are contained within a ServiceRequest:

- ProductionTimetableRequest (ProductionTimetableDelivery) - Timetables
- EstimatedTimetableRequest (EstimatedTimetableDelivery) - Timetable Changes
- StopTimetableRequest (StopTimetableDelivery) - Stop Timetable
- StopMonitoringRequest (StopMonitoringDelivery) - Visits to stop
- VehicleMonitoringRequest (VehicleMonitoringDelivery) - Vehicle Movements
• **ConnectionTimetableRequest (ConnectionTimetableDelivery)** - Connections
• **ConnectionMonitoringRequest (ConnectionMonitoringFeederDelivery, ConnectionMonitoringDistributorDelivery)** - Connection changes
• **GeneralMessageRequest (GeneralMessageDelivery)** – Travel News

A data-delivery is made as one-step *Direct Delivery* or as two-step *Fetched Delivery*. In the *Direct Delivery*, the payload is immediately transmitted as the content of a single message within a Response or a Notification in case of a subscription. As there is the burden of managing and queuing the messages distributed to the clients, another type of Delivery is allowed within SIRI.

The *Fetched Delivery* allows the Consumer to defer the sending of the full payload until it is ready to process it. After Subscription the consumer is notified of Data Availability in case of the existence of new or updated data by a *DataReadyNotification* which is to be answered by a *DataReadyAcknowledgement*. To reduce the number of notifications, the messages may be aggregated. The following Polling of the Data can be done anytime but is normally initiated after the *DataReadyNotification* by the transmission of a *DataSupplyRequest*. It prompts the producer to finally transmit the data which is to be responded within the *ServiceDelivery*.

If the amount of the data to be delivered is large, some implementations could have a need to split the data-transmission into several parts. For the *Multipart Dispatch* of a *ServiceDelivery* a *MoreData* element is used to indicate that the *DataSupply-Response* does not contain all updated data.

**Operation: Capability**

This Operation is used for the first stage of communication in SIRI. It is used during the *Service Discovery* which covers several different levels. The first step is the universal *Discovery of SIRI-supporting Servers*. However, due to the usage of SIRI for Automated Vehicle Management Systems (AVMS) where participants are usually known by each other this functionality is not yet specified in the standard. If this service is nevertheless needed for the implementation, the usage of general purposes web service discovery protocols such as WSDL is suggested.

Once a SIRI server is known, the consumer has to know the SIRI functional services the server supports and at which version level. For the *Discovery of Capabilities of a SIRI-Server* a *CapabilityRequest* is to be sent for each functional service needed, encapsulated in one request which will be answered by a *CapabilityResponse*.

In the next step, once a client knows the provided SIRI functional services, the consumer needs to know which data can be requested and if it is allowed to access this data. For this *Discovery of Functional Service Coverage* there exists a separate service capability discovery request (**xxxRequest**) for each functional service to discover the exact capabilities of the implemented service which is answered by the specific **xxxCapabilityResponse**.

Apart from system level security-mechanisms as e.g. VPN tunnelling, SIRI supports the ac-
Access control on application level allowing a given service provider to specify the access to specific types of content or specific levels of resource usage by the authentication of the requestor or subscriber. For this purpose the requestor is validated against an Access Permission matrix by the server on the basis of its participant reference. It is suggested this validation is done once during a subscription rather than for each request.

During the Discovery of the service capabilities the specific permissions for the requested functional service may be transmitted using the \textit{xxxPermission} element encapsulated in a \textit{xxxCapabilityResponse}. These cover:

- Common \textit{xxxServicePermissions} for each functional service
- \textit{OperatorPermissions} to specify each operator allowed to use the service
- \textit{LinePermissions} to specify which line-data is accessible for the subscriber or consumer
- \textit{ConnectionLinkPermissions} specify the connection-links which are allowed to be accessible by the subscriber or consumer
- \textit{StopMonitorPermissions} specify the Monitoring Points accessible for the participant
- \textit{VehicleMonitorPermissions} specify the accessible Vehicle Monitor references
- \textit{InfoChannelPermissions} specify the Info Channel instances the participant is allowed to access.

**Operation: Subscription**

The Publish/Subscribe interaction allows the asynchronous detection of real-time events by a producer service which sends notifications to one or more interested consumers. It requires the consumer to send a request message for the subscription of a specific functional service to be notified in case of a new event or a data-update. The \textit{SubscriptionRequest} is to be answered by a \textit{SubscriptionResponse}.

The following diagram shows the structure and the properties of the different Subscription-Requests which are relevant for the eMotion project.
Operation: TerminateSubscription

Every Subscription is to be terminated by a **TerminateSubscriptionRequest** if it is no longer needed. The structure of a TerminateSubscriptionRequest can be seen in Figure 4-8. The subscriber can chose between the termination of a specific subscription or all subscriptions belonging to him. In case of a system failure all subscribed services have to be cancelled by using the option **All** (see subscription recovery). The requests are answered by a **TerminateSubscriptionResponse**.
Operation: Subscription Recovery

For the case of a system failure SIRI distinguishes between Client-Failure, Server-Failure and the Interruption of the communication link and discusses them separately. The following description shall give a short overview of the operations to be done for the recovery of existing subscriptions by the subscriber.

In case of a **client failure** the problem is that the subscriber loses the previous state of its subscriptions. For the recreation of all subscriptions of the Subscriber, all old ones at the server must be deleted by a **TerminateSubscriptionsRequest** using the **All** option and be created again subsequently.

There are two options to discover a **server failure** – the continuous HeartbeatNotification by the Server and the periodical transmission of a CheckStatusRequest to the subscription-server. The **HeartbeatNotification** is the simpler strategy for detecting a server-failure. The failure becomes obvious when no more Notifications are submitted by the server. As a consequence a new set of subscription must be recreated if detected. For assuring the correct function of a server the subscriber may create and send **CheckStatusRequests** periodically to the producer. The Request is then be answered by a **CheckStatusReply** containing the start time of the subscribed service. If the subscription-time is older than the returned time the subscriber deletes all subscriptions and creates them again.

The interruption of the communication link is normally detected because of a timeout after a transmitted Request. In this case the messages are simply retransmitted until a Response arrives within the timeout-interval. If the failure occurs during a DataDelivery, the consistent states of the participants cannot be guaranteed. Because of the possibly reset update flag, a new **DataSupplyAllRequest** has to be sent to the producer.

Operation: ProductionTimetable

A **ProductionTimetableRequest** demands delivery of a ProductionTimetable response. The request allows the specification of a **ValidityPeriod** and a **TimeTableVersionReference** which is used to communicate only the differences to the declared timetable. There are several filtering possibilities to be done to reduce the amount of matching data such as the specifica-
tion of the desired Operator, a specific Line and a certain Direction.

![Figure 4-9: SIRI – ProductionTimetableRequest](image)

**Operation: EstimatedTimetable**

The **EstimatedTimetableRequest** is answered by the delivery of a EstimatedTimetable which indicates the actual changes of the timetable regarding the current state of all vehicle-journeys. As in most TimetableRequests it is possible to filter the results to be returned by the specification of the desired Operator, Line and Direction. Additionally a maximum **PreviewInterval** may be specified to limit the real-time prediction-data returned.

![Figure 4-10: SIRI - EstimatedTimetableRequest](image)

**Operation: StopTimetable**

The **StopTimetableRequest** demands a StopTimetable delivery. The request requires the specification of the **DepartureWindow** which defines the time-frame for which data is to be supplied as well as the **StopMonitoringPoint** for which StopVisits will be returned. For additional filtering the amount of data returned, there is again the possibility to specify the desired Line and DirectionReference.
Operation: StopMonitoring

A **StopMonitoringRequest** is set for the response of a StopMonitoringDelivery which returns information about one or more StopVisits. Associated with each MonitoredVehicleJourney instance. The detail level of the StopVisits may be parametrized depending on the application requirements by the **StopVisitDetailLevel** option. For the optimization of data traffic **PreviewInterval**, **MaximumStopVisits** and **MinimumStopVisitsPerLine** can be used. The PreviewInterval and MaximumStopVisits limit the number of returned StopVisits whereas **MinimumStopVisitsPerLine** ensures at least a minimum of one or a given number of StopVisits per line. As well as in most TimetableRequests the number of results can be reduced through filtering by **Operator**, **Line** and **Direction**. Additionally the results can be filtered by visits for vehicles going to a specific **Destination**. The **MonitoringRef** specifies the mandatory **StopMonitoringPoint** for which StopVisits will be returned.

---

**Figure 4-11: SIRI - StopTimetableRequest**

**Operation: ConnectionTimetable**

The **ConnectionTimetableRequest** requests the delivery of a ConnectionTimetable in re-
sponse. The Request allows the specification of a certain ArrivalWindow as the time-frame for which the data is to be supplied. Reference is the arrival time of the feeder at the desired ConnectionLink. StartTime specifies the earliest and EndTime indicates the latest arrival-time. By the specification of the desired Line or the Direction, the returned data may be filtered independently.

![Figure 4-13: SIRI - ConnectionTimetableRequest](image)

### 4.3 Mapping Service

Mapping Services generate maps and make them available to the client by means of a request/response pattern. The data source of the maps may be opaque and hidden in the mapping service. It can also be one of the above data services.

One of the outstanding properties of Mapping Services is the introduction of a new and easy way to manage level of data integration from different sources. By leaving transparent those parts of maps which do not carry information, maps can be easily combined from distributed sources by requesting them with identical geo-reference and size.

#### 4.3.1 OGC Web Map Service (WMS)

The standard specifies a Web Service, which produces maps of geo-referenced data. A "map" is a visual representation of spatial data; a map is not the data itself. An overview of the standard is available in the main document at subsection 4.1.4.

**Functionality**

When requesting a map, a client may specify the information to be shown on the map (one or more layers), possibly the styles of those layers, what portion of the Earth is to be mapped (a bounding box), the projected or geographic coordinate reference system to be used, the desired output format, the output size, background transparency and colour. When two or more maps are requested from different WMS with the same bounding box, coordinate reference system, and output size, the results can be accurately layered to produce a composite map.
Interface

The service is defined as a Web Service and uses the HTTP GET protocol. The necessary parameters of the particular WMS operation requests are therefore encoded as key-value-pair parameters. The formats of the responses depend on the particular operation and can be either XML documents (e.g. operation GetCapabilities), image formats (GetMap), or various document formats including Geography Markup Language (GetFeatureInfo).

Operation: GetCapabilities

This operation, addressed by means of the key-value-pair parameter REQUEST=GetCapabilities, returns service-level metadata, which is a description of the service's information content and acceptable request parameters.

The GetCapabilities response document returns metadata regarding the service (name, title, abstract, keywords, contact information, fees, access constraints) and its implemented interface as well as a list and documentation of the layers and styles provided by the service.

Operation: GetMap

The key-value-pair parameter REQUEST=GetMap selects the GetMap operation, which returns a map image in raster format (GIF, PNG, JPEG, and others) or vector format (SVG, others). The most important parameters of the HTTP GET request encoding are as follows:

The content of the map is selected by the LAYER and STYLE parameters, which select a list of map layers and associated styles from the WMS. Both, available layers and styles, are advertised in the GetCapabilities response document.

The visible portion of the map layers is chosen by means of the BBOX parameter, which specifies a rectangular area in some Coordinate Reference System, which is requested by means of the CRS parameter. Typically (but not exclusively) the CRS is selected in the EPSG namespace. (EPSG, European Petroleum Survey Group, now: OGP Surveying and Positioning Committee, whose activities include administration of an extensive database of geodetic parameters.) The visible portion of the layers is mapped to a rectangular pixel area, which is requested by a HEIGHT and WIDTH parameter.

A FORMAT parameter lets you select a specific output format, like PNG, GIF, JPEG, or SVG. If the chosen format supports transparency, requesting TRANSPARENT=TRUE delivers pictures, where all "background" of the layers (each no-information pixel) is made transparent.

Example\textsuperscript{13}:

\textsuperscript{13} Note: This is a link invoking a free OGC WMS of NASA Jet Propulsion Laboratory. It provides world wide maps in lat/lon coordinates and supports OGC WMS version 1.1.1 and not version 1.3 as is described in this standards analysis.
Besides the described important basic parameters there is a set of additional ones, which controls selection criteria on pictures on axes other than planar geometry. The $TIME$ parameter lets the client select maps by instances or intervals in time, $ELEVATION$ provides a means for selecting on elevation values. More of these alternate sample dimensions can be advertised in the GetCapabilities response document of a particular service and made use of in GetMap requests, for example: $PRESSURE$, $TEMPERATURE$, $WAVELENGTH$, etc.

There is also the possibility to add so-called “vendor specific parameters”, however, there is the requirement that these are advertised in the GetCapabilities response document and that the service returns sensible results even if the vendor specific parameters are omitted from the GetMap operation request.

**Operation: GetFeatureInfo**

The operation addressed by REQUEST=GetFeatureInfo returns information about particular features shown on a map given their image coordinates. This operation is optional and is therefore not necessarily available on every WMS.

The key-value-pair parameters of the GetFeatureInfo operation re-use those of the GetMap operation and provide additional ones to specify the position of the feature(s) to be queried, the layers that are to be considered regarding the feature information contained, and the format of the information which is to be returned. The position (parameters $I$, $J = column$, $row$) is given in pixel coordinates. The layers subject to GetFeatureInfo are specified by means of the QUERY_LAYERS parameter. The selection of information format to be returned is effected by the INFO_FORMAT parameter, which can request formats from those advertised in the GetCapabilities response document. A typical output format of the GetFeatureInfo request is GML, which means that the feature information is returned in Geography Markup Language.

### 4.3.2 OGC Web Map Service (WMS) with Styled Layer Descriptor (SLD)

The SLD enhancement of the OGC Web Map Service extends the attractive concept of combining maps from different WMS sources by just overlaying them, by introducing styling from the client side. An overview of the standard is available in the main document at subsection 4.1.5

**Functionality**

The SLD profile of WMS specifies a Web Service, which extends the functionality of a WMS.

The core functionality the specification adds is providing a so-called SLD document to ac-
company the invocation of the GetMap-operation. By referring to the underlying feature model, this SLD document precisely specifies how the styling of the map shall be accomplished.

The SLD document refers to the individual feature types a WMS layer is composed of and for each of these feature types lets you specify a number of so-called Rules.

Each rule can be restricted to:

- a range of scale factors the rule is bound to, and
- a filter expression selecting features from the complete set of features in the map.

Each Rule again assembles so-called Symbolizers, which are responsible for attaching graphic symbolisation to the geometry of the features. There are Symbolizers for Points, Lines, Areas, Text and Images.

Besides this core functionality, the SLD WMS also receives an extended interface which makes it suited for the extended functionality. Among these there are operations for querying schema metadata and legends.

**Interface**

While the “pure WMS” only provides a http interface for the method GET, the SLD WMS also defines a POST variant because this is more suited for transporting SLD documents.

The individual operations are as follows:

**Operation: GetCapabilities**

This operation is very similar to the original WMS operation. The response additionally includes a short section giving information about the extent of dynamic styling facilities supported.

**Operation: DescribeLayer**

This optional operation is new compared to “pure WMS”. For Layers, which contain data accessible to SLD styling, this operation provides information about this data. If the data source is a WFS, the URL of that WFS together with the feature types which make up the layer are then returned in the response. A SLD WMS can style data from WFS and WCS.

The URL of the data source can subsequently queried for more relevant information regarding the data store semantics. A WFS can, for example, be asked to return schema information for the relevant feature types by means of the DescribeFeatureType operation.

**Operation: GetMap**

The GetMap operation is basically the same as with the “pure WMS”. Additional parameters provide for the transport of the SLD document, which (using the SE standard) controls the
styles.

**Operation: GetFeatureInfo**
The same applies also to the optional GetFeatureInfo operation.

**Operation: GetLegendGraphic**
This is a new operation of the SLD WMS. Its implementation is optional.
The operation returns a small picture, suitable for being used as a map legend, for a specific Rule of the SLD definition.

### 4.4 Routing Service
Routing Services calculate routes according to criteria specified by the client. These comprise starting point, end point, possible way points, preference options. Routing for public transport planning or multi-modal routine additionally requires time and day when the journey is to begin or end.

Routing Services run on local data, which contain the necessary topological network information, respectively with the time-tables and line information for public transport journey planning. This may be supplemented by historic traffic flow data and dynamic information on messages and traffic flow forecasts.

#### 4.4.1 ISO 19133 - Section 7.3, Package: Navigation Service
ISO 19133:2005 describes the data types, and operations associated with those types, for the implementation of tracking and navigation services. An overview of the standard is available in the main document at subsection 3.4.5.


**Functionality**
The standard defines a *Navigation Service*, the functionality of which encompasses the functions of

- routing: finding an optimal (according to some cost function) route through a network from a start position to an end position,
- route traversal: the execution of a route, though the use of instructions at each node, and
- tracking: the process of following the position of a vehicle in a network and associating it to steps in a route.
Interface

As all standards of the ISO 19100 series of standards, the definition of the Navigation Service is on a conceptual level. So there is no concrete and explicit binding to any sort of network protocol, however the definitions are suited to being made available though web services.

The definition explained here is the interface type \textit{NS\_NavigationService}.

**Operation: capabilities**

This operation returns the \textit{route request types} which can be properly handled by the Navigation Service. Possible route request types are:

- \textit{basic}: Ordinary routing service with static time usage
- \textit{predictive}: Routing for some time in the future can be performed
- \textit{dynamic}: Routing is being periodically redone to accommodate for changes during a journey
- \textit{complex}: A cost function can be specified, pricing.

**Operation: route**

The operation delivers an \textit{NT\_RouteResponse} on a \textit{NT\_RouteRequest}.

The \textit{NT\_RouteRequest} contains the following data items:

- \textit{Route request type} as explained above.
- \textit{Type of vehicle} to be used, including pedestrian, public transport, dangerous cargo, etc.
- The \textit{way-point list}, containing start point, end point and intermediate points, given by a flexible location model encompassing coordinates, addresses, POIs, etc.
- An \textit{avoid list} containing items to avoid.
- \textit{Departure time or arrival time}, both being allowable time intervals.
- An extensible \textit{cost function type} selecting among distance, time, number of turns. An explicit cost function may be specified for “complex” services.
- \textit{Route preferences} like scenic or easiest.

A set of \textit{advisories} to be contained in the response: The include items like landmarks, crossroads, road-merge, ramp-merge, toll-booth, etc.

A flag whether the request shall be dynamically updated together with a \textit{refresh interval}.

Options whether the response shall contain full route instructions, route maps, and/or the route geometry.

The operation delivers an \textit{NT\_RouteResponse} as follows:

Besides the request information which caused the route to be generated, the response first of all contains the calculated route, which is an object of type \textit{NT\_Route}, which we explain in 2.11.1.
Additionally, objects of type NS_RenderingResponse are returned, which contain the route rendered as maps, voice streams, explanatory text, etc.

4.4.2 OGC OpenLS - Core Services, Part 5 Route Service

The OGC OpenLS standard defines Location Based Services for a “GeoMobility Server” platform. It defines the interfaces for a couple of core service types, namely Directory Service, Gateway Service, Location Utility Service, Presentation Service, and Route Service. An overview of OGC OpenLS is available in the main document at subsection 3.4.7.

This description refers to Part 5 of this standard, which is named “Route Service”. It is a network-accessible service that calculates routes in a network. The OGC Route Service definition is very close to the ISO 19133 NavigationService explained above.

Functionality

The OpenLS Route Service determines a route in a network, given start point, end point and possible way points. The requestor can also specify preferences and places to avoid. By restricting the search to a specific time of departure or arrival the interface can also be utilised for finding transport plans in public transport networks.

The network itself is hidden in the service.

Interface

The OpenLS Directory Service specification defines an interface for accessing and retrieving directory information. It is defined as a pair of XML request and response documents, which can be transmitted by means of various distributed network architectures, among these (then being a typical web service) HTTP and the POST method.

Operation: Route

The RouteRequest document consists of several parts, which stand for the various options of the request.

The RoutePlan contains the information necessary to define a possible route:

- A list of way points including start point, end point and possibly intermediate points. Points are given by the Location construct as explained in 2.4.5, which means that they can be given by coordinates, but also by addresses or POI references.
- A route preference, which selects key words like Fastest, Shortest, Pedestrian, etc.
- An avoid list, which contains areas, locations and features in which the route should avoid passing through.
- Expected start time or expected end time: This is necessary for setting up proper constraints for public transport journey planning.
- A flag “use real time traffic” to request making use of knowledge about dynamic traffic or transport information when calculating the route.
A RoutePlan may alternatively be given by reference to a so-called RouteHandle. This is an identifier returned by Route requests initially fed with an explicit RoutePlan and it refers to the parameters then specified. This means that in contrast to many other typical web interfaces, the Route operation constitutes a stateful interface.

Other parts besides the RoutePlan request for specific response formats which can be supplied by the operation.

- Route instruction request effects the return of turn-by-turn route instructions and travel advisories in a text, voice, or other representation format.
- Route geometry request makes the service return the route in the form of an explicit geometry in GML.
- A route map request (containing a proper geo-reference) demands for returning a link to an image of the route.

The RouteResponse document contains the following items:

- A RouteHandle if this was requested.
- A route summary which describes the overall characteristics of the route.
- A route geometry, if this was requested.
- A route instruction list, if this was requested: It contains a list of turn-by-turn route instructions and advisories, formatted for presentation. It may also contain turn-by-turn geometries and descriptions.
- A route map, if this was requested.

4.5 Public Transport Journey Planning Services

This section collects services for Public Transport Journey Planning.

The focus of functionality of the services collected is Journey Planning, however they are quite different and also overlap with other service type definitions.

JourneyWeb, for example, is also an extensive interface against public transport information in general, which means it is also a Data Service.

OTA and FerryXML own booking operations, which by their very nature are more complex than the Journey Planning interface proper.

DELFI and JourneyWeb are special in being interfaces which allow the integration of other Journey Planning services by invoking their interfaces.

4.5.1 JourneyWeb

JourneyWeb is an extensible protocol for dynamic data exchange over the Internet between multimodal public transport journey planners. An overview of the protocol is available in the main document at subsection 3.5.9.
Functionality

JourneyWeb is a Client Pull application. JourneyWeb Request and Response messages are exchanged using the HTTP POST protocol. Request and Response formats are defined by an XML Schema.

The Response contains a DataDate element, which indicates the date on which the source database used to handle the request was last built.

Multiple requests may be made in a single Request element, with each identified by a unique RequestID. A Language may also be specified on the request so that data names are returned in the client’s preferred language where these are available.

Response elements include a Message element which includes error exception and/or diagnostic information relating to the request.

Interface

7 types of operation are handled by JourneyWeb.

OPERATION: POINTS

The PointsRequest is answered by a PointsResponse, which contains a list of remote stops and their locations for the requested NPTG locality or coordinate (specified as a UKOS or IrishOS grid reference).

OPERATION: JOURNEYS

The JourneysRequest is answered by a JourneysResponse, which contains a list of best journeys between a combination of stops supplied in the request.

The parameters of the JourneysRequest may be categorized in four groups:

- Scoping parameters - The fundamental constraints such as: Origin(s), Destination(s), Time Constraints, Number of results to return and whether journeys are returned for each origin to any destination, or any origin to each destination. A seed may be added to each Origin/Destination to indicate: number of changes available, last service being travelled on (to avoid interchange if this service could continue to be in the onward journey), start time for the current journey and walk distance still available.

- Additional scoping parameters: Via and Not-Via points, and Modes, Operators and Services to be included or excluded.

- Computational parameters – how the calculation should be carried out: Algorithm, Walk Speeds, Maximum Walking Distance, Interchange Speeds and Accessibility Hazards

- Leg Detail parameters – controlling the level of detail to be returned for legs: which intermediate stops to return, whether to include tracks, textuals instructions, detailed interchange information, real time information, and a statement of the mapping system.
understood by the client.

The JourneysResponse contains a list of Journeys, each of which consists of a sequence of Legs. Legs may be of types: TimedLeg, FrequencyLeg, ContinuousLeg or InterchangeLeg. Contents of Timed, Frequency and Detailed legs are summarised in the table below.

<table>
<thead>
<tr>
<th>Leg element</th>
<th>Description</th>
<th>TimedLeg</th>
<th>FrequencyLeg</th>
<th>ContinuousLeg</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceOrigin</td>
<td>Start point of service providing leg. Both Place and Timing Information.</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>Intermediate A</td>
<td>Collection of timing points between ServiceOrigin and LegBoard. Both Place and Timing Information.</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>LegBoard</td>
<td>Describes point at which passenger board the service. Both Place and Timing Information.</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>IntermediateB</td>
<td>Collection of timing points between LegBoard and LegAlight. Both Place and Timing Information.</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>LegAlight</td>
<td>Describes the point at which the passenger alights from the service. Both Place and Timing Information.</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>IntermediateC</td>
<td>Collection of timing points between LegAlight and Service Destination. Both Place and Timing Information.</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>ServiceDestination</td>
<td>End point of the service providing the leg. Place information.</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>LegTrack</td>
<td>A map plot of the track</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>Notes</td>
<td>Comments on the use of the leg</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>Service</td>
<td>Details of the Service</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>OriginDestinationTimes</td>
<td>Times for complete trip of the vehicle</td>
<td>-</td>
<td>o</td>
<td>-</td>
</tr>
<tr>
<td>BoardAlightTimes</td>
<td>Describing timing information for frequency based service – the average duration of the leg</td>
<td>-</td>
<td>•</td>
<td>-</td>
</tr>
<tr>
<td>DurationWindow</td>
<td>Describes window of opportunity and typical journey time</td>
<td>-</td>
<td>•</td>
<td>•</td>
</tr>
</tbody>
</table>

The InterchangeLeg describes a walk leg within a station or airport, including information on leg Origin and Destination, a description of the path, accessibility hazards and duration, and optionally mapping information.

**OPERATION: LEGDETAILS**
The LegDetailsRequest is answered by a LegDetailsResponse, which contains information about specific journey legs.

A LegDetailsRequest consists of a LegId, to identify the journey leg for which additional details are required, and LegDetail parameters as described in the JourneysRequest. The LegDetailsResponse contains requested Legs as described in the JourneysResponse.

**OPERATION: TIMETABLES**

The TimetableRequest is answered by a TimetableResponse, which contains a matrix of timetable data. Timetables may be requested between specific stops, using a StopTimetableRequest, or for one or more Services using a ServiceTimetableRequest.

A StopTimetableRequest must contain Origin and Destination. The request may optionally specify Modes, Operators and Services to be included or excluded.

A ServiceTimetableRequest specifies the Services either using OperatorCode and ServiceNumber with optional Direction, or using a PrivateID (returned from a previous request and specified to the remote server) and optional Direction. A URL to a schematic map may also be requested.

Both StopTimetableRequest and ServiceTimetableRequest contain a TimetableRequestParametersGroup, which specifies the Date and whether Generic or specific timetable should be returned, and optionally specifies: Earliest Departure Time; Latest Departure Time; the Maximum Number of Columns to return; the First Column to return; and whether to return Timing Points Only in the response.

A Timetable response contains the following elements:

- Description – containing details of services and/or stop requested
- date and day range for which the timetable has been generated (First Date, Last Date and Days)
- StopColumn contains a list of Stops defined as RowPlaceStructure Elements. For each Stop, the Stop Activity (defines whether arrival and departure times are specified) and Stop Type (origin, intermediate, destination and/or origin destination) are specified
- an ordered list of corresponding numbered TimeColumns. Each TimeColumn contains
  - a Header detailing a ServiceReference, and optionally Days of Operation and Notes or Note References
  - a list of Times (optionally including the Stop Activity and whether the Stop is timing point) or Labels corresponding to each Stop Activity
- details of Services referenced in the TimeColumns (Operator, ServiceNumber, Direction, Mode)
- an optional list of Notes (which may have been referenced in the TimeColumn...
OPERATION: STOPEVENTS

The StopEventsRequest is answered by a StopEvents Response, which contains returns arrivals and departures (scheduled and/or real time) for a stop at a time specified in the request.

The StopEventsRequest contains: one or more NaPTANIDs which identify the stop or stops; a StartTime and Range to indicate the departure window; an ArrDep indicator to indicate whether arrival and/or departure times should be returned; and a FirstServiceEventOnly flag to specify whether only the first of each service after the departure time should be returned.

The StopEventsRequest may also specify: whether IntermediateStops should be returned in the response; whether Realtime information should be returned if available; Modes to be included or excluded; Operators to include or exclude; Services to include or exclude; Origins to specify stops the returned services should have passed through; and Destinations to specify stops the returned services should pass through.

The StopEventsResponse consists of a list of Events and Services associated with these events. Each Event describes a movement at a Stop, consisting of timing details and reference to a service. Depending on parameters passed with the request, timing details may also be included for service Origin, Destination and previous or onward Intermediate Stops. Event time information may include Timetabled, Recorded and Estimated arrival and departure times, or Frequency of service.

OPERATION: SERVICES

The ServiceRequest is answered by a ServiceResponse, which contains a list of services for which the remote journey planner can return timetables.

The ServiceRequest must specify at least one of: OperatorCode; ServiceNumber (also allowing specification of the substring to be contained at the beginning of the ServiceNumber); or Direction of the Service. The request may also be constrained by mode.

The ServiceResponse consist of a list of Service Elements. The Service details OperatorCode, Service Number, Mode and optionally a Direction. The Service may also contain: an Operator Name; a Description of the service; FirstDate, LastDate and DaysOf Operation (eg Monday to Friday, Bank Holidays); the Destination Board for the vehicle; and a TimetableLink (which indicates whether timetable matrix data is available, and may also contains URLs for printed times and schematics, a PrivateID to be used in a ServiceTimetableRequest, and service Direction) for the service.

OPERATION: OPERATORS

The OperatorsRequest is answered by a OperatorsResponse which contains a list of transport operators supported by the remote journey planner.
The OperatorRequest may contain a list of Modes for which Operators are required. The OperatorResponseStructure contains a list of Operators characterized by Mode, Operator Code and customer facing Name.

**USAGE**

These requests are expected in the following combinations:

- POINTS + JOURNEYS
- POINTS + JOURNEYS + LEG DETAILS
- SERVICES + TIMETABLES
- OPERATORS + TIMETABLES
- STOP EVENTS

### 4.5.2 Nationwide Electronic Time-table Information - Durchgängige Elektronische Fahrplaninformation (DELFI)

DELFI is a system for distributed public transport journey planning in Germany. An overview of the system is available in the main document at subsection 3.5.2.

**Functionality**

The main component of the DELFI system is a so-called search controller. This controller communicates with the DELFI Servers in order to retrieve the information about the partial connections within responsibility of one public transport operator.

A meta database knows which server provides the requested data for which area. In contrary to a solution where all time-table data has to be integrated into a single data base the DELFI data exchange is limited to the meta database which contains only the responsibility and contact information.

The DELFI approach is also used for the European journey planner EU-Spirit. The EU-Spirit Journey planner contains the following components:

- The Ring Origin Destination Identification (RODI) has the task to clarify definitely at which location customers want to start their itinerary and where they want to travel to.
- Once origin and destination point have been identified, it is task of the Ring Connection Composer (RCC) to build the total itinerary.
- The Reference Ring Data Base (RRDB) is a data base containing information on local sys-
tems. It has to be made accessible to other local journey planners or to the Ring components in order to give them the ability to fulfill their tasks. The data included is called Ring Data and may be data of the local system itself or meta data of local systems. The Ring Data helps local journey planners and Ring components to interpret received external information, to address their requests etc. Copies of a specific partial content of the RRDB are sitting directly at the local journey planners and at the other Ring components and are updated offline in long-term intervals with data provisions from the RRDB.

Every DELFI provider has to extend his information system with an interface which is able to provide partial itinerary information in a way which is well suited to be composed by the search controller.

The following figure shows a simplified DELFI process.
The key systems involved in the system are:

- **Web front End**: Internet web frontend of local or regional journey planner provided by local or regional public transport operator that allows an itinerary request.

- **Location Resolver**: component of local system that has the task to compare the user input for origin and destination with the information coded in the meta data of DELFI and to resolve the city name and to retrieve the GKZ (German city-ID) and the responsible provider for these cities.

- **Search Controller**: This controller is part of the local system and communicates with the
other DELFI Servers in order to retrieve the information. The meta data base knows which server provides the requested data. The search controller is a kind of middleware between the front end web interface and other DELFI servers, communicating with them via DELFI-API. The search controller resolves complete continuous itineraries by means of communication from other DELFI servers.

- One or more (max. three) DELFI Server(s): A DELFI Server delivers partial itineraries and the search controller puts them together to create complete itineraries and selects the optimal one. This result will be returned to the web interface to display it.

### Interface

DELFI consists of three interfaces. The interfaces are divided into the part for resolving locations and the part for the computation of connections in general. The connection part is again separated into an itinerary computation part and its derivatives for the distributed computation.

A search controller should implement the interfaces `StartDestIdentification` and `ScheduleInfo`. A client (i.e. a web front end) can access a search controller like a normal information server accesses. The client does not need to know how the information is computed.

A DELFI Server implements the interfaces `StartDestIdentification`, `ScheduleInfo` and additionally the interface `DistributedScheduleInfo`. If a search controller implements also the `DistributedScheduleInfo` a cascading of system may be possible. In real terms, the existing search controllers implement only the distributed search. The location resolution is a separated component.

<table>
<thead>
<tr>
<th>interface</th>
<th>methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>StartDestIdentification</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>getLocations</code></td>
<td>Resolving locations (stops, stations, addresses or POIs) using other systems</td>
</tr>
<tr>
<td><code>ScheduleInfo</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>getValidPeriod</code></td>
<td>Verification of validity of time table data respectively time and date interval for itinerary calculations</td>
</tr>
<tr>
<td></td>
<td><code>getConnection</code></td>
<td>Access to complete itineraries (not for distributed itinerary calculation) only if origin and destination are located in the area of one journey planning service (one server)</td>
</tr>
<tr>
<td><code>DistributedScheduleInfo</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>getAllTransitions</code></td>
<td>Request of list of all known transition points of the requested server</td>
</tr>
<tr>
<td></td>
<td><code>getTransitions</code></td>
<td>Filtered request of relevant transition points for desired itinerary request</td>
</tr>
<tr>
<td></td>
<td><code>getPartialConnections</code></td>
<td>Calculates the fastest itinerary to its destination at an origin point for a given origin and start point in time and the latest departure at an origin point for a given destination and end point in time respectively</td>
</tr>
</tbody>
</table>
The call of `GetLocations` in the LocationResolver causes the following flow:

1. Both city names (for origin and destination) will be matched with all city names in a table MetaCity of the meta data. The resulting match of every city name input has one (in case of an explicit identification) or more possible entries or an empty resulting list. If there are more than one possible matches the End user will receive a list with the different possibilities in order to choose an appropriate one or renew his request. The End user has, of course, also to renew his request if there are no results available.

2. If there is only one match, the city-ID (GKZ) of the city will be used in order to figure out the responsible provider for the city. Therefore, a table MetaRegion is to be used and the centre co-ordinates of every GKZ from the table InfoRegion. In addition, atable MetaServer provides the actual valid server access data for every provider. If the responsible provider for origin differs from the provider responsible for the destination, a distributed itinerary calculation is necessary.

3. The next step is to verify the input string by entering a stop or address name selected by the End user. The string will be sent to the responsible DELFI Server for the location. This indicates a `GetLocations` call to both of the identified servers (origin and destination) but it is possible that the request of the End user cannot be resolved. In this case, possible choices of stops or addresses will be available for the End user in order to select one of the presented choices. The chosen one will then be sent again to the local server in order to receive the appropriate data to provide a connection for the End user’s request.

Besides the determined responsible servers for origin and destination, an additional server for the long-distance transport modes is necessary. Therefore, the whole connection will be calculated by a maximum of three servers. The LocationResolver resolves and figures out the server responsible for origin and destination.

In the next step the interface `ScheduleInfo` is used. The task of this interface is to provide information about the valid time-table period for requests and the calculation of complete itineraries. The method `GetValidPeriod` can be used to get and store valid time periods internally in the search controller in order to avoid unnecessary calls, but it can also be used for watchdog components to keep information about the availability of servers up-to-date.

The flow to get a complete itinerary information consists of four single activities (forward search = departure search)

1. GetTransitions
2. GetPartialConnections (forward)
3. GetPartialConnections (backwards)
4. GetPartialRoutes

1. Both, origin and destination system, deliver all or all relevant transition points to a Get-Transition or GetAllTransitions-Request. The result of the controller is a 1 (in case only one station is relevant for the resolved location at origin), many combination (1:n connection computation case), and a m:1 combination for the same situation at the destination.

2. With the origin location and the transition points, the partial connections can be requested using the GetPartialConnections method from the origin server in case of a departure search in the first phase. In case of a departure search the result of the first (origin) server will be a set of arrival nodes (transition nodes) with time labels. The main task of the method GetPartialConnection is to determine the arrival time point at the destination node and not the complete route information towards the node.

3. These nodes are used for the next request to the intermediate long-distance transport server. It has to compute the m:n combinations for all possible connections of possible transition nodes from the origin server (m elements, those with retrieved time labels) to all transition points of the destination server (n elements). The result will also be a set of nodes with time labels (n or less).

4. The resulting nodes will be used for the third step of the first phase; the calculation of connections to the destination from all arrival nodes of the long-distance server. The resulting nodes (again those with time labels) will be used to compute connections to the destination.

5. Afterwards, phases 2 and 3 will be performed where the backward search and the forward route computation is done the same way as in the first phase.

The alternative(s) regarding date of arrival or departure can be found if the computation is repeated in a revised order (Beginning at the destination at the calculated point in time backwards). All the partial connections for the final route are sampled in the last processing step. This led to the three steps usually named as search forward, search backward, and again search forward.

In the last step GetPartialRoutes, the process will be finished by means of establishing the route information by using the latest calculated departure time point at origin to the earliest possible arrival time point at destination.

In all cases, the necessary transition times in form of mode changes and/or walks are added by the responsible servers. The complete calculated itinerary might, therefore, not be the only existing one. There can be several alternatives varying in optimization criteria. If so, it leads to a set of itineraries which have to be calculated and combined within the search controller.
Afterwards, all results are returned to the search controller. The search controller adds the partial itineraries to a continuous itinerary and presents the results on the web front end to the End user.

**GetLocations method**

The main element of a `GetLocations`-requests and responses is the element of `tagLocation` with the following data.

<table>
<thead>
<tr>
<th>Element</th>
<th>relevance</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>mandatory</td>
<td>Location key: individual identifier within the requested system, the output of the requested system, usually not referable outside the requested system</td>
</tr>
<tr>
<td>state</td>
<td>mandatory</td>
<td>state of stop identification (STATE_INITIAL (first request), STATE_INCOMPLETE (further request after verification with the customer needed), STATE_COMPLETE (stops or addresses are identified correctly))</td>
</tr>
<tr>
<td>type</td>
<td>mandatory</td>
<td>Type of stop (0 = not initialised, 1 = station or stop, 2 = address, 3 = POI)</td>
</tr>
<tr>
<td>name</td>
<td>mandatory</td>
<td>output name of the requested type, this must be the complete name</td>
</tr>
<tr>
<td>location</td>
<td>optional</td>
<td>information about Location element like top level region, additional names, Coordinates, postal codes, house numbers</td>
</tr>
<tr>
<td>side</td>
<td>optional</td>
<td>A list of additional information like maps and local descriptions</td>
</tr>
<tr>
<td>attrs</td>
<td>optional</td>
<td>More additional information, attribute text for display shown to the End user together with the itinerary i.e. “train with restaurant”</td>
</tr>
</tbody>
</table>

**GetConnections method**

The main element of the `GetConnections` methods are `tConnectionRequest` and `tConnectionReply`.

`tConnectionRequest`

<table>
<thead>
<tr>
<th>Element</th>
<th>relevance</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gains</td>
<td>optional</td>
<td>Optimising criteria for route search (fastest, cheapest, most comfortable (less transitions))</td>
</tr>
<tr>
<td>searchtime</td>
<td>mandatory</td>
<td>Requested travel time (Input time of end user)</td>
</tr>
<tr>
<td>request-Mode</td>
<td>mandatory</td>
<td>Request mode, defines whether the search time is a departure or an arrival</td>
</tr>
<tr>
<td>from</td>
<td>mandatory</td>
<td>Start location</td>
</tr>
<tr>
<td>via</td>
<td>optional</td>
<td>Via locations</td>
</tr>
<tr>
<td>to</td>
<td>mandatory</td>
<td>Destination location</td>
</tr>
<tr>
<td>products</td>
<td>optional</td>
<td>Products (travel modes) per partial itinerary</td>
</tr>
<tr>
<td>features</td>
<td>optional</td>
<td>Features (attributes) per partial itinerary</td>
</tr>
<tr>
<td>fromIT</td>
<td>optional</td>
<td>The sequence of types of Individual transport mode at origin location, each element in the sequence represents one activated individual transport mode. (pedestrian, bike, taxi, Kiss &amp; Ride, Park &amp; Ride).</td>
</tr>
<tr>
<td>toIT</td>
<td>optional</td>
<td>types of Individual transport mode at destination (see fromIT))</td>
</tr>
<tr>
<td>connectionsBefore</td>
<td>optional</td>
<td>Number of desired itineraries before requested time</td>
</tr>
</tbody>
</table>
connection-After | optional | Number of desired itineraries after requested time
interval | optional | Length of search interval in minutes
kontext | optional | Context information
languages | optional | Requested languages

The main element of `tConnectionReply` is the the element `tSchedules`

<table>
<thead>
<tr>
<th>Element</th>
<th>relevance</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>infoMsb</td>
<td>optional</td>
<td>Notices, i.e. “long stay” (values defined in the meta data)</td>
</tr>
<tr>
<td>elements</td>
<td>mandatory</td>
<td>The elements of the itinerary, ride and walk parts (see additional table <code>tElementPart</code>)</td>
</tr>
<tr>
<td>legs</td>
<td>mandatory</td>
<td>The number of rides without walks</td>
</tr>
<tr>
<td>duration</td>
<td>mandatory</td>
<td>Itinerary length in minutes (complete with walks)</td>
</tr>
<tr>
<td>dist</td>
<td>optional</td>
<td>Length of itinerary in kilometres</td>
</tr>
<tr>
<td>fares</td>
<td>optional</td>
<td>Price information if available</td>
</tr>
<tr>
<td>daysOfOperation</td>
<td>optional</td>
<td>Days of operation valid from the given itinerary; if no days of operation are given the itinerary is only valid for the requested date</td>
</tr>
<tr>
<td>sideInfo</td>
<td>optional</td>
<td>Additional information like local maps</td>
</tr>
</tbody>
</table>

`tElementPart`

<table>
<thead>
<tr>
<th>Element</th>
<th>relevance</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>providerCode</td>
<td>mandatory</td>
<td>Provider that computed the itinerary</td>
</tr>
<tr>
<td>From</td>
<td>mandatory</td>
<td>Start station</td>
</tr>
<tr>
<td>To</td>
<td>mandatory</td>
<td>Target station</td>
</tr>
<tr>
<td>DepartureTime</td>
<td>mandatory</td>
<td>Departure time from start date</td>
</tr>
<tr>
<td>DepartureDelay</td>
<td>optional</td>
<td>Delay at departure, delays must not be included in departure or arrival time, departureTime and arrivalTime contain always the point of time from the time table without any additional time offset, delays can be used for user hints</td>
</tr>
<tr>
<td>ArrivalTimes</td>
<td>mandatory</td>
<td>Arrival time and date</td>
</tr>
<tr>
<td>ArrivalDelay</td>
<td>optional</td>
<td>Arrival will be delayed, see comments at departureDelay</td>
</tr>
<tr>
<td>Line</td>
<td>mandatory</td>
<td>Information about the ride</td>
</tr>
<tr>
<td>departureTrack</td>
<td>optional</td>
<td>Additional information if leaving from a main station i.e. platform number</td>
</tr>
<tr>
<td>arrivalTrack</td>
<td>optional</td>
<td>Additional information if arriving at a main station i.e. platform numbers</td>
</tr>
<tr>
<td>attrs</td>
<td>optional</td>
<td>Attributes of the element</td>
</tr>
<tr>
<td>pass</td>
<td>optional</td>
<td>Pass locations, pass must either be empty or filled with all in-between locations including start and destination of this connection element, if filled it must have at least two elements (start, destination)</td>
</tr>
</tbody>
</table>
4.5.3 OpenTravel Alliance (OTA) Standard

The OpenTravel Alliance is a non-profit organization working to establish a common electronic vocabulary, represented in XML format, for use in the exchange of travel information. An overview of the OTA standard is available in the main document at subsection 3.5.11.

Functionality

The OTA Standard defines a Set of XML Message, to cover the whole range of requirements of the travel industry. For the air travel industry a set of 26 message types, and 14 Message Pairs (Request and Response Message) are defined. Some response messages are used for more than one request message. For Example the OTA-AirBookRS is the response message for the OTA_AirBookRQ request message and the OTA_AirBookModifyRQ request message.

Interface

Operation: OTA_AirAvailRQ/RS

This specification addresses the structure and elements of requests and responses for airline flight availability and point of sale information. The availability request message requests flight availability for a city pair on a specific date for a specific number and type of passengers. The request can also be narrowed to request availability for a specific airline, flight or booking class on a flight, all for a specific date. Optional request information may include:

- Time / Time Window
- Connecting cities
- Client Preferences (airlines, cabin, flight types etc.)

The availability response message contains flight availability for a city pair on a specific date. A set of origin and destination options is returned, each of which contains one or more (connecting) flights that serve the city pair. For each flight the following information is returned:

- Origin and destination airports
- Departure and arrival date/times
- Booking Class availability
- Equipment
- Meal Information
- Codeshare information

Operation: OTA_AirBookRQ/RS

The Book Request message requests to book a specific itinerary for one or more identified
passengers. The message contains optional pricing information, allowing the booking class availability and pricing to be rechecked as part of the booking process.

Optional request information can include:

- Seat and meal requests
- Special Service Requests (SSR), Other Service Information (OSI), Remarks
- Fulfillment information—payment, delivery details, type of ticket desired

If the booking was successful, the Book Response message contains the itinerary (including the directional indicator, status of the booking, and number of passengers), passenger and pricing information.

**Operation: OTA_AirCheckInRQ/RS**

The Checkin Request and Response message pair provides an air travel checkin product specifically for the self-service channels (kiosks, web and mobile). These messages will allow customers to checkin for eligible flights using various channels (kiosks, web and mobile) and to provide the ability to perform related functions such as checking baggage, issue bag tags, boarding passes and entering further passenger information (APIS data etc.).

**Operation: OTA_AirDemandTicketRQ/RS**

The Demand Ticketing Request and Response message pair provides an air travel ticketing product used for requesting ticket fulfillment.

**Operation: OTA_AirDisplayQueueRS**

The OTA_ReadRQ and OTA_AirDisplayQueueRS messages are used to display booking files on a queue. The booking file data can be requested: in full format whereby the booking file(s) on that queue will be returned; or in condensed format containing queue information. This data will be used for queuing and booking reference. Additionally, the items can be removed from queue (by displaying them) or left on the respective queue.

**Operation: OTA_AirFareDisplayRQ/RS**

The Fare Display Request message allows a client to request information on fares which exist between a city pair for a particular date or date range. No Inventory check for available seats on flights is performed by the server before returning the Fare Display Response message. The request can optionally contain information indicating that a more specific response is required. This information can include passenger information, specific flight information and information on the types of fares that the client is interested in.

The response to the OTA_AirFareDisplayRQ message is the OTA_AirFareDisplayRS message. This message contains repeating FareDisplayInfo elements, each of which contains information on a specific fare contract including Airline, Travel Dates, Restrictions and Pricing. It also can optionally return information on other types of fares that exist, but which have not been included in this response.
**Operation: OTA_AirFlifoRQ/RS**

The AirFlifo message is a request for updated information on the operation of a specific airline flight. The request requires the airline, flight number and departure date. The departure and arrival airport locations can be also be included.

The AirFlifo response includes realtime flight departure and arrival information. The following flight operation data is included in the response:

- Departure airport
- Arrival airport
- Marketing and operating airline names, when applicable
- Flight number
- Type of equipment
- Status of current operation
- Reason for delay or cancellation
- Airport location for diversion of flight
- Current departure and arrival date and time
- Scheduled departure and arrival date and time
- Duration of flight
- Flight mileage
- Baggage claim location

**Operation: OTA_AirDetailsRQ/RS**

The Air Details Request message requests flight leg and codeshare information for a specific flight on a specific date between a city pair.

The Air Details Response message contains airline, arrival and departure times, equipment, meal, and duration information (total and ground) for each leg of a flight. It also contains codeshare information, “on time” percentage, and electronic ticketing eligibility.

**Operation: OTA_AirLowFareSearchRQ/RS**

The Low Fare Search Request message requests priced itinerary options for flights between specific city pairs on certain dates for specific numbers and types of passengers. Optional request information can include:

- Time / Time Window
- Connecting cities.
- Client Preferences (airlines, cabin, flight types etc.)
- Flight type (nonstop or direct)
• Number of itinerary options desired

The Low Fare Search Response message contains a number of ‘Priced Itinerary’ options. Each includes:
• A set of available flights matching the client’s request.
• Pricing information including taxes and full fare breakdown for each passenger type
• Ticketing information—ticket advisory information and ticketing time limits.
Fare basis codes and the information necessary to make a rules entry.

Operation: OTA_AirPriceRQ/RS
The Air Price Request message requests pricing information for specific flights on certain dates for a specific number and type of passengers. The message allows for optional information such as fare restriction preferences and negotiated fare contract codes to be included.
The pricing request contains the information necessary to perform an availability/sell from availability/price series of entries on an airline CRS or GDS.
The Pricing Response message contains a ‘Priced Itinerary’. This includes:
• The set of flights sent in the Pricing request message
• Pricing information including taxes and full fare breakdown for each passenger type
• Ticketing information
• Fare Basis Codes and the information necessary to make a Fare Rules entry.

Operation: OTA_AirRulesRQ/RS
The Rules Request message requests text rules for a specific fare basis code for an airline and city pair on a specific date. Optional information allows negotiated fare contract codes to be included in the request.
The Rules Response message contains a set of text (human readable) rule information paragraphs. Each paragraph is identified by a rule code.

Operation: OTA_AirSchedulesRQ/RS
The Air Schedules message set is intended to provide a customer or a third party entity with the ability to view airline flight schedules. This message set requires the customer to specify the Departure and Arrival cities for a specific date. It offers flight information on airlines that provide service between the requested cities.
The Schedules messages could be used for the following circumstances:
• Customer may want to determine what airlines offer service to/from specific cities.
• Customer is looking for a specific flight number. By entering the arrival and departure cities and knowing the approximate arrival or departure time the customer can locate their specific flight number.

• Customer needs to determine the days of the week that service is scheduled to and from the requested destination.

• Customer will be able to determine the type of aircraft used to fly that route. Some customers prefer to fly on larger types of aircraft.

The schedules message also contains other information that customers are interested in: meal service, duration of flight, on-time statistics and if smoking is allowed. In addition, these messages provide the foundation for electronic timetables.

**Operation: OTA_AirSeatMapRQ/RS**

Seat maps display which seats are available for a given flight, as well as their location within the aircraft. To make a seat assignment, when using an online booking product, a customer will frequently access a seat map display to determine which seats are available. Then they will make a separate seating request. These messages identify all the information necessary to request and return an available seat map for a particular flight.

Types of information for the seat map request include: airline, flight number, date of travel, class of service and frequent flier status. The response message includes: flight, aircraft and seat description information.

**Operation: OTA_AirBookModifyRQ/OTA_AirBookRS**

The Book Modify Request message requests to modify an existing booking file. The message contains all elements of OTA_AirBookRQ plus a general type of modification, i.e. name change, split, cancel or other is indicated with the attribute ‘ModificationType’.

The modification operation on the different elements is either indicated with the existing attribute ‘Status’ (for air segments, SSR’s and seat requests) or with attribute ‘Operation’ of Type ActionType for the other elements (i.e. Other service information, remarks, AirTraveler Elements).

In the AirBookModifyRQ element all the data to be changed is submitted and in the AirReservation element all existing data may be submitted. This allows the receiving system to perform a consistency check before updating the booking file. In order to keep the message small, this can be omitted.

### 4.5.4 Ferry XML

The Ferry XML standard was developed by the Travel Technology initiative (TTI). Ferry XML was created by providing a XML structured alternative to the existing EDI Unicorn standard for booking ferry reservations. An overview of the standard is available in the main document at subsection 3.5.3.
**Functionality**

The TTI Ferry XML messages cover the business functions of timetable and sailing queries, quotation, booking, recall, amendment and cancellation. They can be used independently of each other in non-sessioned (or stateless) transactions where the awarding system keeps no record of preceding messages. Alternatively they can be used as part of an ongoing transaction in a sessioned environment where the awarding system retains information from preceding messages.

A typical booking process will consist of a series of related messages. For example a client may perform a Timetable request then an Availability check, which is followed by a Booking. Some reservation systems will process these messages in a stateless or unsessioned manner whereby each message is self-contained and is processed without reference to any previous message. This means that the requesting system is obliged to repeat in a later request message any information that may have been passed in an earlier one. The benefit of this type of conversation is that a requesting system may instigate an enquiry at any stage of the booking process and could, if required, make a confirmed booking with a single request.

Additionally, in unsessioned processing, each response message will contain the information supplied in the original request meaning that the requesting system can process a response even if it no longer has the original request.

The following picture shows the Message flow of the Ferry XML Standard.
TTI recommends the use of HTTP POST as the transport protocol for handling Ferry XML messages.

The prior recommendations (from the OTA) regarding ebXML are still available for companies wishing to implement that transport protocol. However the protocol based on a simple
HTTP POST represents a simpler method to create interoperable systems. Future OTA releases may include SOAP (Simple Object Access Protocol).

Ferry XML uses Standard HTTP, which accommodates extensions through the use of custom headers defined for OTA transactions.

Encrypted communication between systems is accomplished with SSL (i.e. HTTPS). If encryption is required, a system MUST support HTTPS. Authentication of a client (requestor) to a server (responder) is achieved via HTTP Basic Authentication. Authentication of a server (responder) to a client (requestor) is achieved via SSL.

Following XML Message pairs define the operations of the Ferry XML interface:

<table>
<thead>
<tr>
<th>Name</th>
<th>with function code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTI_FerryAvailRQ</td>
<td>- Timetable indicator set “true” or “false”</td>
<td>Request a list of possible sailings or availability of space on specific sailings.</td>
</tr>
<tr>
<td>TTI_FerryAvailRS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTI_FerryBookRQ</td>
<td>- Book or Quote</td>
<td>Request a firm booking or a quotation.</td>
</tr>
<tr>
<td>TTI_FerryBookRS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTI_FerryAmendRQ</td>
<td>- Amend or Amend Quote</td>
<td>Request amendment to a booking or a quotation for an amendment.</td>
</tr>
<tr>
<td>TTI_FerryAmendRS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTI_CommitRQ</td>
<td>(after Quote or Amend Quote) - Commit</td>
<td>Commit to a booking or amendment after a quotation.</td>
</tr>
<tr>
<td>TTI_CommitRS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTI_CancelRQ</td>
<td>- Cancel or Cancel Quote</td>
<td>Request cancellation of a booking or quotation.</td>
</tr>
<tr>
<td>TTI_CancelRS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTI_RecallRQ</td>
<td>- Recall</td>
<td>Retrieve the details of a booking.</td>
</tr>
<tr>
<td>TTI_RecallRS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OTA_PingRQ</td>
<td></td>
<td>Check to confirm connection is active.</td>
</tr>
<tr>
<td>OTA_PingRS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 4.6 Positioning Service

A Positioning Service provides location information to the user of the service.

The location can be determined at the client side by means of GNSS (GPS, GALILEO) or by various server side techniques.

#### 4.6.1 OGC OpenLS - Core Services, Part 2 Gateway Service

The OGC OpenLS standard defines Location Based Services for a “GeoMobility Server” platform. It defines the interfaces for a couple of core service types, namely Directory Service, Gateway Service, Location Utility Service, Presentation Service, and Route Service. An overview of OGC OpenLS is available in the main document at subsection 3.4.7.

This description refers to Part 2 of this standard, which is named “Gateway Service”. The
Gateway Service is a network-accessible service that provides the location of one or more mobile terminals.

**Functionality**
The Gateway Service is employed to obtain the position of the client’s mobile terminal from the network.

**Interface**
The OpenLS Gateway Service specification defines an interface for obtaining the current position of a mobile user. It is defined as a pair of XML request and response documents, which can be transmitted by means of various distributed network architectures, among these (being a typical web service) HTTP and the POST method.

**Operation: Gateway Service**
The Gateway Service operation request (named SLIR) defines criteria for the retrieval of the current position. The service answers with a response (called SLIA) containing the retrieved location.

A Gateway Service operation request contains the following parts:

- Details like spatial reference system or priority,
- Identification of the mobile user, encryption options,
- Requested quality of position, like accuracy, possible delays.

The response identifies the mobile user and returns the position.

4.7 **Directory Service**

A Directory Service provides access to an online directory (e.g. Yellow Pages) to find the location of either a specific directory content or one which is closest to a given position.

Directory Services serve POI data. In this sense they are also specialised Data Services and might therefore be defined as profiles of Data Services.

4.7.1 **Directory Services Standards (DSML) v2.0**

DSMLv2 provides a method for expressing directory queries and updates (and the results of these operations) as XML documents. An overview of DSML is available in the main document at subsection 3.4.3.

**Functionality**

DSMLv2 focuses on extending the reach of LDAP directories. DSMLv2 represents the operations that an LDAP directory can perform and the results of such operations.
DSMLv2 is defined in terms of a set of XML fragments that are used as payloads in a binding. A binding defines how the DSMLv2 XML fragments are sent as requests and responses in the context of a specific transport such as SOAP, SMTP, or a simple data file.

DSMLv2 defines two normative bindings:

- SOAP request/response (Simple Object Access Protocol)
- File binding that serves as the DSMLv2 analog of LDIF (Lightweight Directory Interchange Format)

DSMLv2 document can be transported via a variety of mechanisms, so the document itself is not used to authenticate the requestor. DSMLv2 includes an Auth request that may be used to associate a security principal with a collection of DSMLv2 operations.

DSMLv2 also allows multiple LDAP operations to be expressed in one request document, and by specifying a simple positional correspondence between individual requests within a request document and individual responses within a response document.

**Interface**

There are two types of DSMLv2 document - the request document and the response document. In a DSMLv2-based interaction between a client and a server there is a pairing of requests and responses where for each request document submitted by the client there is one response document produced by the server. The structure of the request and response documents depends on the specification of a binding of DSMLv2 to some underlying protocol.

The client and server associate an individual response in a BatchResponse with the corresponding individual request in a BatchRequest using one (or both) of the following methods:

- positional correspondence
- RequestID.

In a positional correspondence, the n\(^{th}\) response element corresponds to the n\(^{th}\) request element. For instance, if the third response element is a delete response, then it corresponds to the third request element, a delete request.

The alternative to positional correspondence is the use of the optional requestID attribute. When the client specifies a value for requestID in a request (for example, in an addRequest) the server must return the same value in the corresponding response (for example, in an addResponse). The client need not specify a requestID when positional correspondence is also used, although in some cases it may find this useful. For example, when using the file binding for a large file, some clients may find it more convenient to associate failed responses with requests using requestID rather than position.
4.7.2 OGC OpenLS - Core Services, Part 1 Directory Service

The OGC OpenLS standard defines Location Based Services for a “GeoMobility Server” platform. It defines the interfaces for a couple of core service types, namely Directory Service, Gateway Service, Location Utility Service, Presentation Service, and Route Service. An overview of OGC OpenLS is available in the main document at subsection 3.4.7.

This description refers to Part 1 of this standard, which is named “Directory Service”. The Directory Service is a network-accessible service that provides access to an online directory (e.g., Yellow Pages) to find the location of a specific or nearest place, product or service.

Functionality

This service provides clients with access to an online directory to find the nearest or a specific place, product or service. The client starts to formulate the search parameters in the service request, identifying the place, product or service that is sought by entering the name, type, category, keyword, phone number, or some other ‘user-friendly’ identifier. A position must also be employed in the request when the client is seeking the nearest place, product or service, or if they desire a place, product or service at a specific location or within a specific area.

Given the formulated request, the Directory Service searches the appropriate online directory to fulfill the request, finding the nearest or specific place, product or service, depending on the search criteria. The service returns one or more responses to the query (with locations and complete descriptions of the place, product, or service, depending upon directory content), where the responses are in ranked order based upon the search criteria.

Interface

The OpenLS Directory Service specification defines an interface for accessing and retrieving directory information. It is defined as a pair of XML request and response documents, which can be transmitted by means of various distributed network architectures, among these (then being a typical web service) HTTP and the POST method.

Operation: DirectoryRequest

A DirectoryRequest defines criteria for the retrieval of directory information. The service answers with a DirectoryResponse containing the retrieved POIs together with their distance from a reference point.

A DirectoryRequest contains two principal parts, which control:

- the location where the search has to occur and the mode of retrieval in that location, and;
- additional selection criteria concerning the search.

The location can principally be specified in four ways:

- Address: A POI has to be pin-pointed at an address location reference.
• **Nearest**: The POIs nearest a Location according to criteria like Proximity, Fastest, Shortest, Easiest shall be found.

• **WithinDistance**: POIs within a given distance from a location are searched.

• **WithinBoundary**: POIs in a given area are searched.

Note that the Location specified in these requests is a generalised concept as specified in 2.4.5, which can also be a Street Address or another POI.

Selection criteria are given specifying a directory type (e.g., yellow pages) together with property descriptions out of a predefined and extensible namespace.

The response is a list of POIs as described in 2.9.2 together with the distance to the reference point if given.

### 4.8 Geocoding Service

Geocoding services translate different forms of geo-referencing into each other, mainly street addresses into coordinates. The reverse process is often called reverse geocoding.

#### 4.8.1 OGC OpenLS - Core Services, Part 3 Location Utility Service

The OGC OpenLS standard defines Location Based Services for a “GeoMobility Server” platform. It defines the interfaces for a couple of core service types, namely Directory Service, Gateway Service, Location Utility Service, Presentation Service, and Route Service. An overview of OGC OpenLS is available in the main document at subsection 3.4.7.

This description refers to Part 3 of this standard, which is named “Location Utility Service (Geocode/Reverse Geocode)”. The standards defined in the document describe network-accessible services that can determine a geographic position given a place name, street address or postal code. The service called reverse geocode creates a normalised place name, street address and postal code from a position.

**Functionality**

This service performs as a Geocode by determining a geographic position, given a place name, street address or postal code. It also returns a complete, normalised description of the place (which is useful, say, when only partial information is known). The service also performs a reverse Geocode by determining a complete, normalised place/street address/postal code, given a geographic position. Both the Geocode and reverse Geocode may return zero, one, or more responses to a service request, depending on the request information of the client, the algorithm being employed, and the match criteria.

**Interface**

The OpenLS Geocode and Reverse Geocode Service specification defines an interface for translating between addresses and positions. It is defined as pairs of XML request and response documents, which can be transmitted by means of various distributed network archi-
tectures, among these (then being a typical web service) HTTP and the POST method.

**Operation: Geocode**

Input into a Geocode request is a collection of addresses as defined in 2.4.5. The Geocode response primarily consists of one output for each of the addresses supplied in the request. Each of these outputs (corresponding to one input address) is a collection of data structures, containing:

- A point
- A normalised address
- QoS information

The output for one input address is delivered ordered by accuracy, highest accuracy first.

**Operation: Reverse Geocode**

The Reverse Geocode request consists of

- A Position as described in 2.4.5,
- A preference indicator for the type of response. Among these are StreetAddress, IntersectionAddress, Place Name, etc.

The response is a collection of structures, each consisting of

- Address,
- Point,
- Distance to search centre

The collection is sorted ascendingly as to the distance.

### 4.8.2 OGC Web Feature Service (WFS) - Gazetteer Service Profile

The Gazetteer Service is defined as a profile of the OGC Web Feature Service Specification. It allows a client to search and retrieve elements of a georeferenced vocabulary of well-known place-names. An overview of the standard profile is available in the main document at subsection 3.4.2.

**Functionality**

The Gazetteer Service (WFS-G) is a specialised Web Feature Service that provides additional capabilities specific to a gazetteer-like feature collection. Instances within a collection of gazetteer features may be – as the terms of a thesaurus – related to each other and constitute a hierarchical vocabulary of geographical places. The overall database structure is implemented as a GML application schema which defines a general abstract feature type (named SI_LocationInstance) to be served by the WFS-G.
Interface
The interface is like WFS basic, see 4.2.1.
Only the operations GetCapabilities, DescribeFeatureType and GetFeature need to be implemented.

4.9 Coordinate Transformation Service

Coordinate Transformation Services provide an interface to coordinate systems metadata and perform transformations between different coordinate systems.

4.9.1 OGC Web Coordinate Transformation Service (WCTS)

The OGC Web Coordinate Transformation Service specifies an interface, which can be used by geospatial applications and other services. An overview of the standard is available in the main document at subsection 3.5.10.

Functionality

The specified service transforms digital geospatial data from one Coordinate Reference System (CRS) to another. The geospatial data transformed is digital feature data, including digital coverages. Such transformations include all the types of coordinate operations, including both coordinate “transformations” and “conversions”. This service inputs digital features in one CRS and outputs the same features in a different CRS. The service inputs include the identifications of the input and output CRSs, and optionally the coordinate transformation between these CRSs.

Interface

The WCTS is an “OGC Web Service” type of interface, where client and server software interact using HTTP messages. This interface thus has similarities to the Web Map Service (WMS), Web Feature Service (WFS), and Web Coverage Service (WCS) interfaces.

This WCTS interface specifies seven operations that can be requested by a client and performed by a WCTS server. Those operations are:

Operation: GetCapabilities

Implementations need to provide this operation. The operation allows a client to request and receive back service metadata (or Capabilities) documents that describe the abilities of the specific server implementation.

Operation: Transform

This operation allows a client to request transformation of the coordinates in a specified set of features, and to receive back the transformed features. This operation allows the client to identify the CRS of the input features and the desired CRS of the output features. Each feature can be a single feature, a feature collection containing multiple features, or a coverage. This is the main operation and is (of course) mandatory.
Operation: IsTransformable
This optional operation allows a client to ask a server if it can perform any coordinate transformation from one identified CRS into another. This operation also allows the client to ask if this transformation can be performed on specified geometric primitive or coverage types. This transformation can be a single transformation, or a ConcatenatedOperation which combines multiple single transformations. The response back is binary, plus problem information when the answer is no.

Operation: GetTransformation
This operation allows a client to request and receive back the definition of the server-known coordinate transformation(s) from one identified CRS into another. Each transformation can be a single transformation, or a ConcatenatedOperation that combines multiple single transformations. Each transformation definition is encoded in XML using the GML 3.1.1 CRS Schemas. This is an optional operation.

Operation: DescribeTransformation
This operation allows a client to request and receive back the definition of one or more identified coordinate transformations. Each of these transformations can be a single transformation, or a ConcatenatedOperation that combines multiple single transformations. Each transformation definition is encoded in XML using the GML 3.1.1 CRS Schemas.

Operation: DescribeCRS
This operation allows a client to request and receive back the definition of one or more identified coordinate reference systems (CRSs). Each of these CRSs can be any one of the nine different concrete types of CRS. Each CRS definition is encoded in XML using the GML 3.1.1 CRS Schemas. The implementation of this operation is optional.

Operation: DescribeMethod
This optional operation allows a client to request and receive back the definition of one or more identified operation methods. Each operation method definition is encoded in XML using a specified GML Application Schema that builds on the GML 3.1.1 CRS Schemas.

4.10 Registry / Catalogue Service
Registry Services (Catalogue Services) cover the interactions of publishing and finding service and data offerings. They allow metadata to be published (registered) and retrieved, such as service descriptions, data descriptions, data specifications, application schemas, code-lists, thesauri, coordinate reference systems, portrayal rules, symbology, and contractual information.

 Though often used interchangeably, the terms ‘catalogue’ and ‘registry’ are sometimes understood in a sense that a registry is a specialised catalogue, which is associated with a formal registration process for users and a strictly controlled life-time management of registered material (as for example described in ISO 19135). Among the standards described in this chapter only the ebRS standard (see 4.10.1) is a ‘registry’ in this strict sense.
Publishing and retrieving follows the well-known *publish-find-bind* pattern of SOAs.

Registry Services cover the interactions of *publishing* and *finding* by offering a platform to the possible service provider to publish his service and to the possible service requestor to search the publications and find a suitable provider for his request. They can also prepare the *binding* step by providing the necessary metadata describing how to access the resource found.

The finding step is the central point. It can be thought of as consisting of three logical steps:

- **Discovery**: The action of determining candidates
- **Evaluation**: The action of checking whether the candidates found are suited for the purpose. This part is hard to perform automatically.
- **Use**: The action of being provided with the necessary metadata to bind.

Discovery can be done by Registries, however, in a quite limited way. The use of Search Engines to find resources and registries may be appropriate if discovery on a wide scale is necessary.

One more word regarding the future use of the *Semantic Web* and *Web Ontology Language (OWL)* metadata [W3C-OWL, 2004] in or in place of specialized registries:

The essential points are again the finding step and also the evaluation step. By means of ontologies resources might be described in a machine readable way permitting automated clients to discover and also evaluate metadata, because semantic facts and relations are represented. During the discovery and evaluation process, facts about the resources may be consulted to find close matches to the required properties.

Due to the lack of a published OWL database and of suitable inference engines (which are a research topic) semantic web issues are currently beyond the current realistic scope of eMOTION. A detailed discussion has therefore been left out in this documentation.

### 4.10.1 OASIS ebXML Registry Services and Protocols

An ebXML Registry is an information system that securely manages any content type and the standardised metadata that describes it. An overview of the standard is available in the main document at subsection 4.2.5.

**Functionality**

A Registry is both a registry of metadata and a repository of content. A typical Registry implementation uses some form of persistent store such as a database to store its metadata and content. Architecturally, the registry is distinct from the repository. However, all access to the registry as well as repository is through the operations defined by the Registry Service interfaces.

The ebRIM information model is a general and flexible one with several extensibility points. A cohesive set of extensions that address the needs of a particular application domain or
community of practice can be declared as parts of an extension package. Particularly, the various classification schemes defined in the registry data model are extensible.

The ebRIM defines an ExtrinsicObject type as a sort of proxy for an internally (or externally) managed repository item that conforms to some well-known content type, as indicated by the value of the mimeType attribute, which must correspond to a registered MIME media type.

Arbitrary relationships among catalogued items can be expressed by creating links between any two resource descriptions. For example, a service offer may be associated with descriptions of the data sets that can be acquired using the service.

Repository items and many RegistryObjects underly version control. This feature is based upon DeltaV.

A ebXML registry may have the capability to notify clients (or other registries) when certain content-based events occur. Registration for notification is performed by submitting an instance of a Subscription object to the registry.

An ebRIM Registry is required to perform authentication of clients either by itself (on the basis of the SOAP protocol) or by relying on an external authentication authority. In the latter case the registry has to support a defined SAML profile. On the basis of authentication the registry has to provide authorisation and access control according to mechanisms required by the standard.

All significant operations on the registry are auditable – an audit trail is automatically written.

**Interface**

The ebXML Registry consists of the following service interfaces:

- A LifecycleManager interface that provides a collection of operations for end-to-end lifecycle management of metadata and content within the registry. This includes publishing, update, approval and deletion of metadata and content.
- A QueryManager interface that provides a collection of operations for the discovery and retrieval of metadata and content within the registry.

The specification defines the following concrete protocol bindings for the abstract service interfaces of the ebXML Registry:

- SOAP Binding that allows a registry client to access the registry using SOAP 1.1 with Attachments (SwA). There is a binding of the abstract registry service interfaces to the SOAP protocol in WSDL format.
- HTTP Binding using the GET method that allows a client using a restricted part of the QueryManager interface using HTTP 1.1.

**Operation: LifeCycleManager::SubmitObjects**

A list of RegistryObjects and/or repository items is submitted to the registry. The objects may be new, in which case they are added to the registry, or they may already exist, in which case they are updated. Object ids may be specified along with the submitted RegistryOb-
jects, if they are not specified the registry will generate them. The content of the submitted RegistryObjects may reference existing objects in the registry or objects in the submitted list.

**Operation: LifeCycleManager::UpdateObjects**

A list of already existing RegistryObjects and/or repository items is updated in the registry. All attributes are replaced and only references to objects, which already exist in the registry are allowed. If repository items are not contained along with RegistryObjects, the old repository items remain in use.

**Operation: LifeCycleManager::ApproveObjects**

This operation approves one or more RegistryObjects in the registry. The set of RegistryObjects to be approved can be specified by an explicit list of references and/or by an AdhocQuery.

**Operation: LifeCycleManager::DeprecateObjects**

This operation deprecates one or more RegistryObjects in the registry. The set of RegistryObjects to be deprecated can be specified by an explicit list of references and/or by an AdhocQuery. Once an object is deprecated no new references to that object may be submitted.

**Operation: LifeCycleManager::UndeprecateObjects**

This operation undoes the deprecation of one or more previously deprecated RegistryObject instances. When a RegistryObject is undeprecated it goes back to the Submitted status and new references to that object can now again be submitted. The set of RegistryObjects to be undeprecated can be specified by an explicit list of references and/or by an AdhocQuery.

**Operation: LifeCycleManager::RemoveObjects**

The RemoveObjects operation is used to remove one or more existing RegistryObjects and/or repository items from the registry. The set of RegistryObjects to be removed can be specified by an explicit list of references and/or by an AdhocQuery. It can be chosen to delete only the repository item or both a RegistryObject and the associated repository item. Deletion of a RegistryObject requires that all references to that object have been removed beforehand.

**Operation: QueryManager::AdhocQuery**

The AdhocQuery operation of the QueryManager service interface allows a client to query the registry and retrieve RegistryObjects and/or repository items that match the specified query. The retrieved objects and items are returned to the requesting client in the AdhocQueryResponse.

The query itself is specified as part of the AdhocQuery operation either by Filter elements, a special XML language which is part of the Registry specification, or by means of a subset of SQL. Filter elements allow comparisons between object properties and constants to be nested with logical operations.

It is also possible to invoke stored queries. Queries can be stored as AdhocQuery objects
and can contain substitution parameters.
Additional options control federated queries stretching over more than one registry, allowing
the query result to be obtained in iterations of limited numbers of objects and give control
over format and content of the result.

4.10.2 OGC Catalogue Services Specification
The OGC Catalogue Services Specification specifies the interfaces, bindings, and a frame-
work for defining application profiles required to publish and access digital catalogues of
metadata for geospatial data, services, and related resource information. An overview of the
standard is available in the main document at subsection 4.2.2.

Functionality
Metadata act as generalised properties that can be queried and returned through catalogue
services for resource evaluation and, in many cases, invocation or retrieval of the referenced
resource. Catalogue services support the use of one of several identified query languages to
find and return results using well-known content models (metadata schemas) and encodings.
The Dublin Core metadata schema is recommended as a common core of all application
profiles being derived from CAT 2.0.2. The document specifies Catalogue Services through
the presentation of abstract and implementation-specific models. It subsequently binds the
abstract models to different distributed computing environments including HTTP - in this
binding the service is dubbed Catalogue Services for the Web (CSW).

Interface
We only consider the interface of the CAT binding on HTTP (called CSW 2.0.2). It estab-
ishes CAT as a web service.
Besides the typical GET and POST bindings, there is also a binding for SOAP.

Operation: GetCapabilities
The mandatory GetCapabilities operation allows CSW clients to retrieve service metadata
from a CSW server. The response is an XML document containing metadata about the ser-
vice, its provider, its invocation, its contents, and filtering capabilities. The contents part of
the response is specified in application profiles of the service.

Operation: DescribeRecord
This mandatory operation allows a client to discover elements of the information model sup-
ported by the target catalogue service. The operation allows some or all of the information
model to be described. The request lists type names from the servers information model, the
response contains descriptive entities concerning these type names, usually expressed in
XML Schema.

Operation: GetDomain
The optional GetDomain operation is used to obtain runtime information about the range of
values of a metadata record element or request parameter. The runtime range of values for a
property or request parameter is typically much smaller than the value space for that property
or parameter based on its static type definition.

This type of runtime information about the range of values of a property or request parameter
is useful for generating user interfaces with meaningful pick lists or for generating query
predicates that have a higher chance of actually identifying a result set.

**Operation: GetRecords**

This mandatory operation is the primary means of resource discovery in CSW. One or more
catalogue records from the information model can be queried from a directly addressed
server or a distributed set-up of such servers. Selection is performed by a query syntax (CQL
or Filter Encoding). The response is either synchronous or asynchronous (via a call-back).

**Operation: GetRecordById**

The mandatory operation request retrieves the default representation of catalogue records
using their identifier. This operation presumes that a previous query has been performed in
order to obtain the identifiers that may be used with this operation. The operation is also a
subset of the GetRecords operation, and is included as a convenient short form for retrieving
and linking to records in a catalogue.

**Operation: Transaction**

The optional Transaction operation defines an interface for creating, modifying and deleting
catalogue records. The specific payload being manipulated must be defined in a profile.

**Operation: Harvest**

Harvest is similar to Transaction insofar it also serves for adding metadata records to a
CSW. Whereas Transaction explicitly "pushes" data into the catalogue the optional Harvest
operation "pulls" the data into the catalogue from elsewhere. That is, the operation only re-
ferences the data to be inserted or updated in the catalogue, and it is the job of the catalogue
service to resolve the reference, fetch that data, and process it into the catalogue.

### 4.10.3 OGC Catalogue Services - ebRIM Profile of CSW

The ebXML Registry Information Model (ebRIM) Profile is an application profile of the Cata-
logue Services for the Web (CSW) (see 4.10.2). An overview of the standard is available in
the main document at subsection 4.2.3.

**Functionality**

The information model of the profile is based on the OASIS ebXML Registry Information
Model (ebRIM), version 3.0. This logical model specifies how catalogue content is structured
and interrelated; it constitutes a public schema for discovery and publication purposes.

The mapping of the ebRIM concepts on the model of CAT 2.0.2 (CSW) is basically as fol-
lows: An instance of ebRIM RegistryObject (or one of its subtypes) may be included in a
CSW message in any context where a record representation is allowed; for example, as a
child of the csw:SearchResults element in a GetRecordsResponse message. The ebRIM representations are provided if the value of the outputSchema attribute specified in the GetRecords request matches the namespace of the ebRIM schema. The ebRIM registry object representation must be used in all insert and update statements appearing within a csw:Transaction request.

The application profile is intended to provide a flexible, general-purpose catalogue service that can be adapted to meet the needs of diverse communities of practice within the geospatial domain. The principal means of customizing the behaviour and content of a catalogue service is by defining an extension package to exploit the extensibility points offered by ebRIM that enable it to be tailored for specific purposes.

**Interface**

The interface is built on HTTP/1.1 using the GET and POST methods.

The interface is as defined for CAT 2.0.2 / CSW, see 4.10.2.

There is one operation in addition to the CSW interface:

**Operation: GetRepositoryItem**

This required operation requests the repository item for some extrinsic object.

### 4.10.4 OGC Catalogue Services - ISO19115/ISO19119 Profile of CSW

The ISO 19115/ISO 19119 Profile is an application profile of the Catalogue Services for the Web (CSW) (see 4.10.2). An overview of the standard is available in the main document at subsection 4.2.4.

**Functionality**

The information model of the profile is based on the international standard for metadata description ISO 19115 (see 2.15.1), which specifies a general purpose model for metadata descriptions. In addition, the catalogue uses a metadata description for service metadata based on the ISO 19119 standard to facilitate the management of service metadata. The encoding of any information object in this profile is based on ISO/TS 19139.

The main purpose of the information model is to provide a formal structure for the description of information resources that can be managed by a catalogue service that complies with the application profile.

A metadata repository managed by a catalogue implementing this application profile deals with metadata about geospatial data, geospatial services and geospatial applications. It is intentional that the profile specified does not attempt to specify a general-purpose catalogue. Rather, it allows the retrieval and management of the metadata objects referred to above.

The application profile has no specific disciplinary focus. All communities working with these sorts of geospatial information are addressed. Typical communities are surveying, environ-
ment, geology, landscaping, water management, power industry, telecommunications etc.
The intention is to implement a generally understood information model based on standard metadata with only a few relationships among the catalogue items. Usage should be as simple as possible, implementing a set of use cases typical in the geospatial community.
The profile allows for a catalogue to accept a request from a client and distribute the request to one or more other catalogues within a federation. In this case, the metadata entries managed by the other catalogues become available to their own clients. It is possible to start a search from only one known location and to search as many catalogues as possible with the same set of attributes.
The functionality does not support life-time control of the registered object.
The storage of associated repository items in the catalogue is not possible. It is only possible to refer to external resources by reference.

Interface
The interface is built on HTTP/1.1 using the GET and POST methods.
The interface is as defined for CAT 2.0.2 / CSW, see 4.10.2.

4.11 Event Notification Service
Event Notification services are required to realise PUSH-type services. They notify a user or trigger a remote service as soon as some condition occurs. The user or service has to subscribe to such a service and is called back on arrival of the condition. For calling back a user various channels have to be utilised, like email, SMS, telephone, etc. This is because end users usually are not generally web accessible.
Please note that in this documentation, domain specific services like DATEX 2 and SIRI have been treated along with “data services” (see 4.2 Data Service) because the delivery of data is the main task of these services. Being complete domain specific services, fulfilling well-defined tasks, these services, however, also contain functionality which in our taxonomy would better appear in here, because both services offer interfaces for registration and event-driven call-back.

4.11.1 OASIS Web Services Notification (WSN)
The purpose of the Web Services Notification (WSN) is a set of specifications that standardise the way Web services interact using notifications or events. With this the implementation of event-driven interaction patterns should be provided. An overview of the standard is available in the main document at subsection 4.1.7.

Functionality
A Notification is an artifact of some occurrence known to one party and of potential interest to
third parties. In the context of the WSN a notification is represented as an XML element with a Namespace qualified QName and a type defined using XML schema.

The relationship between the party which is able to produce notifications and the party which wants to consume these notifications is called Subscription, that means someone has to subscribe for notification message exchange between the consumer and the producer.

There are different roles in the context of the WSN:

- **NotificationProducer**
  A NotificationProducer is capable of producing notifications based on situations that occur and on the parameters supplied with the requests from which subscriptions were created. It is a web service that implements the message exchanges associated with the NotificationProducer interface.

- **NotificationConsumer**
  A NotificationConsumer is a web service designated to receive notifications produced by a NotificationProducer as a result of a subscription.

- **SubscriptionManager**
  A SubscriptionManager provides operations that allow a service requestor to query and manipulate subscription resources that it manages. It is a web service that implements message exchanges associated with the SubscriptionManager interface. The SubscriptionManager may, but must not necessarily, be implemented by the NotificationProducer service provider.

- **Subscriber**
  A Subscriber is any entity that sends the SubscribeRequest message to a NotificationProducer. It may be a different entity from the NotificationConsumer for which notifications are actually produced.

Notifications can be associated with topics. The Topics mechanism provides a convenient means by which subscribers can reason about notifications of interest. As part of the publication of a notification, a publisher may associate it with one or more topics. These topics can be used as a filter for creating a subscription (e.g. associating the subscription to a certain topic). Also the NotificationProducer delivers a notification if the set of topics associated with the subscription has a non-empty intersection with the set of topics associated with the notification.

Every topic is assigned to an XML namespace, called Topic Namespace. With this it is possible to avoid naming collisions, and to facilitate interoperability between independently developed NotificationProducers and Subscribers. The WS-Topics XML Schema contains element and type definitions used to create topic namespace documents. These contain XML representations of these topics.

**Interface**

The WSN is defined independent of transport protocol details, in fact the specification can be used over a variety of different transports. For example it is possible to use a SOAP encod-
ing in conjunction with HTTP GET and/or HTTP POST.

There are different interfaces, assigned to the different roles, which are defined by the WSN with XML schema. The most important interfaces are described below:

**NOTIFICATIONCONSUMER INTERFACE**

A NotificationConsumer can receive a notification in two forms: first as a "raw" notification (i.e. the application specific content), or as a well specified Notify message. A Notify message can supply additional information in contrast to the simple notification. In which form a notification should be sent is decided by the subscriber.

**NOTIFICATIONPRODUCER INTERFACE**

The NotificationProducer has the possibility to support the required message exchanges defined in the OASIS Web Services Resource Properties Standard (the NotificationProducer Resource Properties). With this a customer of a service can get information about some resource properties of a NotificationProducer, e.g. the set of topics the NotificationProducer supports. Furthermore the NotificationProducer must be able to handle a Subscribe request message, that means to receive subscriptions of notifications, to process a subscription and to response in an appropriate way (send a SubscribeResponse or a fault).

Relevant parts of a Subscribe message:

- **ConsumerReference**
  The ConsumerReference contains an endpoint reference element (a Web Service Address). This component should provide all information items which are necessary to specify how the NotificationProducer should send notifications to the NotificationConsumer. In advance the NotificationProducer must specify the information items which should be present in a ConsumerReference, e.g. via WSDL.

- **Filter**
  A Filter enables the subscriber to express the subset of notifications that the NotificationConsumer should receive, e.g. filter elements can be topic expressions.

- **TerminationTime**
  This defines a termination time of the subscription being created.

**SUBSCRIPTIONMANAGER INTERFACE**

The SubscriptionManager can handle organisational tasks regarding to existing subscriptions. There are two styles of a SubscriptionManager interface: base, which must be implemented by all SubscriptionManagers, and pausable, which allows subscriptions to be suspended and resumed. In the following only the messages of the base interface are described.

It is possible to modify the current lifetime of a subscription with the **Renew** command. Therefore the SubscriptionManager must handle the appropriate request and must respond with a RenewResponse message or with a fault. Also a **Unsubscribe** message must be processed
to terminate a subscription.

4.11.2 OGC Web Notification Service (WNS)

The Web Notification Service (WNS) is a network-accessible service by which a client may conduct asynchronous message interchanges with one or more other services. An overview of the standard is available in the main document at subsection 4.1.6.

Functionality

The Web Notification Service is a definition for a component, which helps in constructing push-mode services.

Reactions on externally driven events or long-term actions mandated to services demand functions to support asynchronous communications between a user and the corresponding service, or between two services, respectively. The Web Notification Service fulfils these needs by forwarding incoming messages (based on HTTP) to the recipient using arbitrary protocols. The WNS acts as a transport transducer: It simply changes the protocol between incoming and outgoing messages. The WNS handles messages as a black box. It does not have any knowledge about the message content.

The possible means of notification are described in the “capabilities document” of the service. The palette may include:

- e-mail
- http-call (as HTTP POST: in case of sophisticated clients that act as web services themselves)
- SMS
- XMPP (IETF als RFC 3920–3923)
- phone call or fax

By default, a WNS provides at least the http transport protocol.

To make use of the notification capabilities, users have to be registered beforehand. This registration will be performed by either a user, or by another service that can act as a proxy for the user, which makes use of the notification functionality. In both cases, the WNS returns a registrationID. This ID, which is unique for every WNS instance, will be used to identify the receiver when a message shall be delivered using the WNS.

The Web Notification Service Model includes two different kinds of communication patterns. First, the “one-way-communication” delivers the message to the client without expecting a response. Second, the “two-way-communication” delivers the message to the client and expects some kind of asynchronous response.

Interface

The WNS interface is invoked over http using the GET and POST methods. The following
Operations are available:

**Operation: DirectoryRequest**
This operation allows a client to request and receive back service metadata documents that describe the abilities of the specific server implementation. This operation also supports negotiation of the specification version being used for client-server interactions. The implementation of this operation is mandatory.

**Operation: GetWSDL**
This operation is optional and allows a client to request and receive back the WSDL definition of the server interface.

**Operation: Register**
This operation allows a client to register itself by providing its communication endpoint. The implementation is mandatory.

Two cases are differentiated: SingleUserRegistration and MultiUserRegistration.
While the former links multiple communication endpoints to a (single) user ID the latter links multiple user IDs to another (multi) user ID, thus creating a group. Any message sent to this group will be delivered to all group members. The WNS is responsible for avoiding circular dependencies between different multi user groups.

**Operation: Unregister**
This operation allows a client to unregister itself, all implementations need to provide this.

**Operation: UpdateSingleUserRegistration**
This operation allows a client to update a previous registration by providing a new communication endpoint (e.g. an email address or a telephone number). Implementation is optional.

**Operation: UpdateMultiUserRegistration**
This operation allows a client to update a previous MultiUserRegistration by adding or deleting individual group members. Implementation is optional.

**Operation: DoNotification**
This operation allows a client to send a message to the WNS, which will be forwarded on the protocol defined by the registered client. In addition to the message, the calling client has to provide the registrationID of the registered client. Implementation is mandatory.

**Operation: GetMessage**
This operation allows a client to retrieve a message which has not been delivered by the WNS because of restrictions set by the chosen transport protocol. If notification via SMS or phone call is desired, then the WNS will forward the contents of the ShortMessage element of the DoNotification request together with a unique ID assigned to that message (for later retrieval of the complete message via the GetMessage operation). A mandatory operation.
4.12 Digital Rights Management and Security

Under Digital Rights Management (DRM) Services we understand services suited and needed to control all aspects of the use of services and data based on rights and licensing information. This can be authentication and access control (classical security), but also far-reaching concepts which enforce strict regulations of re-use and distribution of resources at the client/licensee side.

All descriptions in this section have in common that they are build around a specific language, an encoding, which is suited to express the demands of the specialised Digital Rights Management Service.\(^{15}\)

DRM and Security are issues of on-going research. Only a small part of available standardisation endeavours can be presented here. Most current standardisation, especially in the direction of security in web services, occurs in OASIS.

4.12.1 OGC Geospatial Digital Rights Management Reference Model (GeoDRM RM)

GeoDRM RM is a reference model for digital rights management (DRM) functionality for geospatial resources. An overview of the standard is available in the main document at subsection 4.3.3.

**Functionality**

The GeoDRM RM defines the framework for web service mechanisms and rights languages to articulate, manage and protect the rights of all participants in the geographic information marketplace, including the owners of intellectual property and the users who wish to use it.

DRM primarily is a metadata-tracking problem. Both resources and actors against these resources (called principals) are associated to metadata records, which must be tracked and matched for the actions to proceed. The resource metadata is the resource identity and description and the principal metadata is the set of licences he or she possesses.

Second, DRM is an enforcement problem. Once identity and licences have been checked, the DRM system has to control the scope of the actions of the licensor against the resources to a degree determined by the design of the DRM system.

Essential components of the GeoDRM functional architecture are:

- any service in the usual sense, the working of which is to be DRM controlled,
- a security component, which does authentication and signatures,
- a GeoDRM Gatekeeper, which verifies if request and licences match to allow execution of the request.

\(^{15}\) Actually, following the logic of this document, we might have explained the content of those encodings, the encodings themselves, and the DRM services at different places of the document. However, though DRM is an important issue, it is peripheral to eMOTION and therefore everything is being described in this section.
Only the GeoDRM Gatekeeper is in the scope of the definitions of the GeoDRM RM standard. Particularly the security component, is to be realised using mainstream standardisation, like SAML etc.

**Interface**

The GeoDRM Gatekeeper essentially has to deal with the service requests finally to be directed to the DRM controlled service. These requests will have to be enriched by authentication information and validated documents (supplied by security).

Therefore GeoDRM Gatekeepers have interfaces, which are closely related to the service interfaces they are providing DRM control for.

### 4.12.2 OASIS eXtensible Access Control Markup Language (XACML) / GeoXACML

The *eXtensible Access Control Markup Language (XACML)* centres around an encoding for expressing security policies regarding any possible resources. There is also an extension to XACML by OGC, called *GeoXACML*, which adds spatial restrictions to XACML. An overview of the standard is available in the main document at subsection 4.3.5.

**Functionality**

The specification defines the syntax and semantics for XML-encoded access control rule statements and policy statements. The XACML schema builds on SAML.

XACML is described in two models that describe different aspects of its operation. These models are:

- the policy language model and
- the data-flow model.

The language model defines the XACML encoding. It consists of a system of Rules, which belong to Policies, which again belong to PolicySets. The constructs own a high expressiveness and generality and are capable of describing complex access constraints. The language is extensible and can be employed to nearly arbitrary environments.

The data-flow model is relevant from the service perspective, because it points out the responsibilities of various (logical) components in a XACML controlled computing environment.

The model operates by the following steps.

- Policy Administration Points (PAPs) write Rules, Policies and PolicySets and make them available to Policy Decision Points (PDPs). These Policies and PolicySets represent the complete set of policy for a specified target.
- The access requester sends a request to a Policy Enforcement Point (PEP).
- The PEP sends the request for access to a so-called Context Handler in its native request format, including attributes regarding the requesting subject, resource, action, environment.
• The Context Handler constructs a XACML request context and requests a decision from the PDP, which has complete knowledge of all policies. PDP and Context Handler may negotiate further attributes.

• If further attributes are requested from the Context Handler, it may ask the Policy Information Point (PIP) to receive them. Optionally, the Context Handler also includes the Resource in the context it provides to the PDP.

• The PDP returns the response context to the Context Handler. This includes the authorisation decision.

• The Context Handler translates this back to the native response format and returns it to the requesting PEP.

Interface

The interfaces of the logical components are not specified by the standard.

The PAP to PDP interface accepts XACML. Other interfaces are implementation dependent.

The access requester to PEP interface is of course implementation dependent, however, it is required to carry security (authentication) information.

4.12.3 OASIS Security Assertion Markup Language (SAML)

OASIS Security Assertion Markup Language (SAML) specifies an encoding (an XML language) for use in a service oriented environment. An overview of the standard is available in the main document at subsection 4.3.4.

Functionality

Security Assertion Markup Language (SAML) is an XML standard for exchanging Authentication and Authorization data between Security Domains, that is, between an identity provider (a producer of assertions) and a service provider (a consumer of assertions).

The single most important problem that SAML is trying to solve is the Web Browser Single Sign-On (SSO) problem. Single sign on solutions are abundant at the Intranet level (using HTTP cookie, for example) but extending these solutions beyond the Intranet has been problematic and has led to the proliferation of non-interoperable proprietary technologies. SAML has become the definitive standard underlying many web Single Sign-On solutions in the enterprise Identity management problem space.

SAML assumes the principal (often a user) has enrolled with at least one identity provider. This identity provider is expected to provide local authentication services to the principal. However, SAML does not specify the implementation of these local services; indeed, SAML does not care how local authentication services are implemented (although individual service providers most certainly will).

Thus a service provider relies on the identity provider to identify the principal. At the principal's request, the identity provider passes a SAML assertion to the service provider. On the
basis of this assertion, the service provider makes an Access control decision.

**Interface**

Neither the interface for accessing identity providers, nor the provision of these identities (assertions) by identity consumers are specified.

There are other OASIS standards, which bind SAML to service oriented architectures, like OASIS SOAP Message Security, etc.

### 4.13 Pricing and Ordering

For openly accessible Web Services, registration retrieval and discovery of openly accessible web resources (usually by means of a Registry Service), also demands a Web Service, which makes an automated pricing, ordering and delivery workflow possible.

Regarding this subject there was only one generally usable service definition found. We will give only a short description because the issue is rather peripheral to eMOTION.

#### 4.13.1 OGC Web Pricing and Ordering Service (WPOS)

The OGC Web Pricing and Ordering Service specification describes a web service which covers all standard geo-eBusiness processes like pricing, ordering and online delivery for spatial products. An overview of the standard is available in the main document at subsection 4.4.8.

**Functionality**

The pre-standard defines a web service and the necessary encodings to support e-Business processes (like pricing, ordering and online delivery) for spatial products. The definition used to express formula for pricing and pricing models is based on MathML, which is a W3C standard.

WPOS instances can be cascaded.

**Interface**

The web service uses defined request and response packages for HTTP and the GET and POST methods.

**Operation: GetCapabilities**

This operation returns service metadata including a list of products sold by the WPOS instance.

**Operation: GetPriceModel**

For a specified list of products this operation returns the pricing models in XML Configuration and Pricing Format (XCPF), which will not be examined in this analysis. This operation is optional.
**Operation: GetPrice**

For a list of products and associated explicitly given service requests, the operation returns the price for these, in the case that they were ordered.

**Operation: OrderProduct**

This operation orders the given products and associated service requests targeting the ultimate service, whose services are priced and being ordered. For this the client also has to supply authentiction information. The result of this operation is a TAN.

**Operation: GetProduct**

Given an TAN, the ordered products are returned.

**Operation: GetOrderList**

This operation is optional and returns the current status of ordering as a XCPF document.

### 4.14 Payment and Billing

Payment and Billing procedures are very much determined by the communication channel. In the case of mobile devices, where various cost models and charging procedures are already well established, a solution will probably be possible in the realm of existing infrastructure.

For openly accessible Web Services, things are not so easy. Registering, retrieving and finding openly accessible web resources (usually by means of a Registry Service), consequently also demands for a Web Service, which makes an automated pricing, ordering and delivery workflow possible.

#### 4.14.1 Electronic Payment System (eps)

esps (Electronic Payment System) is a simple, safe and free of charge standard of Austrian Banks for e-Commerce and e-Government payments. An overview of the standard is available in the main document at subsection 4.4.1.

The following diagram shows the workflow of an eps transaction.

The data transfer between customer and bank or between the Austrian banks takes place with the aid of internationally normed Edifact notices.

For further reference see [http://www.eps.or.at/1109_EN](http://www.eps.or.at/1109_EN).

This web reference also makes available the “EPS Standard Implementation Guideline”, a document, which also contains the figure below.
4.14.2 Single Euro Payments Area (SEPA)

The Single Euro Payments Area (SEPA) will be created to give citizens, companies and other economic actors the opportunity to make and receive payments within Europe. An overview of the standard is available in the main document at subsection 4.4.6.

SEPA Components

SEPA Credit Transfer (SCT) Scheme

The scheme, rules and standards will enable the offering by providers of payment services of a core and basic credit transfer service throughout SEPA whether for single, bulk and/or repetitive payments. The scheme’s standards will facilitate payment initiation, processing and reconciliation based on straight-through-processing. Payments will be made gross and separate from charges; with no deduction from principal amount at any stage in the process. The scheme will permit the end-to-end carrying of remittance data on a structured and unstruc-
tured basis appropriate to the nature of payment. The scheme provides for a maximum execution time of 3 days between any point in SEPA, although many payments will arrive more quickly as is common today.

**SEPA Direct Debit (SDD) scheme**

Similarly the scheme rules provide for a new SEPA-wide core and basic direct debit service, whereby one-off and repetitive payments are collected on behalf of creditors from debtors for items such as utility bills, subscriptions, insurance premiums and goods and services. The scheme provides for mandate creation, predictable and convenient time-cycles, appropriate refund rules and the necessary features to permit end to end straight-through-processing.

![Figure 4-17: SEPA “replacement” of Direct Debit and Credit Transfer schemes](image)

For SCT and SDD, a “replacement” strategy has been chosen with new common credit transfer and direct debit schemes for the whole SEPA being envisaged to progressively replace the same services provided today in euro in national markets.

Text copied from:

http://www.europeanpaymentscouncil.eu/content.cfm?page=sepa_components

### 4.15 Workflow Support

When a series of automated service invocations has to be combined to accomplish the flow of business processing, Workflow Support Engines may be employed.

Processes in a Workflow Support Engine represents a set of publicly observable behaviors in a standardised fashion. They includes information such as when to wait for messages, when to send messages, when to compensate for failed transactions, etc.

From the analysis it has not become clear whether eMOTION has indeed a requirement for Workflow Support or whether automated invocations of services will be performed within Application Service environments.
4.15.1 OASIS Web Services Business Process Execution Language (WS-BPEL)

The OASIS Web Services Business Process Execution Language (WS-BPEL) specifies the common concepts for a business process execution language. An overview of the standard is available in the main document at subsection 4.6.11.

Functionality

WS-BPEL defines a model and a grammar for describing the behaviour of a business process based on interactions between the process and its partners. The interaction with each partner occurs through Web Service interfaces. The data model used by WS-BPEL processes is provided by WSDL messages and XML Schema type definitions. WS-BPEL processes will run on any WS-BPEL-compliant engine.

Interface

A business process can be modelled with the following XML elements:

- **process**: The outermost container of a business process is an XML element called `process`.

- **scope**: It is possible to divide a business process into scopes, which are different portions of the overall business logic. Process data can be declared local to a scope, so that they are only visible inside this scope.

- **partnerLink**: With partnerLink elements one can describe the interaction between business partners. For one partner there can be a set of partner links, each of them can be regarded as one particular communication channel. A partner link is characterized by a partner link type and a role name.

- **variable**: Variables hold process data, which constitute the state of a BPEL business process during runtime.

Further XML elements are provided with the purpose of consuming messages from and providing messages to web service partners (receive, reply, invoke), for structuring process logic (sequence, if-else), to allow the repeated execution of a piece of business logic (while, repeatUntil, forEach), and to enable parallel processing (flow). Also there are possibilities to manipulate data (assign) and to cope with exceptional situations with the concept of fault handlers (faultHandlers).

There are more advanced concepts of WS-BPEL available, which are described in the WS-BPEL 2.0 specification.

4.16 Network Management

As is the case with all distributed services infrastructures, the eMOTION framework is sus-
ceptible to the possibility that parts of the network fail or malfunction in unperceived ways. This can have serious consequences, entail customer irritation and financial losses. Therefore measures have to be taken to ensure that all vital components of eMOTION are working reliably all the time and that malfunctions are detected as soon as possible.

Network Management is a general concept, which is for example described in ISO/IEC 7498-4 (OSI Management). The concepts pointed out in this standard will have to be evaluated for the eMOTION architecture of service components.

4.16.1 Simple Network Management Protocol (SNMP)

The Simple Network Management Protocol (SNMP) is used to administrate and to monitor arbitrary network-attached devices. An overview of the standard is available in the main document at subsection 4.6.6.

The Simple Network Management Protocol (SNMP) is defined by the Internet Engineering Task Force (IETF) as part of the Internet Protocol Suite. It is used to administrate and to monitor arbitrary network-attached devices. With SNMP the system configuration of a network-attached device can be described in the form of variables, which can be queried and in some cases set by managing applications.

There are three different version of the SNMP in use. Since 2004 SNMPv3 is the current standard version of SNMP, earlier versions are considered obsolete by the IETF. SNMPv3 is defined by RFC 3410 - RFC 3418, which can be found at [http://www.ietf.org/rfc.html](http://www.ietf.org/rfc.html).

Relevance

As described in the introduction of the Network Management section, it is important that relevant parts of the eMOTION framework can be monitored, so that malfunctions can be detected.

Functionality

Different roles can be identified in SNMP:

A _Managed Device_ is a network node that provides information to management applications and may accept control information. The managed device contains an SNMP _Agent_, which is a network-management software module and has local knowledge of management information and translates that information into a form compatible with SNMP.

Any managed network must contain at least one _Network Management System_ (NMS). This is the managing application which controls and monitors managed devices.

The variables, which contain the information of an agent, are instances of _Managed Objects_. Managed objects are organized hierarchically in a so-called _Management Information Base_ (MIB). The MIB hierarchy can be depicted as a tree, where a managed object can be uniquely identified either by the object name or by an equivalent object descriptor. Different MIBs do exist: the main MIB is MIB-II, which is specified in RFC 1213, other MIBs ranging
from various aspects of network management, to system management, and application management.

**Interface**

Objects in a MIB are defined using a subset of Abstract Syntax Notation One (ASN.1) called "Structure of Management Information Version 2 (SMIv2)" to accommodate communication between diverse systems.

Attributes of managed objects may be monitored or set by the network management system using the following types of protocol operations:

- **Reads** - *Get, GetNext, GetBulk* (since v2):
  Used by NMSs to monitor managed devices. NMSs read variables maintained by the devices.

- **Traps** - *Trap, Inform* (since v2):
  Used by managed devices to asynchronously report certain events to NMSs.

- **Writes** - *Set*:
  Used by NMSs to control managed devices. NMSs write variables stored within the managed devices.

**4.17 Natural Language Translation**

An SOA for traffic information in Europe has necessarily to deal with the issue of multilinguality. There are several solutions to this problem, each of which does not solve the problem entirely.

On the side of Application Services there are well-known methods to create multi-lingual user interfaces, they can of course only deliver user interfaces for a predefined (usually small) set of natural languages, which is usually not a problem.

The problem is in the data delivered by services. Here there are three ways to cope with the problem:

- **Full modelling**: The server delivers its data items coded using predefined data models and enumerations of standardised meaning. Alert-C is an example for this. The Application creates clear-text messages from this, which effectively transforms the server issue to an application issue.

- **Multilingual server data**: The server returns data items in several languages, possibly selectable by request parameters. This is of course sometimes hard to accomplish, and there is often the problem that the languages offered by a service do not match the requirements of an application. Such a facility is built into TPEG services.

- **On the fly translation**: There is an service instance, which can translate clear text from one language into any other. The quality of translations is limited by the state of the art of machine natural language translation.

The last possibility shall be addressed in this section. It is covered with the OASIS Transla-
tion Web Service, which is a standard to automate the translation and localization process as a Web service, and a proprietary solution offered by SYSTRAN, the SYSTRANLinks translation service.

4.17.1 OASIS Translation Web Service (Trans-WS)

The OASIS Translation Web Service (Trans-WS) provides an encapsulation of all the information which is needed by publishers of content to be translated so that they are able to automatically connect to and use the services of any translation vendor. An overview of the standard is available in the main document at subsection 4.6.8.

**Functionality**

A localisation or translation task is a *job* in the context of Trans-WS. The interaction between the customer and the publisher of a job in the case of machine translation is not handled in the standard specification, so the functionality can not be described here.

4.17.2 SYSTRANLinks

SYSTRANLinks is a proprietary solution of a translation service offered by SYSTRAN. An overview of the standard is available in the main document at subsection 4.6.7.

**Functionality**

Using the SYSTRANLinks' API the customer of a translation must send the source document, the source language and the desired language using the HTTP POST method to a special SYSTRANLinks URL. SYSTRANLinks' API returns an XML document containing the translated document, with UTF-8 encoding. The service offers a mono-target and multi-target translation.
5. Analysis of Network and Communication Standards

This chapter deals with the network infrastructure of eMOTION and lists the relevant standards regarding the distribution of information in the network.

The main distribution channel of eMOTION information will be the Internet. So, regarding the distribution of information between the actors of the value chain of eMOTION, only little needs to be said. Though also the end user will receive his or her information through the Internet, it will (according to the project goals of eMOTION) be the mobile Internet, which has its own particularities. These have to be pointed out and standards will have to be explained.

5.1 Internet Communication Standards

*The Internet protocol suite*

The Internet protocol suite is a collection of network protocols implementing the stack that constitutes the basic “building blocks” for the Internet. An overview of the Internet protocol suite is available in the main document at subsection 4.5.6.

![The TCP/IP model](image)

At the lowest level there is the IP (Internet Protocol), defining how to carry blocks of data from one node to another by datagrams or packets. Next come TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) - the protocols that are used to send data to one host to another. On top come application protocols that define the format for messages and data exchanged by applications. As pointed out in section 1 the Internet Protocol suite can work over any communications network, wired or wireless capable of carrying two-way digital data.

In this section an overview of TCP/IP’s Application layer standards that are of relevance for eMOTION are provided. For completeness the main relevant Transport layer standards are introduced. For the Internet layer standards see section 5.5.1.
5.1.1 Hyper Text Transfer Protocol (HTTP)

An overview of HTTP is available in the main document at subsection 4.5.6 - HTTP.

Functionality

HTTP works with a request/response mechanism (client/server): the client (e.g. a browser) sends a request and the server (e.g. a web server) gives a response. Two types of messages can then be defined: a request message and a response message.

A request message is composed by:

- A Request line
- Headers
- An empty line
- An optional message body

The request line is composed by a Method, a URI and a protocol version. A method can be:

- GET
- POST
- HEAD
- PUT
- DELETE
- TRACE
- OPTIONS

URI stands for Uniform Resource Identifier and identifies the resource addressed by the request (e.g. a page, an image etc.) For example a GET method, combined with the URI of a page enables the retrieval of a specific HTML page.

The most common Headers includes:

- Host (server name the URI is referring to)
- User-Agent (identifies the type of client: browser, manufacturer, version etc)

A response message is made of three parts:

- A status-line
- An header section
- A Body (response content)

The status line contains information about the result of the request (success, error, redirection etc.) A code is returned and a more detailed message can be include, depending on a particular code, on the body. Examples of status line can be:

200 (OK)

301 Moved Permanently.
302 Found. (the resource can be reached by another URI)
400 Bad Request.
404 Not Found.
500 Internal Server Error

etc.

The most common **Response headers** are:

- **Server**: indicates the type and version of server (similarly to the User agent request header)
- **Content-Type**: Indicates the type of content that has been returned. (MIME types: Multimedia Internet Message Extensions types – for example text/html, text/plain, image/jpeg etc.)

**Interface**

The HTTP protocol is an application protocol. It interfaces directly to level 4 (Transport) of the TCP/IP stack using its services. Typical purpose of transport level is to provide a packed-oriented end-to-end communication channel. Protocols used by HTTP may include for example TCP or UDP at transport level.

5.1.2 **Hyper Text Transfer Protocol Secure (HTTPS)**

An overview of HTTPS is available in the main document at subsection 4.5.6 - HTTPS.

**Functionality**

HTTPS has an URI (Uniform Resource Identifier) with a syntax identical to HTTP but it uses another communication port (443) and an encryption/authentication level is introduced between the TCP and HTTP protocol.

HTTPS isn’t a separate protocol but combines HTTP with an encryption mechanism that can be Secure Sockets Layer (SSL) or Transport Layer Security (TLS) as described in the following sections.

**Interface**

See 5.1.1

5.1.3 **Secure Sockets Layer / Transport Layer Security (SSL/TLS)**

An overview of SSL/TLS is available in the main document at subsection 4.5.6 - SSL/TLS.

**Functionality**

Both TLS and SSL protocols defines some basic steps:

- Negotiation by the two parties on the algorithm to use
- Exchange of private keys with a public-key based encryption method and identification with certificates
- Symmetric-cipher based encryption of data for traffic between the two parties.
Recent implementations uses 128 symmetric keys for encryption.

**Interface**

Security protocols are under an application protocol such as HTTP, SMTP, NNTP and work over the TCP transport protocol.

### 5.1.4 Secure Shell (SSH)

An overview of SSH is available in the main document at subsection 4.5.6 - SSH.

**Functionality**

The SSH-2 protocol has a layer-based internal architecture:

The *transport layer* handles the server authentication and initial key exchange. It sets up compression, encryption and integrity verification. The functionality of this layer is comparable to TLS.

The *user authentication layer* deals with client authentication and provides different methods for authentication (password-based but also other types). This layer is highly extensible thanks to the custom authentication methods.

The *connection layer* handles channels. A single SSH connection can host multiple bi-directional channels simultaneously to fit several channel requests originated by different user operations. Standard channel types include: "shell" for terminal shells, "direct-tcp/ip" for client-to-server forwarded connections, and "forwarded-tcp/ip" for server-to-client forwarded connections. This layer allows many secondary sessions to be multiplexed into a single SSH connection (a feature not available in TLS).

**Interface**

SSH is typically used to log into a remote machine and execute commands, but it also supports tunnelling, forwarding arbitrary TCP ports and X11 connections; it can transfer files using the associated SFTP or SCP protocols. An SSH server, by default, listens on the standard TCP port 22.

### 5.1.5 Domain Name System (DNS)

An overview of DNS is available in the main document at subsection 4.5.6 - DNS.

**Functionality**

DNS principles can be illustrated as follows:
DNS systems have a hierarchical namespace implementation allowing to forward a name request for a given zone to another DNS and so on. An iterative search procedure is necessary because the complete information is not available at the first step and more servers have to be queried. Each time a query is forwarded, a pointer to the next DNS server to consult is provided.

The information about names is updated with automated mechanisms from a server to another.

The key element for storing information is the DNS Records. Since various types of information can be associated to a DNS name, several types of DNS records exist.

Here’s an overview of some of the most common record types:

- The “A” record is the most common record and indicates the correspondence between a name and one (or more) IP addresses.
- “MX” (Mail eXchange) records are used to indicate to what server the email traffic has to be sent for a given domain
- “CNAME” records are used to create an Alias or, in other words, to make a single host identifiable by more than one single name.
- “PRT” records are used for the reverse DNS lookup.
- “NS” (Name Server) records are used to indicate what DNS must take over the request.

Multiple records (of the same type or of different type) can be associated to the same domain name to soften the workload in case of many accesses.

A cache system is used to improve performances and since records often vary along the time a “time to live” (TTL) is associated to each record to define its period of existence.

**Dynamic DNS** (DDNS) indicates a group of technologies that allow a non-predefined address, obtained for example with the DHCP or PPP protocol, to be automatically inserted into a DNS zone. This is a typical functionality of the Windows Active directory service and is commonly used by Internet providers to allow dial-up users to register a name corresponding...
to the address dynamically assigned to them at each connection.

Interface

The DNS uses TCP and UDP on port 53 to serve requests. Usually DNS queries consist of a single UDP request from the client followed by a single UDP reply from the server.


An overview of SMTP/POP3/IMAP is available in the main document at subsection 4.5.6 - SMTP/POP3/IMAP.

Functionality

SMTP is a very simple protocol based on text commands. A SMTP-based communication between a server and a client is based on the exchange of such commands. Here’s an example:

- S: 220 www.mail.com ESMTP Postfix
- C: HELO www.client.com
- S: 250 Hello www.client.com
- C: MAIL FROM: <sender@client.com>
- S: 250 Ok
- C: RCPT TO: <emotion@mail.com>
- S: 250 Ok
- C: DATA
- S: 354 End data with <CR><LF>.<CR><LF>
- C: Subject: test message
- C: From: sender@client.com
- C: To: emotion@mail.com
- C: this is a test
- S: 250 Ok: queued as 12345
- C: QUIT
- S: 221 Bye

POP3

After the connection between client and server has been established a POP3 session is opened. An exchange of text commands is then used to perform the requested operation. For example the retrieval of an email can be done with the following commands:

- S:+OK <22593.1129980067@example.com>
- C:USER emotion
- S:+OK
- C:PASS emotion
- S:+OK
- C:LIST
- S:+Ok
- 1 817
- 2 124
- C:RETR 1
- S:+OK
- Return-Path: <emotion@example.org>
- Delivered-To: emotion@example.org
- Date: Mon, 18 Jun 2007 10:14:00 +0100
- From: eMOTION Consortium < emotionconsortium@example.org >
- Subject: test
- Content-Type: text/plain; charset=ISO-8859-1
- test
- C:DELE 1
- S:+OK
- C:QUIT
- S:+OK

Interface
SMTP, POP3 and IMAP works over a TCP/IP connection.

5.1.7 File Transfer Protocol (FTP)
An overview of FTP is available in the main document at subsection 4.5.6 - FTP.

Functionality
The functionality principles of FTP can be represented with the following diagram:
A connection to the server’s port 21 from the FTP Client forms the control stream used for the commands exchange from client to server and vice versa. The actual file transfer is carried on a separate connection called the data stream.

The server Protocol Interpreter (PI) waits for a connection from the User PI and establishes the control connection for the communication. It receives standard FTP commands from the User IP and send responses.

The Server Data Transfer Process (DTP) establishes connections using the listening port. It sets parameters for transfer and storing and implement the transfer on the basis of appropriate commands.

**Interface**

FTP uses TCP to establish a connection for control information (out of band connection) and then it established a second TCP connection for data transfer. FTP servers by default listen on port 21 for incoming connections from FTP clients.

**5.1.8 Transmission Control Protocol (TCP)**

An overview of TCP is available in the main document at subsection 4.5.6 - TCP.

**Functionality**

Three phases characterizes a TCP connection:

- connection establishment
- data transfer
- connection termination
Connection establishment

TCP make provision of a connection establishment phase before sending data. There is a difference in that sense from UDP which can immediately start sending packets.

The server must bind to a port that will be used for the connection (passive open) before a client attempts to connect to the server. At this point an active open can be initiated by the client using a three-way handshake: (a SYN is sent to the server - the server replies with a SYN-ACK - the client sends an ACK back to the server)

Data Transfer

The data transfer in TCP has several key features:

- Connection-oriented protocol.
- Functionality to create, keep and close a connection
- Guaranty of error-free, ordered and “at most once” delivery of data (discarding of duplicate packets) by means of acknowledgements and re-transmission mechanisms.
- Data flow and congestion control done by means of a “shifting window” (the amount of received data, in bytes, that can be buffered during a connection).
- Possibility to establish multiplexed connections by using several ports.

Connection termination

The Connection termination phase can use different handshakes in different ways. A FIN signal is sent by the end point willing to terminate the connection. The other end acknowledges it with an ACK. This must happen from each side to terminate the communication. The FIN/ACK response from the receiver can be combined forming thus a 3-way handshake.

Interface

In a TCP connection each side has an associated 16-bit port number ranging from 1 to 65535. This is a reserved channel for the transmission between the sender and the receiver end points (host or Internet socket). TCP data packets belongs to a communication socket identified by the combination of source host address, source port, destination host address, and destination port. Several TCP connections can then be initiates as long as the mechanism of socket-based connection is used by applications or processes. The application protocol used by an application or process will then provide use of the specific port to establish the communication. For instance some applications running as servers can be in a state of “listening” for connection on a dedicated port. Examples are HTTP on port 80, FTP on port 21, SSH on port 22 etc.

At a lower level TCP is nearly always used in combination with the IP protocol.

5.1.9 User Datagram Protocol (UDP)

An overview of UDP is available in the main document at subsection 4.5.6 - UDP.

Functionality
Key features for UDP are:

- Connection multiplexing, by means of ports mechanism
- Error check carried out with a checksum embedded in the packet header.

A basic UDP datagram is structured as follows:

<table>
<thead>
<tr>
<th>Bit 0-15</th>
<th>16-31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Port (optional)</td>
<td>Destination Port</td>
</tr>
<tr>
<td>Length</td>
<td>Checksum (optional)</td>
</tr>
<tr>
<td>64+</td>
<td>Data</td>
</tr>
</tbody>
</table>

- Source port [16 bit] – is the host port number of the datagram’s sender
- Destination port [16 bit] – is the destination host port number
- Length [16 bit] – is the total length of the UDP datagram (header and data)
- Checksum [16 bit] – is the control code for the datagram whose computation algorithm depends on the lower level protocol used (e.g. it vary from IPv4 to IPv6)
- Data – the datagram’s data

UDP is stateless regarding the connection, hence there is less information to handle. A server dedicated to UDP-based applications can then support more active clients.

**Interface**

In the Internet protocol suite, UDP provides a simple interface between the network layer at the lower level (e.g., IP) and a session (OSI model) or application layer on the upper level. Similarly to TCP, UDP uses the mechanism of ports to allow application-to-application communication.

5.1.10 IPv4/IPv6/IPsec

For a description of IP and its versions see section 5.5.1.

5.1.11 SOAP Version 1.2

An overview of SOAP is available in the main document at subsection 4.5.6 – SOAP.

**Functionality**

SOAP is a standardised packaging protocol providing a simple XML based envelope for the information being transferred. It also provides a distributed processing model which assumes a SOAP message originates at an initial SOAP sender and is sent to an ultimate SOAP re-
The structure of a SOAP message can be depicted as in the following figure:

![SOAP Message structure diagram](image)

**Figure 5-4: SOAP Message structure**

The *message body*, which is enclosed in the *SOAP body* contains the actual message to be delivered or processed, also called the payload of the SOAP message. Exactly one *SOAP body* must be present in a SOAP message. Anything expressed in XML can go in the body of a message.

The *SOAP header* is optional and contains (if present) zero or more so-called *header blocks*. *Header blocks* contain routing and delivery settings, authentication or authorisation assertions, and transaction contexts.

The following XML snippet depicts a SOAP message without any content:

```xml
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://www.w3.org/2001/12/soap-envelope">
  <s:Header>
  </s:Header>
  <s:Body>
  </s:Body>
</s:Envelope>
```

While a SOAP message is sent by an SOAP sender and is received by a SOAP receiver, any number of so-called intermediary web services may be involved, which do some specific processing with the message. The specification of the exact path a message takes is not part of the SOAP specification. However, a mechanism for identifying which *headers* of the SOAP message are intended for processing by specific actors can be specified by means of the *role*-Attribute. Processing by an intermediary actor will interpret the parts of the message "targeted" to the actor and will possibly change the message while passing it along. An intermediary actor might i.e. extract a digital signature from a header, validate it, and add a
new header indicating to the ultimate receiver service that the signature is valid.

SOAP packaging has two related applications:

- **Remote Procedure Call (RPC)**
  
  RPC is the basis for distributed computation, where one computational entity calls another passing arguments and receiving results.

- **Electronic Document Interchange (EDI)**
  
  EDI is a basis for business transactions and defines a standard format and interpretation of documents and messages. It is often called “document-style SOAP”.

While document-style SOAP is intended to transport any payload coded in XML in its message body, RPC-style SOAP provides additional mechanisms for serializing the transmitted data to make the structure of “calls” and “argument passing” a common knowledge of sender and receiver.

SOAP supports encoding rules to achieve this. Principally, sender and receiver can agree to any encoding. However, the most interoperability is achieved when general encoding rules are employed. Here the SOAP specification makes use of the XML Schema definition, which can either be used to fully define the message structure, or by just using the type simple system of XML Schema with some additions from the SOAP specification.

**Interface**

SOAP 1.2 is defined by its message model, the processing model and the various protocol bindings.

The SOAP message constructs can be exchanged over a variety of underlying protocols, such as HTTP, FTP, raw TCP, SMTP, POP3. The most important transport is of course SOAP over HTTP.

5.2 **Public Wireless Networks (GSM/GPRS/EDGE/UMTS)**

An overview of Public Wireless Networks is available in the main document at subsection 4.5.8.

5.2.1 **Standards**

The standardization of GSM/GPRS/EDGE/UMTS is a real successful story that has started in the European ETSI (European Telecommunications Standards Institute) GSM/SMG groups context, more than 20 years ago (for curiosities and information see the recent 20 years of GSM workshop proceedings). The activities were passed in 1999 to the global 3GPP project, which was officially tasked to carry on the system enhancement and development, and to maintain the previous specification developed in the ETSI context. 3GPP received this task, not only from ETSI, but also from the major telecommunication standard organization of Japan, Korea, China and North America. The work is carried out by individual members that are affiliated to the mentioned organizations. Currently the members of 3GPP are more than 300. The organization includes 22 technical standardization groups that have very frequent
face to face meetings. All the recommendations are publicly available along with the workplan.

A description of the technical content of the specifications releases is also available on the 3GPP website.

A reduced and incomplete, but simple to remember view could be:

- Pre-release 96: is basic GSM for voice and SMS, very low speed connection oriented data.
- Release 96 to release 99(excluded): is GPRS/ EDGE plus several enhancements to GSM,
- Release 99 covers UMTS access
- Release 4: softswitch approach, basic HSDPA, IP transport, EDGE enhancements
- Release 5: Basic IMS
- Release 6: Enhanced IMS, Interworking with W-LAN,
- Release 7: Enhancements to all the system, basics for LTE/SAE, some basics of FMC
- Release 8: currently under definition: LTE/SAE, ALL–IP approach, FMC developments

5.2.2 Data Service Throughput

In this section, some information is provided about the sustainable bit rates when using different technologies. The reference is not about the technical theoretical potential capabilities of a specific access network, but refers to the reasonable values that can be achieved by the final user considering the current deployments and those planned for the near future.

<table>
<thead>
<tr>
<th>GSM</th>
<th>GPRS</th>
<th>EDGE</th>
<th>U MTS</th>
<th>HSxPA</th>
<th>SAE/LTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>75Kbit/s</td>
<td>160 Kbit/s</td>
<td>&lt;0.5 Mbit/s</td>
<td>2 Mbit/s</td>
<td>14,4 Mbit/s (6Mbit/s downlink)</td>
<td>100 Mbit/s (50Mbit/s downlink)</td>
</tr>
</tbody>
</table>

*Table 1 – Maximum data bit rates per bearer (cell)*

<table>
<thead>
<tr>
<th>Dedicated channel</th>
<th>GSM</th>
<th>GPRS</th>
<th>EDGE</th>
<th>U MTS</th>
<th>HSxPA</th>
<th>SAE/LTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;10 Kbit/s</td>
<td>&lt;64 Kbit/s (Real networks) (384 max)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Shared</td>
<td>&lt;50 Kbit/s</td>
<td>80-200</td>
<td>150-384</td>
<td>1 Mbit/s</td>
<td>5 Mbit/s</td>
<td></td>
</tr>
</tbody>
</table>
The actual bit rates are related to the medium radio quality offered by the network, and can be very different depending on the specific location and the specific traffic conditions. In particular GPRS and EDGE bit rates depend strongly on the distance from the radio BTS (Base Transceiver Station), and form the different fading conditions. Additionally UMTS, being interference limited, is sensitive to the traffic generated by the others users, and even few users at the cell border sending a significant amount of data, could result in a strong degradation of the bit rate of the overall users.

### 5.2.3 Cell radius, Coverage and capacity

The cell radius really depends on the environment: factors such as orography, urbanization, the buildings, the type of construction material are particularly significant. The figures reported in the table below can be considered typical values.

<table>
<thead>
<tr>
<th></th>
<th>GSM</th>
<th>GPRS</th>
<th>EDGE</th>
<th>UMTS</th>
<th>HSxPA</th>
<th>SAE/LTE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Min</strong></td>
<td>0,5-5 Km (900 MHz)</td>
<td>Same as GSM</td>
<td>Same as GSM</td>
<td>400-800m (2GHz)</td>
<td>Few hundred meet-ers</td>
<td>Few hundred meet-ers</td>
</tr>
<tr>
<td></td>
<td>0,4-1 Km (1800-1900 Mhz)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Max</strong></td>
<td>25-28 Km (900 MHz)</td>
<td>Same as GSM, but the bit rate decreased dramatically</td>
<td>Same as GSM, but the bit rate decreased dramatically</td>
<td>2-10 Km (2GHz)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60-80 Km (900 MHz extended range capacity reduced to ½-1/3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2-6 Km (1800-1900 MHz)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 3 – Typical Coverage radius**

Additionally, it shall be underlined that the previous numbers only consider coverage aspects, and are valid with limited traffic conditions.

For deployment in densely populated areas, such values are really reduced to the minimum, in order to reuse the scarce radio resource as much as possible in the geographical space (i.e. to reduce inter-call interference, the transmitting power shall be significantly limited)
5.2.4 Mobility

The system is fully designed to support mobility. GSM/GPRS/EDGE/UMTS are capable to work at speeds of more than 500 Km/h. At this speed the system is capable to provide a seamless experience to the users with dedicated channels, and shortly also with shared channel (currently the change of cell may be noticeable to the application, especially in case of streaming data with no buffer).

The system is based on mobile specific mechanisms to control mobility, based on radio mobility for micromobility and GTP (GPRS Tunnelling Protocol, working on top of IP) for Macro-mobility.

Additionally, the system may also be used on top of GTP Mobile IP. But the use of Mobile IP is only theoretic, it is not currently used in real networks, it being the mobile specific GTP more efficient and performant in providing mobility.

5.2.5 Application protocols

The mobile systems are substantially agnostic with respect to the user application protocols used over IP. Every application protocol could be used to access servers external to the mobile system (obviously specific protocols shall be used to access to mobile system services).

Obviously mobile oriented protocols present some benefits in terms of efficiency (e.g. XDML, C-HTML, I-mode protocol suite, etc), although such benefits are slowly becoming less significant with the development of broadband access, which are making the Mobile network broadband more and more similar to the fixed networks one.

Moreover the IP layer itself could become an transported application for the Mobile system: in parallel to IP service, L2TP feature is also supported in GPRS/EDGE/UMTS packet services, so a transparent tunnel could be set up between the terminal and a specific network service based on the configuration of appropriate APN addressing.

5.3 IP Multimedia Subsystem (IMS)

IP Multimedia Subsystem is a system consisting of a telecommunication IP network with an innovative call/session control layer. An overview of IMS is available in the main document at subsection 4.5.5.

5.3.1 IMS general description

The IP Multimedia Subsystem is a new domain that uses SIP (Session Initiation Protocol) to establish IP connections between mobile terminals and between mobile terminals and network. The IP connections can then be used to carry any IP traffic, for example interactive game sessions or push to talk communications. IMS domain can be considered as an intermediate layer between connectivity layer and application layer for control signalling to set-up, maintain and terminate multimedia sessions.
IMS capable user equipments (UE) utilize the GPRS network (with GERAN and/or UTRAN radio access) as an access network for accessing the IMS domain. The UE has to activate a PDP Context to the IMS before it can proceed to use the IMS services; PDP-Context gives IP-Connectivity to the mobile terminal.

Since IMS is supposed to be *access independent*, any PS networks can be used as IMS access network; the goal is an all-IP mobile network where IP is used in all networks domains.

### 5.3.2 Architecture

In the following figure the IP Multimedia Subsystem (IMS) architecture is described.
The IP multimedia domain architecture core is the **Call State Control Function (CSCF)** which is an SIP server handling session set up and release, admission control, charging and security. In this way, the CSCF can be compared functionally to the MSC Server of the CS domain. The CSCF operates on top of the basic IP connectivity services offered by the packet domain. It is based on the SIP protocol and provides mechanisms for advanced coordination of single/multiple session to support establishment of multimedia services for the end user. Logically, the CSCF can be divided into 3 components: Interrogating CSCF, Proxy CSCF and Serving CSCF. The CSCF has an interface towards the Application and Service environment to provide value added IM services.

The **Serving CSCF** (S-CSCF) provides users services. The S-CSCF hosts the network functionality for the provision of calls/sessions. It interacts with the HSS and the application platform to obtain subscriber data and for application service provision. It holds registration state and call/session states. It generates Call Detail Records (CDRs) for charging and resource utilization. The architecture is based on the principle that the service control for Home subscribed services for a roaming subscriber is in the Home network; for this reason the Serving-CSCF is always located in the Home network.

The **Proxy CSCF** (P-CSCF) provides a first point of contact for the handset. All signalling to and from the handset goes through the P-CSCF. Its address is discovered by UEs after PDP context activation, using the mechanism “CSCF Discovery”. The P-CSCF behaves like a Proxy, i.e. it accepts requests and services. The P-CSCF may also, in abnormal conditions, terminate and independently generate SIP transactions.
It generates CDRs.

The Policy Decision Function (PDF) is a logical entity of the P-CSCF that plays a role in quality of service reservations (authorization of bearer resources and QoS Management) to enable coordination between events in the application layer and resource management in the IP bearer layer. The PDF makes decisions with regard to network based IP policy using policy rules, and communicates these decisions to the GGSN, which is the IP Policy Enforcement Point (PEP).

The Interrogating-CSCF (I-CSCF) is the contact point within an operator’s network for all connections destined to a subscriber of that network operator, or a roaming subscriber currently located within that network operator’s service area. The HSS keeps track of this information so that all transactions and all calls for the same user are routed to the correct service node. The I-CSCF generates CDRs. The operator may use the I-CSCF or other techniques to hide the configuration, capacity, and topology of the network from the outside.

The Home Subscriber Server (HSS) is a master database storing subscription and location information for subscribers belonging to the network. The HSS is an upgrade of the HLR capable of handling SIP functions as well. The HSS functionalities are:

- LR functionality: Home Location Register and Authentication Centre (HLR/AUC) functionality
- IP Multimedia functionality: Stores IMS related data including: private and public user identities, IMS access parameters, service triggering information, user-specific requirements for S-CSCF capabilities.

The Multimedia Resource Functions (MRF), that are divided into two functionalities, Controller (MRFC) and Processor (MRFP), are resources of the IMS that provide support for bearer related services such as for example multi-party sessions, announcements or bearer transcoding.

The Interworking functionalities are performed by:

- **Media Gateway (MGW)**: connects the GGSN node, that is directly connected to surrounding Multimedia IP networks, to circuit switched networks;

- **Media Gateway Control Function (MGCF)**: control the circuit interface on the Media Gateway;

- **Breakout Gateway Control Function (BGCF)**: the CSCF communicates with it for sessions to/from legacy networks;

- **Roaming Signalling Gateways (R-SGW)**: supports IP connection to PSTN and other external networks

- **Transport Signalling Gateways (T-SGW)**: it is used to interconnect different signalling networks i.e. SCTP/IP based signalling networks and SS7 signalling networks;
5.3.3 Quality of Service (QoS)

IMS is SIP centric: during a SIP session the user equipment can negotiate its capabilities and can express its QoS requirements. For this reason the network has to have the capabilities to reserve the resources for the required services.

The end to end QoS that can be offered to the end users is a very large and complex issue because it involves several elements: IMS system, Radio Access Network (RAN), Core Network (CN), intra and extra mobile Operator Application Servers, all the external networks involved in the provision of the services and Mobile Terminals:

QoS control features implemented in IMS are only able to control the maximum level of QoS that a user can request to the network, according to his subscription profile and service requirements. After that these service requirements have to be mapped in the supporting networks to reserve the necessary resources. For this reason it should be considered which QoS procedures are implemented in the following network elements.

- **Radio Access Network (RAN):** e.g. Admission control, all the functionalities that are able to do the mapping between services requirements and RAB, priority and pre-emption policies;
- **Core Network (CN):** e.g. DiffServ, scheduling policies, admission and flow control;
- **Application Servers:** e.g. availability time, downtime, Mean Time Between Failures;
- **External networks involved:** e.g. appropriate SLAs in order to guarantee the QoS in the interconnecting backbone out of the mobile operator control.
- **Mobile Terminals:** e.g. RAB supported, codec compatibility;

Regarding Radio Access Network, the functionalities used for QoS negotiation and/or radio resources reservation are different in WCDMA and in GSM networks.

TIMS is the first step towards an all-IP mobile network: end-to-end QoS has to be provided when IP is used in all network domains, including both the radio access network (RAN) and the core network (CN) domains.

5.3.4 Security

Following the layering paradigm, the IMS domain, with the functionality of application layer, has to be independent and separate from the lower technology used both as access (GERAN, UTRAN, W-LAN, xDSL,...) and as Core Network (PS or CS). Therefore IMS layer needs to authenticate the users for the application level, in addition to network authentication.

The authentication techniques differ in the algorithms and in the usage of security information included in the IMS terminals, in the S-CSCF and in the HSS. The proposed authentication techniques are:

**HTTP-Digest Authentication:** which doesn’t use parameters in the SIM/USIM, but it is a challenge/ response algorithm based on the use of username and a pre-shared
password

**SIM/USIM based authentication**: it creates a secure binding between the GPRS Bearer Level Authentication (based on the parameters in the SIM/USIM) and the IMS layer. This is a proprietary solution and it has been proposed in the 3GPP Group for its standardization (early IMS). The advantage of this solution is that it is not necessary to enforce users to change their old SIM (as it can be used with a SIM too).

**IMS AKA User Authentication**: it is the only one actually standardized by 3GPP. It can be implemented both for ISIM and USIM:

- **IMS AKA User Authentication for ISIM**: the user card implements the ISIM with the relevant authentication parameters;
- **IMS AKA User Authentication for USIM**: the parameters used for authentication to the IMS domain are taken from the USIM.

Additionally, for fixed networks, network access authentication is considered (NASS Bundled authentication)

### 5.3.5 Charging

There are two models of charging supported in IMS:

- **Offline charging**: charging information is mainly collected after the session and users typically receive a bill on a monthly basis;
- **Online charging**: IMS entities interact with the online charging system that interacts in real time with the user’s account

Subsequently the Billing System (BS) creates the bill for the customer, including, for example, the following details: number of sessions, duration, destination, type of sessions.

### 5.3.6 Service opportunities

IP Multimedia Subsystem is a service infrastructure based on the use of Session Initiation Protocol (SIP). SIP allows the introduction of SIP-based User-to-User services through mobile phone, that is an important evolution after current client-server based services. Example of such services are:

- Content sharing applications (file sharing, video sharing, ...);
- Conferencing tools;
- Voice over IP applications;
- Video telephony;
- Peer-to-peer gaming

With IMS it is also possible to provide new client-server based services that are based on the SIP protocol feature. Some of these services are:

- Presence;
- Push to Talk over Cellular (PoC);
- Instant Messaging;
- Gaming server based.

Basically, the IMS Service Architecture is based on three alternative solutions: SIP Application Server, OSA Gateway and CAMEL SCP.

The main elements of the Service Architecture are:

- SIP AS: it is a programmable SIP Application Server;
- OSA SCS: it is one or more Service Capability Servers (SCSs) implementing one or more OSA Service Capability Features (SCFs) implemented by means of SIP;
- IM-SSF: it is a inter-working module between SIP and CAP protocols; from the point of view of CAMEL it implements a gsmSSF;
- CSE (CAMEL Service Environment): it is a CAMEL-enabled SCP implementing the gsmSCF.

The call/session server function S-CSCF provides a single service control protocol, named ISC (IMS Service Control interface). It is a SIP-based protocol that connects the call/session server function to the service platforms.

![Image: IMS - Service Architecture](image)

*Figure 5-7: IMS - Service Architecture*

Therefore, a service for IMS may be developed by exploiting one of the following service platforms:

- a SIP Application Server: the service is implemented according to the solutions provided by the SIP application server; examples are scripting languages (e.g., CPL), extension of CGI mechanism, and APIs (e.g., JAIN SIPlite, JAIN SIP, SIP Servlet, or proprietary solutions);
- an OSA Application Server: the service is implemented by exploiting the OSA APIs (Framework and SCFs), including the ones controlling the S-CSCF, and implemented
through SIP, such as MultiParty Call Control SCF or MultiMedia Cal Control SCF;

- a Camel-enabled SCP (i.e., CSE), by means of the related SCE tools.

Additional proprietary OSA SCFs could be deployed, e.g., to provide Instant Messaging capabilities.

5.4 Worldwide Interoperability for Microwave Access (WiMAX)

WiMAX (Worldwide interoperability for Microwave Access) is today the user-friendly name associated with IEEE 802.16a/REVd/ e standards. An overview of WiMAX is available in the main document at subsection 4.5.13.

5.4.1 The WiMAX Forum

The WiMAX forum (www.wimaxforum.org) is in charge of defining the profiles and certifying the interoperability of products and technologies in the broadband wireless access systems. The standardization process is at the basis of WiMAX Certified products: it will allow product interoperability and it will favour large scale economies.

The target is to achieve a cost effective broadband radio access solution by using advanced coding techniques, new NLOS technologies, a planned evolution to low speed mobility and to take benefit of interoperability and higher economies of scale through standardization.

<table>
<thead>
<tr>
<th>Definition</th>
<th>IEEE 802.16</th>
<th>IEEE 802.16a/REVd</th>
<th>WiMAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel conditions</td>
<td>LOS</td>
<td>NLOS</td>
<td>NLOS</td>
</tr>
<tr>
<td>Frequency bands</td>
<td>10 – 66 GHz</td>
<td>2 – 11 GHz</td>
<td></td>
</tr>
<tr>
<td>Modulation</td>
<td>QPSK, 16QAM and 64QAM</td>
<td>OFDM 256 sub-carriers, plus QPSK, 16QAM, 64QAM and OFDMA 2048 carriers</td>
<td>OFDM 256 sub-carriers</td>
</tr>
</tbody>
</table>

Figure 5-8: IEEE 802.16 and WiMAX

5.4.2 WiMAX performances

Performances declared by manufacturers for WiMAX systems are up to 70Mbps (with 64QAM modulation), with a channelization of 20MHz and in optimal conditions. The table below reports WiMAX gross bit rates, considering the channel assignments adopted mainly in American and European countries. It should be remembered that the WiMAX forum focuses on the WirelessMAN-OFDM Air Interface as its initial and primary interoperability mode.
In case of rollout of the products, performances will depend on:

- different characteristics of the systems;
- different working conditions (LOS/NLOS, indoor/outdoor, modulation, etc.);
- different profiles (in terms of frequency, channelization, duplexing).

Typical values that can be considered valid, taking into account of a channel of 7MHz, correspond to a capacity per cell of 17Mbps with 16QAM, and 9Mbps in 4QAM.

Dimensioning in terms of cell radius will be the output of all the hypothesis here above described, and the related values are reported in the following table.

<table>
<thead>
<tr>
<th>Modulation</th>
<th>Net bit rate Per sector for 3,5 and 7 MHz channel</th>
<th>Radius LOS At 3.5 and 7 MHz</th>
<th>Radius NLOS For outdoor “CPE” at 3,5 and a 7 MHz</th>
<th>Radius NLOS With “indoor” CPE at 3,5 e and at 7 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 QAM</td>
<td>2.3-&gt;4,5</td>
<td>10-&gt;9</td>
<td>2-&gt;1,8</td>
<td>0.4-&gt;0.35</td>
</tr>
<tr>
<td>16 QAM</td>
<td>4.6-&gt;9</td>
<td>7.5-&gt;6,5</td>
<td>1.5-&gt;1,3</td>
<td>0.3-&gt;0.25</td>
</tr>
<tr>
<td>64 QAM</td>
<td>9.2-&gt;18</td>
<td>6-&gt;5</td>
<td>1-&gt;0.8</td>
<td>0.25-&gt;0.2</td>
</tr>
</tbody>
</table>

The practical expected reference values are, for the systems working at 3.5GHz, could be considered as follows:

- in urban area the expected cell radius in NLOS conditions with outdoor antenna will be between 0.5 and 2 km, with indoor penetration between 0.15 and 1 km;
- in rural area the expected cell radius in NLOS conditions with outdoor antenna will be between 0.8 and 2.5 km, with indoor penetration between 0.3 and 1.5 km.

As described in the previous sections, mobility in WIMAX terminals will be introduced only
with 802.16 e compliant equipment. WiMAX mobility is forecast by using not only Mobile IP technology, but also in a manner similar to mobile handover. Nevertheless mechanisms to support layer 2 seamless continuity are up to now under study

5.4.3 WiMAX and services

Wi-Fi service
When the WiMAX user terminals were integrated in PCMCIA cards, it would be possible to adopt mobility solutions complementary to WLAN technology: in fact WiFi is a technology that has been designed on (short reach) private indoor applications, while WiMAX is focused on (longer reach) public outdoor applications.

Taking into account the state of the art, WiMAX is not foreseen to replace WiFi for private/home networking mainly for cost reasons and WiFi is not foreseen to compete with WiMAX in the public domain for performance reasons. Solutions are rather complementary and dual mode chipset for laptops have been announced.

WiMAX as a public outdoor solution can operate in licensed bands allowing for interference free transmission providing the QoS needed for some specific applications.

Broadband services
WiMAX is clearly a broadband data access solution. Even if voice can be carried over IP over the radio access solution, first standard WiMAX releases will be less optimized than those proven technologies to transport a majority of voice calls. Nevertheless, the growing broadband service penetration makes it attractive for operators to the upgrade their current proprietary LMDS or wireline solutions.

By means of Class of Services implemented in WiMAX systems, it is possible to support different traffic profiles for:

- SOHO (Small office/home office, e.g. providing of guaranteed bandwidth at medium rate for data connections; GPRS access to backbone);
- Business customer (e.g. providing of guaranteed bandwidth at high rate for data connections);
- corporate (LAN interconnection service; GPRS access to backbone).

Mobility services
When IEEE802.16e products will be available on the mass market, WiMAX user terminals can be seen as a complementary option for 3G mobile products (HSDPA over UMTS) for fixed “Wireless DSL” or nomadic “Portable Internet” applications.

Moreover the IEEE802.16e standard is far from being closed, so an effective evaluation of effective capabilities and comparison with 3G solutions has to be delayed for further study.
5.4.4 Interworking with the mobile network: short term scenario

In a short term scenario, the standard WIMAX products will be IEEE 80216d compliant, so:

- mobility feature shall not be supported;
- weight and size of Subscriber Stations will be comparable with the LMDS ones.

Therefore the mobile operator could adopt IEEE802.16 solution as IP connectivity solution to provide backhauling of different traffic present in the network operators, i.e.:

- GSM Abis (TDM-based);
- UMTS Iub (ATM-based);
- WiFi (IP-based)

The WIMAX backhauling scenario is an evolution of LMDS proprietary backhauling solution yet adopted by many mobile operators. Moreover the adoption of WiMAX compliant products allows improved coverage in NLOS and indoor conditions.

![Figure 5-10: WIMAX - WLAN-UMTS Interworking Architecture: short term scenario](image)

5.4.5 Interworking with the mobile network: long term scenario

In a long term scenario, it is possible to consider the introduction in the mass market of IEEE802.16e compliant WiMAX products. Different from the previous versions, these systems:

- will be enabled to support at least limited mobility with standardized inter-cell handover based on layer three mobile IP technology;
- will have dimensions comparable to the currently available 802.11 PCMCIA or could be integrated in the laptops.
In this scenario (see Figure 5-11) the WiMAX architecture could offer complementary solutions to UMTS (and WiFi if used). In this configuration the network operator could benefit to adopt WiMAX technology getting a maximum reuse of the existing mobile infrastructure in terms of radio site, backbone infrastructure, AAA and service servers.

Figure 5-11: WiMAX-WLAN-UMTS Interworking Architecture: long term scenario

5.5 Wireless Communication Protocols

An overview of Wireless Communication Protocols is available in the main document at subsection 4.5.14.

5.5.1 IP Versions - Transition tools

The IETF NGtrans working group has designed a set of IPv4-to-IPv6 transition tools to address the various needs of different networks. The two most basic building blocks of the toolbox are the hybrid stack mechanism and tunneling.

A hybrid stack host, also known as a dual stack host, implements both IPv4 and IPv6 — usually in a single stack in which most of the code is shared by the two protocols. The host “speaks” IPv4 with IPv4 peers, and IPv6 with IPv6 peers.
Tunneling provides a convenient way for an IPv6 island to connect to other IPv6 islands across IPv4 networks.

In addition some tools that attempt to automate the configuration of tunnels connecting isolated IPv6 networks over the IPv4 Internet has been developed: the tunnel broker (which aims to connect single hosts or small networks to the IPv6 Internet backbone) and the 6-to-4 mechanism.

In particular the 6-to-4 mechanism automatically derives a 48-bit IPv6 prefix from any global IPv4 address. It then spans stateless tunnels over the IPv4 infrastructure to connect to other 6-to-4 domains. The beauty of this mechanism is that a single global IPv4 address can bring IPv6 connectivity to an entire site. Adding IPv6 support and 6-to-4 functionality to IPv4 NAT boxes presents a very attractive transition scenario.

5.5.2 IP Versions in GSM/GPRS/EDGE/UMTS Standards

GSM/GPRS/EDGE/UMTS basic standards are foreseeing dual stack features, so de facto are independent from the IP version used, for both the internal network addressing and the
relevant aspect of the address provided to the user.

![Diagram](image1.png)

**Figure 5-14: Wireless Communication Protocols - Dual Stack UE connecting to IPv4 and IPv6 nodes**

IPv6 is capable of supporting the telecommunication network because it is quite mature and most of the new routers and of the most recent telecommunication equipments are IPv6 – ready, (even if some Rfc of the IP protocol suite for IPv4 are still not well developed for IPv6).

Meanwhile, IPv4 has demonstrated still being capable of supporting Internet growth. IPv4 is supported by routing algorithm and processing power that allow fallback to hierarchical addressing, and is also supported by NAT (Network Address Translation) technology which is mature and, to a certain extent, scalable. Port “natting” in NAT technology integrates the address “natting”.

The result is that today a telecommunications operator, in order to choose suitable IP technology, has to consider IPv4 and IPv6 strengths and weaknesses.

3GPP Mobile IMS standards are designed for the use of IPv6 (it shall be noted that the fixed IMS is only foreseen IPv4). De facto this leads to a complication that is unwanted and unnecessary for the first commercial IMS deployments, that almost all are based on IPv4 independently from a full compliance to the standard.

Most recent developments in 3GPP standards are going into the direction of removing this enforcement, in favour a full free dual stack option.

### 5.5.3 IP Versions in Mobile Telecom Networks

IPv6 is capable of supporting the telecommunication network because it is quite mature and most of the new routers and of the most recent telecommunication equipments are IPv6 – ready, (even if some Rfc of the IP protocol suite for IPv4 are still not well developed for IPv6). Meanwhile, IPv4 has demonstrated still being capable of supporting Internet growth. IPv4 is supported by routing algorithm and processing power that allow fallback to hierarchical addressing, and is also supported by NAT (Network Address Translation) technology which is mature and, to a certain extent, scalable. Port “natting” in NAT technology integrates the address “natting”.

The result is that today a telecommunications operator, in order to choose suitable IP technology, has to consider IPv4 and IPv6 strengths and weaknesses.
IPv4

- User behavior and configurations are homogeneous respect to Internet
- Full benefit from the Internet services and environment

IPv6

- Easy management of high numbers of customers

**Strength**

<table>
<thead>
<tr>
<th>IPv4</th>
<th>IPv6</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Needs to implement dynamic addressing and deploy NATs</td>
<td>• Has to implement interworking with IPv4 (NT-NAT)</td>
</tr>
<tr>
<td>• NATs are not compatible with some (few) Internet services (e.g. IPsec) so some kind of flexible addressing assignment is needed (some user may require static public addressing)</td>
<td>• NT-NATs are not compatible with some (few) IPv4 Internet services so flexible addressing assignment is needed</td>
</tr>
<tr>
<td>• Has to manage complications in case of very high number of customers (port natting may be required)</td>
<td>• May experience difficulties to benefit from some Internet originated services</td>
</tr>
<tr>
<td></td>
<td>• Difficulties in a full exploitation of the Internet services</td>
</tr>
</tbody>
</table>

**Weakness**

<table>
<thead>
<tr>
<th>IPv4</th>
<th>IPv6</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Needs to implement dynamic addressing and deploy NATs</td>
<td>• Has to implement interworking with IPv4 (NT-NAT)</td>
</tr>
<tr>
<td>• NATs are not compatible with some (few) Internet services (e.g. IPsec) so some kind of flexible addressing assignment is needed (some user may require static public addressing)</td>
<td>• NT-NATs are not compatible with some (few) IPv4 Internet services so flexible addressing assignment is needed</td>
</tr>
<tr>
<td>• Has to manage complications in case of very high number of customers (port natting may be required)</td>
<td>• May experience difficulties to benefit from some Internet originated services</td>
</tr>
<tr>
<td></td>
<td>• Difficulties in a full exploitation of the Internet services</td>
</tr>
</tbody>
</table>

As a result there are two aspects that have to be considered by operator to choose the most suitable IP technology, which are technology and user equipment:

- In terms of technology used to build the network, the use of IPv4 or IPv6 is a secondary aspect. Some benefits arrive from IPv6, but with some complications towards the user and the external networks

- In terms of user equipment addressing, the need to have a full interaction with the Internet as leverage to benefit of the Internet services, leads to a preference for IPv4, with some complications for very high numbers of customers.

At this state of development of the IP services, the results of such kind of analysis have led to the situation that almost all the mobile operators have preferred IPv4 for their infrastructure (with few notable exceptions like Japan, where IPv6 is enforced by the regulator), and, to avoid interworking problem with the “big Internet” for their customers also, with relevant developments of NAT.

Currently it is difficult to forecast when and if this situation will change, also taking into consideration that practically all the fixed networks have clearly indicated a medium-long time-frame trend in favour of IPv4.

Therefore is possible to summarize the trends as follows.

- At this stage is quite clear that Mobile networks and services will not lead a change from IPv4 to IPv6 due to the complications of the interworking with general offered Internet services.

- On the other side, if the general Internet will switch to IPv6, mobile networks will presumably follow immediately this switch, under that consideration that most of the mobile networks will shortly be IPv6 ready, and that the use of IPv6 simplifies the management and the processing of the IP addressing of the terminals.
5.6 Hand-Held Devices and User Terminals

An overview of Hand-Held Devices and User Terminals is available in the main document at subsection 4.5.4.

The most relevant software platform technologies and standards are outlined in the following.

5.6.1 J2ME

An overview of J2ME is available in the main document at subsection 4.5.4 - J2ME.

Functionality

J2ME has a “vertical” architecture being constituted from JVM and low-level APIs as a basis, forming a configuration and allowing extensions thanks to profiles that adds further functionalities.

Optional packages can be added to integrate even more features.

Choosing and combining the appropriate configuration, profiles and optional packages lets an optimized configuration be reached that fits specific hardware requirements for the device.

Device performances may vary significantly and the J2ME platform can make a full and optimized use of device-dependent features.

Configurations

Two configurations are described:

The Connected Limited Device Configurations (CLDC) and the Connected Device Configuration (CDC).

Connected Limited Device Configurations (CLDC) is designed for low-memory, slow-processors devices with a weak and slow connectivity (typically wireless). Usually such devices are smartphones and low-end PDAs with a 16 bit processors and 128-512 KB memory for the Java platform and applications.

CLDC is built on top of KVM (also known as the K Virtual Machine) a compact, portable Java virtual machine designed for small, resource-limited devices such as smartphones, personal organizers, mobile Internet devices, etc.

Connected Device Configurations (CDC) is designed for high-memory, fast-processors devices with a reliable and fast connection. It is based on a complete JVM with an almost complete set of J2SE APIs (Java 2 Standard Edition is the Java platform for desktop systems).

Typical devices have at least 2MB of memory for the Java platform and application and a 32 bit processor.

Profiles

A high-level set of APIs has to be introduced to built a complete environment for the execution of applications. These high-level sets of APIs are called profiles.
Profiles provides a model for the lifecycle of an application, a model for generating the user interfaces and modes used to access device-specific properties and features.

*The Mobile Information Device Profile (MIDP)* is designed for entry level PDAs and smartphones and offer functionalities for:

- Design of the user interface
- Connectivity to the network
- Local storage of data
- Application management

When combined with the Connected Limited Device Configuration (CLDC), MIDP provides a standard Java runtime environment for popular mobile information devices, smartphones and PDAs. CLDC and MIDP provide the core application functionality required by applications in a mobile environment. Using MIDP a developer can write applications once, then deploy them easily to a various information devices.

Recent features of MIDP (MIDP 2.0) include an enhanced user interface, multimedia and game functionality, more extensive connectivity, over-the-air provisioning (OTA), and end-to-end security.

MIDP give access to the native messaging capabilities and data network of mobile information devices supporting connectivity standards such as HTTP and HTTPS. MIDP also supports the Short Message Service and Cell Broadcast Service capabilities of GSM and CDMA networks, through an optional Package: the Wireless Messaging API (WMA)

*The Foundation Profile (FP)* is the "lowest level" among those defined for CDC.

It does provide network connection facilities but doesn’t have methods to create a graphical user interface.

*Personal Profile (PP)* is the profile for CDC devices that needs to have a complete graphical user interface and support for applets.

*Personal Basis Profile (PBP)* is a subset of the previous profile providing an environment for devices connected to the network that needs some basics functionalities for graphic-based presentations and user interfaces

*Layering*

CDC and CLDC profiles are organized as a layered structure so that a new functionality can be added by simply providing the support for other profiles implementing those functionalities. Furthermore CDC and CLDC support a number of optional packages that allow a balanced set of functionality to be built satisfying the needs against possible resource constraints.

Optional Packages have been created to satisfy the needs that arise with emerging tech-
Technologies that were not present at the time when the first J2ME environment was designed. These Packages have, in turn, a modular structure so that only the necessary parts of the packages can be included.

CLDC-based optional packages for example includes the Mobile Media API (MMAPI) and Wireless Messaging API (WMA)

The Mobile Media API (MMAPI), extends the functionalities of the J2ME platform with audio, video, and other multimedia support to devices which may have limited resources. For that reason it is an optional component and has been designed to be lightweight.

It is mainly used on devices that implement profiles based on the Connected Limited Device Configuration (CLDC)

The Wireless Messaging API (WMA) is an optional package for J2ME providing platform-independent access to wireless communication resources such as Short Message Service (SMS) and Cell Broadcast Service. WMA is typically used on top of CLDC and MIDP.

![Diagram of J2ME Layers](image)

Figure 5-16: J2ME - Layers for the Java 2 Platform Micro Edition

**Interface**

The J2ME layers uses different sets of APIs that can be used by other layers (such as WMA used on top of CLDC) or by the developer (for “top level” layers).
5.6.2 SyncML

An overview of SyncML is available in the main document at subsection 4.5.4 - SyncML.

**Functionality**

SyncML main components are:

- Representation Protocol
- Synchronization Protocol
- Transport Bindings

*The SyncML Representation Protocol* specifies a format used to establish different models for data synchronization. Synchronization is defined as an exchange of packages, where a package is a set of one or more messages. The messages defined by the protocol (either XML documents or MIME content type) have three DTDs for the high-level representation of SyncML messages and commands, for the definition of a format for meta-information, and for the definition of a format for information about the device and its capabilities. A message consists of a header and a body. The header defines session, routing and security information. The body has a list of commands.

*The SyncML synchronization protocol* defines the sequence of packages that has to be exchanged to allow synchronization; each package has one or more messages. The protocol defines two roles: client and server. Both one- and two-way synchronization are supported.

*SyncML Transport Bindings* ensures interoperability for SyncML messages. Transport bindings being defined in SyncML 1.0 are HTTP (widely used), WSP (Wireless Session Protocol), and OBEX (Object Exchange Protocol for short range connectivity such as IrDA and Bluetooth).

**Interface**

SyncML is a XML based protocol family. It uses binary XML coding reducing the amount of data that has to be transferred.

It works over leading network protocols: TCP/IP, HTTP, Wireless Session Protocol (WSP), and OBEX (Bluetooth, IrDA). It can be deployed over SMTP, POP3, IMAP, and proprietary wireless communication protocols.
END OF DOCUMENT