

CEN/TC 278

Date: 2025-10

prTS OPRA:2025

Secretariat: NEN

Public transport — Operating raw data and statistics exchange

Einführendes Element — Haupt-Element — Ergänzendes Element

Élément introductif — Élément central — Élément complémentaire

CCMC will prepare and attach the official title page.

Contents	Page
1	Scope..... 12
2	Normative references..... 12
3	Terms and definitions..... 13
4	Symbols and Abbreviations 13
5	Compatibility with Existing Standards..... 14
5.1	Compatibility with Transmodel..... 14
5.1.1	General..... 14
5.1.2	Data structure representation 14
5.1.2.1	Generic Loggable Object model 14
5.2	Compatibility with EN 13816:2002 — Service Quality Definition, Targeting and Measurement..... 15
5.3	Compatibility with EN 15140:2006 — Basic requirements and recommendations for systems that measure delivered service quality 17
6	Links to regulations 17
6.1	Data required by the MMTIS regulation 17
6.1.1	Observed Data..... 18
6.1.2	Historic Data 18
6.1.3	Key Performance Indicator 18
6.1.4	Indicator..... 18
7	Types of Reference Data Used in OpRa 19
7.1	General..... 19
7.2	Identification of Objects beyond system borders 19
7.2.1	General..... 19
7.2.2	Object identifiers and namespaces 20
7.3	Participant (information provider) identification 20
7.3.1	General..... 20
7.3.2	Model..... 21
7.3.3	Namespace and Scope 21
7.4	Participant pair identification (service participant pair code) 21
7.4.1	General..... 21
7.4.2	Model..... 21
7.4.3	Namespace and Scope 21
7.5	Data formats 21
7.5.1	Date & time..... 21
7.5.2	Time Zones 21
7.5.3	Georeferences 21
7.5.4	National language of text elements 22
7.5.4.1	Two character ISO Language code. 23
8	Use cases 23
8.1	General..... 23
8.2	Purpose..... 23
8.3	Actors and use case categories..... 23
8.3.1	Actors 23
8.3.2	Use case categories..... 23

8.3.3	Collection of use cases.....	25
8.3.3.1	General.....	25
8.4	Prioritisation of Use Cases for Modelling.....	29
8.4.1	Service Offer	30
8.4.2	Service Demand.....	32
8.4.3	Service Efficiency	35
9	Conceptual Data Model.....	39
9.1	Concepts introduced in OpRa.....	39
9.1.1	Data preparedness.....	39
9.1.1.1	Concept overview.....	40
9.1.1.2	Preparedness level raw	41
9.1.1.3	Preparedness level cleansed	41
9.1.1.4	Preparedness level processed	42
9.1.2	Statistics.....	42
9.1.2.1	Distribution analysis support	42
9.1.2.2	Distribution of duration type indicators	43
9.2	UML models	43
9.3	Modelling Explicit Frames	44
9.4	Modelling Generic Information and Statistics.....	45
9.5	Use case EFFICIENCY-011: Number of late journeys.....	46
9.5.1	Use case modelling.....	46
9.6	Use case EFFICIENCY-014: Number of cancelled journeys	50
9.6.1	Use case modelling.....	50
9.7	Use case DEMAND-001-001: Expected number of passengers.....	54
9.7.1	Use case modelling.....	54
10	Physical Data Model.....	55
10.1	Introduction	55
10.2	Model Driven Design	55
10.3	Open Implementation and Technology Use	56
10.4	Summary of Modelling Approach	56
10.5	Common OpRa Framework.....	56
10.6	Implementing Transmodel framework features in OpRa	57
10.7	Versions & Validity.....	58
10.7.1	Introduction	58
10.7.2	Version & Validity – Model Dependencies.....	58
10.7.2.1	Generic ENTITY – Conceptual MODEL	59
10.7.2.2	Generic Entity – Physical Model.....	60
10.7.2.2.1	General.....	60
10.7.2.2.2	Codespace – Physical Model.....	61
10.7.2.3	Generic Entity – Attributes and XSD.....	62
10.7.2.3.1	Entity – Model Element.....	62
10.7.2.3.2	Codespace – Model Element	62
10.7.2.4	TypeOfEntity – Model Element.....	64
10.7.2.4.1	ClassInRepository – Model Element.....	65
10.7.2.5	XML Examples for generic Entities	65

10.7.2.5.1	Codespaces & Identifiers – XML Example Fragment.....	65
10.7.3	Generic Version	69
10.7.3.1	General.....	69
10.7.3.2	Generic VERSION – Conceptual MODEL.....	69
10.7.3.3	Generic Version – Physical Model.....	70
10.7.3.3.1	General	70
10.7.3.3.2	Naming Entities & Entities in Version in the OpRa Physical Model.....	71
10.7.3.3.3	Version States and the Edit / Version / Release Cycle	72
10.7.3.4	Version – Attributes and XSD	72
10.7.3.4.1	Version – Model Element.....	72
10.7.3.4.2	EntityInVersion – Model Element.....	73
10.7.3.4.3	EntityInVersionReferences (VersionOfObjectRef)– Model Element.....	76
10.7.3.4.4	Entity_Entity – Model Element.....	77
10.7.3.4.5	TypeOfVersion– Model Element.....	79
10.7.3.5	XML Examples of use of Versions	80
10.7.3.5.1	Version Instances.....	80
10.7.3.5.2	Including Version Instances	80
10.7.3.5.3	Use of an Entity with Versions of that Entity In Version – XML Example fragment 81	
10.7.4	Implementing relationships in NeTeX	82
10.7.4.1	Entity identifiers and uniqueness.....	82
10.7.4.2	Associations and Entity References.....	82
10.7.4.3	Relationship metaclasses	84
10.7.4.4	XML Examples of Relationships.....	85
10.7.4.4.1	Use of versions & Version references with Integrity checking – XML fragment ...	86
10.7.4.4.2	Use of dummy versions & references with Integrity checking – XML fragment ...	86
10.7.4.4.3	Use of unversioned references – XML fragment	87
10.7.4.4.4	Use of version references without integrity checking – XML fragment.....	87
10.7.5	Generic Version Frame	88
10.7.5.1	Generic VERSION FRAME – Conceptual MODEL	88
10.7.5.1.1	Types of Frame – Conceptual MODEL	89
10.7.5.1.2	Version Frames and Versioning.....	90
10.7.5.1.3	General Frames and Explicit Frames	91
10.7.5.1.4	Using Version Frames to Requests	92
10.7.5.1.5	Types of Version frame.....	93
10.7.5.2	Generic Version Frame – Physical Model	93
10.7.5.2.1	Version Frame – Physical Model.....	93
10.7.5.2.2	Type of Version Frame – Physical Model.....	94
10.7.5.2.3	Profile Model – Physical Model.....	95

10.7.5.3	Version Frame – Attributes and XSD.....	96
10.7.5.3.1	VersionFrame – Model Element.....	96
10.7.5.3.2	EntityInVersionInVersionFrame – Model Element.....	100
10.7.5.3.3	TypeOfFrame – Model Element.....	100
10.7.5.3.4	TypeOfValidity – Model Element.....	102
10.7.5.3.5	DataNature – Allowed Values.....	103
10.7.5.3.6	ClassInFrame– Model Element.....	103
10.7.5.3.7	ClassAttributeInFrame– Model Element.....	105
10.7.5.3.8	ClassRelationshipInFrame– Model Element.....	106
10.7.5.3.9	Class (ClassRef) – Model Element.....	107
10.7.5.3.10	Relationship (RelationshipRef) – Model Element.....	107
10.7.5.4	XML Examples of Version Frames.....	108
10.7.5.4.1	Simple Version Frame – XML fragment.....	108
10.7.5.4.2	Use of Type Of Frame – XML fragment.....	109
10.8	Responsibility.....	112
10.8.1	Introduction.....	112
10.8.2	Responsibility – Model Dependencies.....	112
10.8.3	Generic Responsibility.....	113
10.8.3.1	Generic Responsibility – Physical Model.....	114
10.8.3.1.1	Summary of Responsibility model.....	114
10.9	Operating Days.....	115
10.10	Notation of XML-Elements and XML-Structures.....	116
10.10.1	General.....	116
10.10.2	Display of XML Elements in the Text.....	116
10.10.3	Display of Relationships.....	116
10.10.4	Table Notation of XML Structures.....	117
10.10.4.1	General.....	117
10.10.4.2	Grouping.....	118
10.10.4.3	Element Name.....	119
10.10.4.4	Multiplicity & Choice (min:max).....	119
10.10.4.5	Data Type.....	119
10.10.4.6	Explanation.....	119
10.10.4.7	Message Exchange.....	120
10.10.5	Use of SURI Procedure.....	120
10.10.6	HTTP and REST.....	121
10.10.7	Roles of Server and Client.....	122
10.11	Service Discovery.....	122
11	Services and XML Schemas.....	122
11.1	General.....	122
11.2	Services Provided.....	122
11.3	XML Schemas Used Across Services.....	122
11.4	Imported Schemas.....	123
11.5	Problems and Error States When Operating OpRa Services.....	123

11.5.1	Error Codes from SIRI	123
11.5.2	OpRa ErrorCondition.....	124
11.5.3	General OpRa Problems	125
12	Common XML Structures.....	125
12.1.1	General.....	125
12.2	Root Element OpRa	125
12.2.1	General.....	125
12.2.2	Service Request Structure	126
12.2.3	Frames	126
12.3	Generic XML Issues.....	128
12.3.1	Indicator-related base definitions.....	128
12.3.2	Generic scope for indicators	129
12.3.3	Specific scopes for indicators	130
12.4	OpRa Service Capabilities	130
12.4.1	Service Capabilities Modelling Approach	131
12.4.1.1	Generic OpRa Payload Delivery.....	132
12.5	Sample network used in use cases	133
12.6	FRAMEs within the exchange.....	133
12.6.1	PLANNED SERVICE FRAME.....	133
12.6.1.1	Requesting.....	133
12.6.2	ACTUAL SERVICE FRAME.....	135
12.6.2.1	Requesting.....	135
12.6.3	RAW DATA FRAME.....	136
12.6.3.1	Raw Data Frame PreparednessLevel.....	139
12.6.4	KPI FRAME.....	140
13	Worked Used Cases	140
13.1	EFFICIENCY-011: Number of late journeys.....	140
13.1.1	Discovery	141
13.1.2	Requesting.....	141
13.1.3	Payload	142
13.1.4	Example	143
13.2	EFFICIENCY-014: Number of cancelled journeys.....	144
13.2.1	Discovery	145
13.2.2	Requesting.....	146
13.2.3	Payload	147
13.2.4	Example	148
13.3	Vehicle load related use cases.....	148
13.3.1	DEMAND-001-001: Expected number of passengers	151
13.3.1.1	Discovery	151
13.3.1.2	Requesting.....	151
13.3.1.3	Payload.....	152
13.3.1.4	Example	153
13.3.2	DEMAND-002-001: Measured number of passengers.....	155
13.3.2.1	Discovery	156
13.3.2.2	Requesting.....	157
13.3.2.3	Payload.....	159

13.3.2.4 Example..... 162

13.4 Generic KPI Example 163

European foreword

This document (prTR OPRA:2025) has been prepared by Technical Committee CEN/TC 278 “Intelligent transport systems”, the secretariat of which is held by NEN.

This document is currently submitted to the Vote on TS.

This document will supersede TR 17370:2019.

This document has been prepared under a standardization request addressed to CEN by the European Commission. The Standing Committee of the EFTA States subsequently approves these requests for its Member States.

For the relationship with EU Legislation, see informative Annex ZA, which is an integral part of this document.

Introduction

Status of Document

This document has been produced for technical review through the Committee Internal Balloting process.

It is acknowledged that the document is not complete, most notably, additional use cases need to be completed. The working group requests the review of the parts included in this version to ensure that the documented approach can be supported by the CEN community before further work to complete the document is undertaken.

General

The Public Transport (PT) sector, particularly for Public Transport Operators (PTO) and Public Transport Authorities (PTA), has identified the necessity to develop a data strategy, based on cost and value, focusing on the strategic benefits of data. It is crucial that mobility stakeholders are not only able to compete against potential new market entrants but also benefit from a suitable environment to develop new business models and services. Public transport is becoming a data-enabled or data-driven business and needs to answer different local conditions.

This led to the need to standardize data analysis in Public Transport, to understand formally how information can be created as added value, and indicator calculation can facilitate moving from a qualitative analysis of the PT service towards a quantitative one.

With this perspective in mind, OpRa (Operating raw data and statistics exchange) defines a minimum set of Public Transport raw data needed as PT quantitative analysis enabling factor.

Transport modes

OpRa takes most public transport modes, except air transport, into account. This specifically includes rail, bus, metro, tramway, trolleybus, ferry, coach, funicular railway, suspension railway, and rack railway.

OpRa management information

Management information deals with functions analysing production data, in order to evaluate the service quality or to take corrective measures in planning and managing operations. In PT, for instance, the study of operational data (e.g. observed run times, passenger load) collected during service operations is an input for strategic planning (e.g. how and when to amend the schedules), tactical planning (e.g. when to undertake a certain control action), quality follow-up, etc.

Management information uses, therefore, two main types of data:

- data resulting from the planning stages, i.e. theoretical data on the production orders (e.g. timetables, run times, driver rosters, etc.);
- data describing the daily actual production (e.g. observed passing times, actual number of passengers, missed interchanges, modifications operated to the plan, etc.).

Advances in technology, in particular high density data storage, allow the creation and retention of all the necessary underlying data in production databases, against which the desired requests can be made by PT managers. Consistent data structures for management data will make the design of such requests easier, and offer the flexibility required to OpRa exchange PT raw data, enabling indicator calculation.

Raw data is identified according to specific use cases and, where applicable, defined and described in Transmodel (EN 12896) compliant structures. In this perspective, the exchange will be performed using similar NeTEx modalities, for compatibility reasons.

OpRa exchanging data modality

Similar to NeTEx, the primary software resource for implementing an OpRa interface is based on XML schema.

Two main variants of the schema are available, each providing a different protocol for embedding the same content model subschemas.

- Simple OpRa documents exchange (see NeTEx _publication.xsd): a schema to use with NeTEx documents input or output by a system that are exchanged as files using FTP, email etc.
- OpRa document exchange using SIRI HTTP requests (see NeTEx_siri_SG.xsd): a schema that embeds the OpRa elements in a sequence of HTTP messages that define request/response and publish/subscribe interchanges for exchanging data. Requests use OpRa elements to specify the desired data. Responses wrap in version frames. The messages are specialisations of the SIRI framework.

Motivation

Measured Public Transport data describing the public transport network fulfilment are essential for studies, control, service improvement and contractual relations between stakeholders. It is important that they can be shared among PTO, PTA, engineering and design office, researchers, and other actors in a clear and unambiguous way, to provide accurate and intelligible information.

Furthermore, the OpRa covered scope is fully complementary to other existing exchange standards and allows covering one of the remaining gaps of public transport standardisation, with references to Transmodel, NeTEx and SIRI.

NeTEx (CEN TS 16614-1/-2/-3) is an exchange protocol dedicated to scheduled public transport data, based on the Transmodel (EN 12896) conceptual data model. NeTEx supports the exchange of information relevant to public transport services for passenger information and automated vehicle management systems (AVMSs) and is divided into three parts:

- Part 1: network topology exchange;
- Part 2: timetables exchange;
- Part 3: fare information exchange.

SIRI (EN 15531-1 to 5) is complementary with NeTEx and provides operators and manufacturers with a standard framework for exchanging data concerning public transport real-time information, along with a set of functional services for specific types of real-time data. As for NeTEx, the underlying conceptual definitions used by SIRI are provided by Transmodel. SIRI and NeTEx share the same communication protocol.

Using these standards, scheduled and real-time information can be made available for passenger information, and for operations and for process review. However, there is still one final aspect missing, just after real-time: to provide information about what has been performed, with the same overall view as for scheduled information. Such data allows a feedback loop to improve existing services.

The OpRa standard is therefore about operating raw data and statistics, regularly requested by PTA, aims to meet that need.

CEN Standards context

OpRa has been developed under the aegis of CEN and draws on several existing national Public Transport Service scenarios and EU standards.

The keystone is the Transmodel standard, a conceptual model that names and represents PT info concepts for a wide set of functional areas and can be used to compare and understand different models. Transmodel project outputs have been used both to underpin several CEN concrete data standards such as NeTEx, SIRI or IFOPT¹. It underpins many national standards to allow for harmonization and interoperability. Transmodel generic model has been used to develop OpRa and OpRa-specific requirements itself being updated to include OpRa additions provided also some of Transmodel enhancements (present in prEN 12896-8:2025, concerning Management Information and Statistics).

¹ IFOPT has been included in Transmodel Part 2: Public Transport Network (EN 12896-2) and is no more a standalone standard.

1 Scope

The OpRa scope is the definition of a minimum set of Public Transport raw data needed to enable PT quantitative analysis.

OpRa does not go into the domain of service quality measurement and reporting: service quality analysis will use data provided by OpRa, but quality definition remains a contractual level issue between a Public Transport Authority and a Public Transport Operator or an operator's internal choice for a purely private service. OpRa reports unbiased actual data (i.e. measured or observed), described and aggregated in a shared and understandable way.

OpRa focuses mainly on observed/measured and registered information, i.e. information which cannot be changed anymore in the future. This information is mainly an output of the domain "operations monitoring & control" as defined by Transmodel.

Based on the actual service, including delays and cancelled vehicle journeys, etc., OpRa considers the recorded reality of operation, either through individual observed and registered data at a given sampling interval (raw data) or in an aggregate way (indicators) and provides the data exchange format.

The approach documented in detail in this document is coherent with EU Directive 2010/40. It relates to the Article 4 of the Delegated Regulation EU 2017/1926 [16], as regards the historic data. OpRa proposes to complement NeTEx (dedicated to the static scheduled information), for the historic data, based on the underlying conceptual data reference model Transmodel EN 12896, similarly to the requirement of the Delegated Regulation EU 2017/1926 referring to the static scheduled information².

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

EN 12896 (all parts), Public transport. Reference data model

EN 16614 (all parts), Public transport. Network and Timetable Exchange (NeTEx)

EN 15531 (all parts), Public transport. Service interface for real-time information relating to public transport operations

EN 13816:2002, Transportation. Logistics and services. Public passenger transport. Service Quality Definition, Targeting and Measurement

EN 15140:2006, Public passenger transport. Basic requirements and recommendations for systems that measure delivered service quality

ISO 8601, Date and time — Representations for information interchange

ISO 639, Code for individual languages and language groups

² For what concerns the exchange of static scheduled data (such as public transport, long distance coach and maritime including ferry), the relevant data in the national access point should use the CEN data exchange standard NeTEx CEN TS 16614 based on the underlying conceptual data reference model Transmodel EN 12896:2006 and subsequent upgraded version.

3 Terms and definitions

For the purposes of this document, the terms and definitions given in in prEN 12896-8 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp/>
- IEC Electropedia: available at <https://www.electropedia.org/>

4 Symbols and Abbreviations

The common symbols and abbreviations used in this document set are presented in EN 12896. In addition, the following terms are used:

API	Application programming interface
AVL	Automated vehicle location
AVLS	AVL system
AVMS	Automated vehicle management system
CEN	Comité Européen de Normalization
ERA	European Union Agency for Railways
IFOPT	Identification of fixed objects in public transport
ISO	International Standards Organization
KPI	Key performance indicator
MI	Management Information
MMTIS	multimodal travel information services
MTBF	Mean Time Between Failure
NeTEx	Network and Timetable Exchange
OJP	Open Journey Planner
OpRa	Operating raw data and statistic exchange
PT	Public transport
PTA	Public transport authority
PTO	Public transport operator
QoS	Quality of service
SIRI	Service Interface for Real-time Information
TR	Technical Report
UC	Use case
UIC	International Union of Railways
UITP	International Association of Public Transport
UML	Unified Modelling Language
UPT	Urban Public Transport

WGS	World Geodetic Standard
WSDL	Web Services Description Language
XMI	XML Metadata Interchange
XML	Extensible Markup Language
XSD	XML schema document

5 Compatibility with Existing Standards

5.1 Compatibility with Transmodel

5.1.1 General

As previously mentioned, OpRa is strongly related to the EN 12896-8:2019, Public Transport – Reference Data Model – Part 8: Management Information and Statistics, that mainly describes how to structure data which either refers:

- to the planning stages (e.g. timetables, run times, driver rosters, etc.), and/or
- to the daily actual production, which is registered for different purposes, in particular to build indicators and for the provision of different types of statistics.

In Transmodel the term indicator means a set of data (calculated or measured) which may be either qualitative or quantitative that is used to provide information on the status (may be a measure, a functional state, etc.) or the quality of a service or a function.

In particular, EN 12896 8, Public Transport – Reference Data Model – Part 8: Management Information and Statistics is about data semantics and structuring of raw data used to provide indicators, meanwhile OpRa is about the different types of indicators, each specialized for reporting information on a specific aspect of Public Transport.

5.1.2 Data structure representation

5.1.2.1 Generic Loggable Object model

According to the type of indicator (use case), a different granularity of data and different data types are recorded. However, the mechanism for creating records and the associated information in view of the provision of indicators is similar in all cases.

Therefore, rather than describing multiple data structures for each use case, Transmodel provides a generic structure (a design pattern) that shall be used to represent the changes of state of objects, their recording and representing indicators (Figure 2).

Transmodel provides some examples to explain how this generic pattern can be applied to concrete use cases.

ENTITY is defined in Transmodel as any data instance to be managed in an operational version management system. When several data sources coexist, an ENTITY must be related to a given DATA SOURCE in which it is defined.

LOGGABLE OBJECT is considered as a TRANSMODEL VIEW, i.e. it represents selected properties of one or more ENTITIES being monitored/observed for significant events and changes of state.

Successive states and events affecting any observed ENTITY are recorded as a set of LOG ENTRIES, making up a temporal trace that can be analysed for management purposes or examined for audit purposes.

The LOG ENTRIES are triggered by EVENTS occurring in the lifecycle of an object and may be grouped together in any kind of storage to form a LOG.

Successive states affecting any observed ENTITY are recorded as a set of LOG ENTRIES, making up a temporal trace that can be analysed for management purposes or examined for audit purposes.

The LOG ENTRIES are triggered by EVENTS. The EVENT itself is not necessarily represented in the implementation system, only the LOG ENTRY capturing the salient aspects of the event.

When an EVENT has spatial attributes, it becomes a LOCATED EVENT and has geographical attributes that can refer to:

- the ZONE in which the LOCATED EVENT is located;
- the POINT where the LOCATED EVENT occurs;
- the LINK on which the LOCATED EVENT is located.

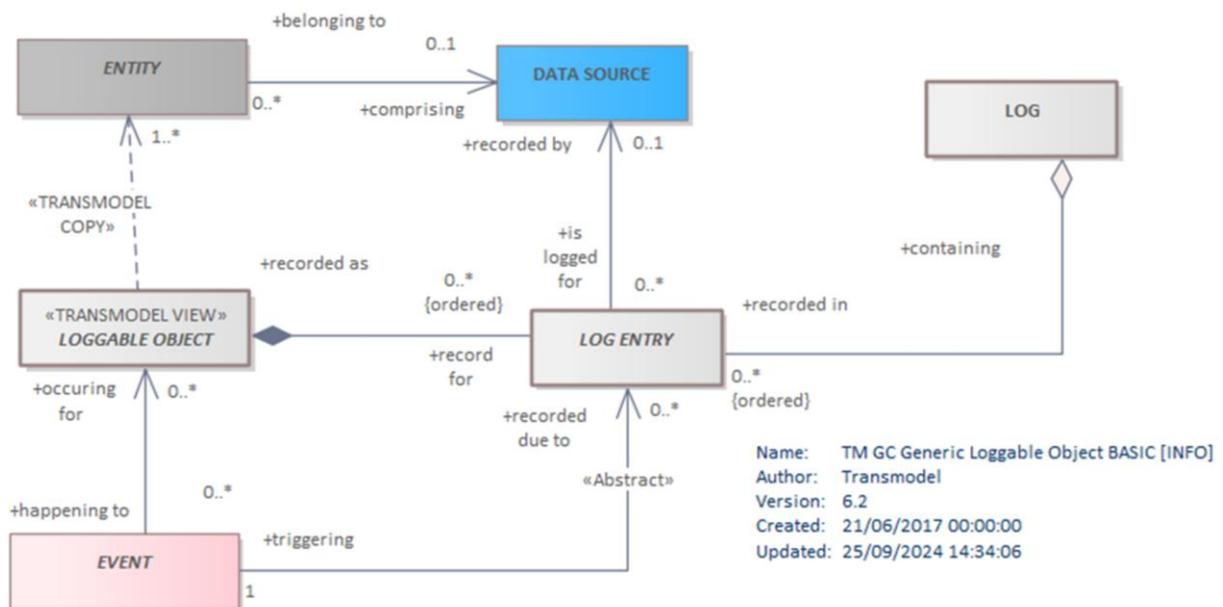


Figure 1 — Loggable Object model (source Transmodel Part 1)

5.2 Compatibility with EN 13816:2002 — Service Quality Definition, Targeting and Measurement

EN 13816:2002 is a specific type of service evaluation standard that establishes a framework for defining, selecting the appropriate measurement method and measuring the quality of service in public passenger transport.

The standard introduces:

- **Eight “first level” quality criteria:** Availability, Accessibility, Information, Time, Comfort, Customer Care, Security and Environment.
- **Further second and third level sub-criteria** corresponding to variables that affect the quality of the service.

The standard does not set specific quality requirements and leaves the organization free to set the appropriate level of service. The definition of the quality of service offered needs to consider a list of

criteria covering all the first level criteria, and selecting second and third level sub-criteria according to the needs.

It also recognized that the quality of urban public transport (UPT) was dependent on a number of factors like the capacity of the operator to manage his organization, operating conditions, role of the PTAs (roads department, traffic police, etc.).

Quattro’s research developed a specific quality management tool, “the UPT quality loop” (Figure 3), which can be applied at the firm’s level as well as at the whole UPT system’s level.

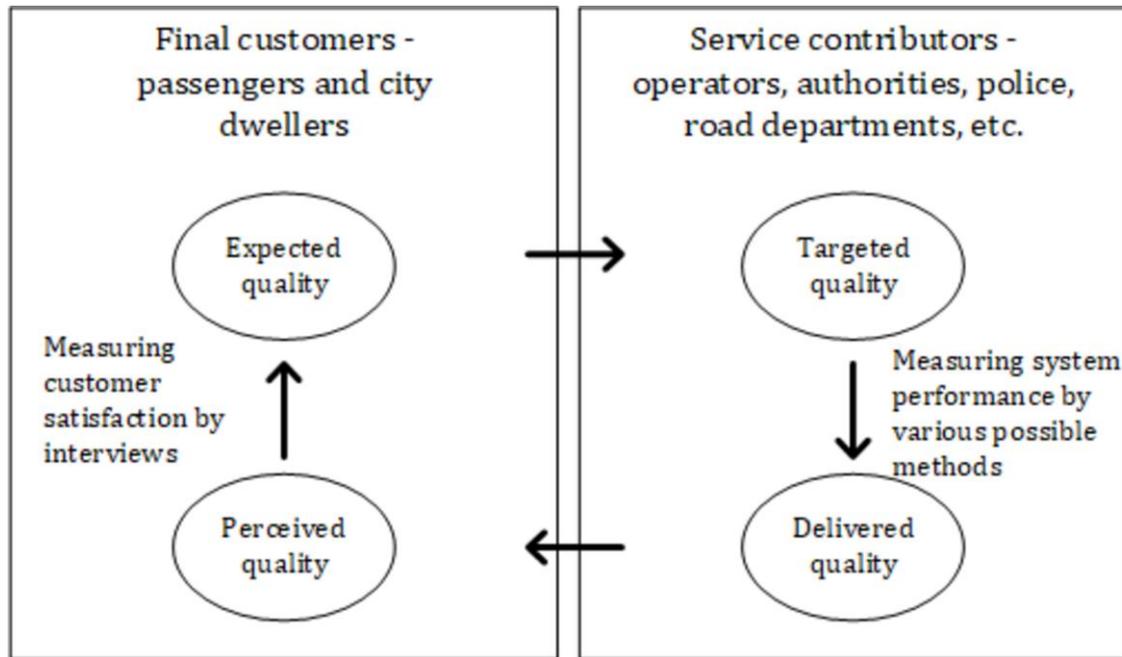


Figure 2 — The Urban Public Transport Quality Loop. Source: QUATTRO taken from TCRP 2010

The report describes the quality loop to be based on four distinctive benchmarks:

- a) **Expected Quality:** This is the level of quality demanded by the customer. It can be defined in explicit and implicit expectations. Tools for evaluation: revealed and stated preference methods.
- b) **Target Quality:** This is the level of quality that the transport undertaking aims to provide for its passengers. It should be defined according to the level of quality expected by the passengers, external and internal pressures, and budgetary constraints and competitor/market performance. Tools for evaluation: customer charters and guarantees of service, partnership agreements, quality standards and certification, quality contracts, quality tenders and evaluation procedures, etc.
- c) **Delivered Quality:** This is the level of quality achieved on a day-to-day basis in normal operating conditions. Disruptions to service, whether they are the fault of the undertaking or not, are considered. Tools for evaluation: compensation schemes for the benefit of the users, reward/penalty schemes concerning operators and authorities, internal quality measurement, self-assessment methods and benchmarking (KPIs).
- d) **Perceived Quality:** This is the level of quality perceived by passengers during their journeys. Tools for evaluation: customer satisfaction index (CSI), customer-charter feedback systems.

From the above, it is evident there are many approaches that operators and public authorities can adopt to improve public transport quality. For performance measures, each operator/authority should define a target level of reference taking into account in particular the number of passengers affected.

5.3 Compatibility with EN 15140:2006 — Basic requirements and recommendations for systems that measure delivered service quality

EN 15140:2006 supports the development of a (robust) performance measurement system with reference to what indicated in EN 13816.

With respect to data collection methods, it introduces the concepts (and related requirements) of:

- surveyors;
- technical measurements;
- continuous measures;
- sampling measures.

It proposes some recommendations such as:

- refer to good practices applied to similar contexts;
- selection of the criteria to be measured should be based on surveys of users' expectations.

The number of criteria to be measured needs to be large enough to cover the essential criteria (first level) but limited by the ability to manage the data collection of these criteria. It is recommended to report the measures to the number of users of the service.

6 Links to regulations

The development of the OpRa standard has been largely driven by the requirements of the European Commission Delegated Regulation (EU) 2017/1926 supplementing the ITS Directive with regard to the provision of EU-wide multimodal travel information services (MMTIS) amended by Delegated Regulation (EU) 2024/490.

It establishes the specifications needed to ensure the accessibility, exchange and update of standardised travel and traffic data, as well as for distributed journey planning within the EU.

The MMTIS Delegated Regulation aims to create appropriate conditions for cooperation of all the relevant stakeholders along the travel information value chain, including transport authorities, transport operators, travel information service providers, infrastructure managers and transport on demand service providers among others.

The 2024 revision added new data types in an Annex, including static, historic and observed travel and traffic data types which are the focus of this OpRa standard.

6.1 Data required by the MMTIS regulation

The MMTIS describes four key types of data which need to be provided: static, historic, observed and dynamic travel and traffic data.

Of these, historic and observed are defined within the scope of this OpRa standard for public transport related data. Static data is described by TS 16614 – Public transport – Network and Timetable Exchange (NeTEx) format and dynamic data by EN 15531 – Public transport – Service interface for real-time information relating to public transport operations both of which are referenced within this document where required.

To provide the required historic and observed data a combination of planned, real time and raw data indicators are used.

6.1.1 Observed Data

Observed data is empirical data collected through the direct measurement or monitoring of public transport operations, passenger behaviour, vehicle performance, and related infrastructure elements. This includes data recorded from sensors, automated systems, manual counts, fare transactions, GPS tracking, and environmental monitoring tools. Observed data reflects real-world conditions without predictive modelling.

Observed data is gathered by observing and recording events as they occur, without any manipulation or intervention of the data observed. This data is likely to be classed as raw data as described in section 9.1.1. Data preparedness.

6.1.2 Historic Data

Historic data is derived from previously recorded observations of public transport operations, passenger usage patterns, vehicle movements, and infrastructure performance over defined past time periods. Historic data includes validated datasets such as schedule adherence logs, fare transaction records, passenger counts, environmental conditions, and operational metrics. It serves as a basis for trend analysis, benchmarking, forecasting, and retrospective assessment within standardised public transport frameworks. Historic data does not include real-time or predictive data, and its validity depends on consistent data collection and curation practices.

6.1.3 Key Performance Indicator

A Key Performance Indicator (KPI) is a quantifiable measure of performance over time and/or space for a specific objective, and is typically tracked over a defined period, allowing for analysis of trends and performance over time. There is normally a target or threshold to compare with, depending on the quality system in use to enable the evaluation of monitored behaviours and identify if they are acceptable or not. KPIs are created from data that has been collected, analysed, and summarised and used to help decision-making.

There are two categories of measurements for KPIs.

- Quantitative facts presented with a specific objective numeric value measured against a standard. Usually, they are not subject to distortion, personal feelings, prejudices, or interpretations.
- Qualitative represents non-numeric conformance to a standard, or interpretation of personal feelings, tastes, opinions or experiences. An example of a qualitative indicator would be a passenger's satisfaction about an aspect of a journey.

6.1.4 Indicator

Key Performance Indicators define a set of values to measure against. These raw sets of values, which can be fed to systems that aggregate the data, are called indicators.

An 'indicator' can only measure what 'has' happened, so the only type of measurement is descriptive or lagging. Any KPI that attempts to measure something in a future state as predictive, diagnostic or prescriptive is no longer an 'indicator', it is a 'prognosticator' – at this point, it is analytics (possibly based on a KPI) but leading KPIs are also used to indicate the amount of front-end loading activities.

7 Types of Reference Data Used in OpRa

7.1 General

The exchange of data between two systems requires that both systems can refer to data instances unambiguously. In OpRa, this involves the use of a number of concrete DATA SOURCE systems to specify the scope and nature of the identifiers of the different types of entity that appear as elements in interfaces, for example LINES, Vehicle SCHEDULED STOP POINTS, DIRECTIONS and Vehicle Features. Table 1 summarises these entities.

Two features need to be specified for each data reference system: a data model and an identifier scope (i.e. a namespace for the allocation of identifiers within the model).

Some data elements, such as time, can be assigned a universal scope, other namespaces such as product references, need to be mutually agreed between communicating pairs of systems, and will be specific to their implementations.

Those data systems that need to be agreed upon for OpRa fall into two groups:

- Simple models, for which there is an exactly equivalent entity in both systems, for which only the namespace and identifier element need to be agreed.
- Complex models, where not only the namespace, but also the submodel needs to be agreed upon – for example, the stop and place location model.

In the latter case it is desirable to try and reduce the model to a simple reference if possible

Table 1 — Data References for OpRa

Reference Data	Type	SIRI XML Element	Scope	Basis	Discovery
Date & Time	Simple	<i>xsd:dateTime</i>	Universal	ISO 8601	--
Language	Simple	<i>xml:language</i>	Universal	ISO 639	--
Location	Complex	<i>LocationStructure</i>	Universal	WGS84 / gml	--
Information Provider	Simple	<i>ParticipantRef</i>	Participants, Regional?	agreement	--

The Participant Reference is used as a general purpose Namespace for data references. The Participant Reference of each system communicating messages must be unique and so provides a guaranteed unique scope, both for message and subscription identifiers, and for data element types originating from a given system.

7.2 Identification of Objects beyond system borders

7.2.1 General

To ensure that different systems can reference the same object, an object ID is required that is recognised by all systems. Within the scope of the OpRa interfaces, stops, routes and transport authorities are examples of such object types that require information exchange over interface services. For this reason, there is a need for referencing systems that are generally known and can be commonly used.

That does not necessarily mean that a software system has to use these object keys in their own operation. It suffices, when the system understands the general object references and is able to depict the relevant information using internal identifiers.

7.2.2 Object identifiers and namespaces

Object identifiers in NeTeX and related standards are made up of a prefix and an identifier string, separate by a colon:

- The prefix is a namespace (or 'Codespace' in the NeTeX sense), a string indicating the coding system being used (for long identifiers it may also indicate a common root value, as say declared by a Codespace element). The prefix should not contain colons. Where there is a nationwide system the country code may conveniently be used as a namespace (e.g. de, fr, ch, etc.).
- The identifier string is a key value within the chosen coding system that uniquely identifies an instance of the object. It may be atomic - based on a single attribute - or composite - based on multiple attributes, some of which may be keys of other object types. Colons may be used to separate the attribute values of a composite key, for example '9162:1:2'. Where a composite key is used, a namespace should also always be used so as avoid confusion between the namespace delimiter and the first colon of the composite key, ie: that 'de:9162:1:2' should be read as 'de' + '9162:1:2' and not 9162 + 1:2. An IFOPT-oriented syntax is used for the schemas to reference objects presented in this chapter. It uses a colon to separate namespaces. For this reason, a colon is a syntactic separator and cannot be used in identifiers.

In an implementation of the OpRa Schema it is possible to declare one or more namespaces and namespace identifiers that are in use in a given data exchange (e.g. by adding a tag to do this to the request/response). The namespace identifiers can then be used both to formally indicate the coding system being used, and to reduce verbosity.

All relevant objects need to have id that are:

- **Persistent:** They do not change over time.
- **Globally unique:** In some cases, a SystemId is added in OpRa to make it clear, which system is using a given id.
- **Managed by one master (system):** This means one organisation is responsible for the id of a given object.
- **Distributed if necessary:** Other systems should use the same id.

In what follows the referencing systems required for different object types will be presented.

It should never be necessary to interpret the content of an id. If information is necessary for users, it needs to be stored in elements or attributes separately.

7.3 Participant (information provider) identification

7.3.1 General

In order to be able to distinguish between the messages from different communicating partners using a service, a unique scope of reference is needed. This scope is indicated in each message by a unique Participant Reference for each participant, using a Participant Code mutually agreed between the participating systems, and which forms part of the unique message reference. The Participant Reference also provides a default scope for all other data references originated from the participant.

7.3.2 Model

Single scope, the ParticipantRef. must be unique to each system in the network.

7.3.3 Namespace and Scope

The Participant Reference is unique between any pair of communicating OpRa participants.

7.4 Participant pair identification (service participant pair code)

7.4.1 General

If necessary, the agreed set of unique identifiers for some or all data references may be specific to the interaction of a pair of participants. That is to say a separate Participant Pair Code may be created (as opposed to the actual Participant Codes of either Participant) to serve as a Namespace that is unique to the interaction of the two parties for a specific service. Both Participants may then map the data values to their own internal identifiers.

7.4.2 Model

The Participant Pair have a single scope, the ParticipantPairRef. This will either be the ParticipantRef of the Producer system, or a more specific mutually agreed scope.

7.4.3 Namespace and Scope

Participant Pair code is unique between any pair of communicating OpRa participants.

7.5 Data formats

7.5.1 Date & time

All timestamps are stated in UTC (Coordinated Universal Time). The use of UTC avoids problems with changeover between summer and winter time zones. Differences from the UTC time zone are coded in accordance with ISO 8601 (e.g.: 2000-04-07T18:39:00+01:00).

In accordance with ISO 8601, if no time difference is given, the time is in UTC; this may be further indicated by the presence of a Z suffix (2002-04-30T12:00:00 corresponds to 2002-04-30T12:00:00Z). In other words, the first 19 characters are obligatory and correspond to local time or UTC.

Time units less than one second are ignored.

```
<ExpectedDepartureTime>2001-12-17T09:30:47+01:00</ExpectedDepartureTime>
```

7.5.2 Time Zones

It should be noted that *xs:dateTime* does not require timezone to be declared, but this always shall be declared in systems which involve more than one timezone.

7.5.3 Georeferences

Several of the OpRa functional services include geospatial point coordinates among their response data, for example for the positions of moving PT vehicles. The actual coordinate system to be used is parameterised: OpRa supports the Geographic Mark-up Language (GML) coordinate formats and data reference systems. The following example shows the encoding of a point for a vehicle LOCATION, using epsg: 4326 as indicated by the srsName attribute.

```
<VehicleLocation srsName="epsg:4326">
```

prTR OPRA:2025(E)

```
<Coordinates>52.5600 3.000012<Coordinates>  
</VehicleLocation>
```

Note that a default value for the srsName attribute may be set on the delivery header, or request context, to apply to all points in the response, and so does not need to be repeated on individual points (though it can be if wished, as shown in the example above). Points may also be assigned an identifier.

```
<ServiceDelivery srsName="epsg:4326">  
...  
<VehicleLocation id=" point96"  
<Coordinates>52.5600 -3.000012<Coordinates>  
</VehicleLocation>
```

The OpRa encoding uses tags and attributes in the OpRa and or appropriate NeTeX or SIRI namespace that correspond exactly to equivalents in the GML namespace. The following example shows an equivalent GML encoding.

```
<gml:location>  
  <gml:Point gml:id="point96" srsName="epsg:4326">  
    <gml:coordinates>52.5600 -3.000012</gml:coordinates>  
  </gml:Point>  
</gml:location>
```

As an optimisation, an alternative encoding is also supported in OpRa: this is an efficient, fixed mark-up comprising a pair of explicit Latitude and Longitude elements, stated in WGS 84 (World Geodetic System 1984) decimal degrees of arc. This encoding is used as a default in OpRa, and allows the validation of the coordinates to be enforced by the schema validator (for a parameterised gml coordinate system, validation must otherwise be enforced by the application according to the srsName parameter).

```
<VehicleLocation>  
  <Longitude>-3.000012</Longitude>  
  <Latitude>52.5600 <Latitude >  
</VehicleLocation>
```

The above optimised OpRa encoding is equivalent to a general gml:coordinates format encoding:

```
<gml:location>  
  <gml:Point gml:id="point96" srsName="epsg:4326">  
    <gml:coordinates>52.5600 -3.000012</gml:coordinates>  
  </gml:Point>  
</gml:location>
```

7.5.4 National language of text elements

In accordance with standard W3C usage, text elements may include an xml:lang attribute to indicate the language in which a string is specified. ISO standard language values should be used to indicate the specific languages, see ISO 639-1, ISO 639-3 and RFC 1766.

In SIRI a Natural Language Data type, which is a string incorporating the XML lang attribute, is used for text elements that may be in different languages. For example:

```
<Name lang="de">München, Hauptbahnhof</Name>  
<Name lang="fr">Munich Gare Centrale</Name>  
<Name lang="en">Munich Main Station</Name>  
<Name lang="it">Munich Main Station</Name>
```

7.5.4.1 Two character ISO Language code.

The global namespace used is as per ISO 639-1.

8 Use cases

8.1 General

A deep and detailed study of the public transport sector has been conducted, focusing on current or desired usage of raw data and statistics on information exchange processes among stakeholders (mainly authorities, operators and system providers) in several EU countries. Particular attention has been posted on the identification of all possible measured data and measured data aggregation and statistics exchanges.

All the aspects have been described in terms of Public Transport National scenarios, including high-level description of the role of indicators in PT Governance. Thanks to this contribution, national requirements have been collected and analysed, focusing on data input for PT network study including by organisations including ERA, UITP and UIC.

8.2 Purpose

This section to outline the use cases for OpRa according to the defined scope carefully identified on the base of national Public Transport Service scenarios and Public Transport QoS in EU projects with the aim to have as precises as possible a standard.

8.3 Actors and use case categories

8.3.1 Actors

The main actors involved in OpRa are the following:

- **Public Transport Operator** (PTO) in charge to provide PT Service in accordance with a contract with Public Authority. Interest in having an efficient Service to reduce costs
- **Public Transport Authority** (PTA) at several levels (Central and delegated ones) in charge to create contracts and verify results of the PT Service respect contract Service Level Agreement (SLA).
- **System Integrators** that provide hardware devices and software platforms and applications to facilitate the operations of PT Service.
- **PT Service passengers** that represent the demand of transport to be satisfied in a sustainable way by Public Administrations and stakeholders.

8.3.2 Use case categories

The OpRa main use cases cover data exchange for operation analysis and contractual reporting purposes. They have been divided into the following use cases groups:

- **Situation analysis.** This use case group covers all the data exchange scenarios where the aim of the exchange is to provide data to examine and study the operating status (e.g. delays because of traffic lights, road construction, traffic jams, etc.). It requires some recording of aspects of the operational context and of events along with the real-time data performance.
- **Contractual reporting.** This use case group covers data exchange where the actual service is accounted towards the local, regional or national administration body responsible for providing public transportation. It requires some representation of the accounting of undertaken services.

- **Providing data for Quality-of-Service analyses and processes.** Based on the planned timetables and the exchanged data, quality-of-service (QoS) analyses may take place, including delays and cancelled vehicle journeys,

During the information gathering phase, a more detailed categorisation of use cases has been defined according to the specific aspect of the Public Transport Service it dealt with.

The following types have been defined:

- **High Level Generic:** This is a particular category, which gathers generic use cases for OpRa potentially relevant to data format and update criteria, and in general all the UCs relevant to XSD format design and future implementation of exchanging data.
- **Service Offer:** It gathers all the themes and use cases relevant to the Offer of Transport for PT Service (e.g. Spatial and time coverage, offered seats, etc.). This category is further divided into following sub-categories:
 - Planned Service Offer (as result of Strategic and Tactical planning phases);
 - Actual (measured) Service Offer.
- **Service demand:** It gathers all the themes and use cases relevant to the Demand of Transport for PT Service (O/D matrix, load factor, etc.). This category is further divided into following sub-categories:
 - Expected Service Demand;
 - Actual (measured usage) Service Demand.
- **Service externality:** It gathers all the themes and use cases relevant to PT Service Externality, that imply cost or benefit that affects an external party who did not choose to incur that cost or benefit (pollution emissions, safety, etc.).
- **Service Economy:** It gathers all the themes and use cases relevant to the Economy of Transport for PT Service in terms of incoming funds (sold tickets, etc.) and expenses (costs, etc.). This category is further divided into following sub-categories:
 - Income Service Economy;
 - Expenses Service Economy.
- **Service efficiency:** It gathers all the themes and use cases relevant to the Efficiency of Transport for PT Service (lines overlaps, MTBF, delays, etc.).

A correlation between the two categorisation and the involved actors have been defined as reported in the following Traceability Matrix.

Table 2 — OpRa use case categories and actors, traceability matrix

Macro-category	Sub-category (if any)	Actors				Analysis scenarios		
		PTO	PTA	System Integrator	Passenger	Situation analysis	Contractual reporting	Data for QoS

Macro-category	Sub-category (if any)	Actors				Analysis scenarios		
		PTO	PTA	System Integrator	Passenger	Situation analysis	Contractual reporting	Data for QoS
OFFER	PLANNED	X	X	X			X	x
	ACTUAL (measured)	X	X	X	X	X	X	x
DEMAND	EXPECTED	X	X	X			X	
	ACTUAL (measured usage)	X	X	X	X	X	X	
EXTERNALITY		X	x	X	X	x	X	
ECONOMY		X	X				X	
EFFICIENCY		X	X	X	x	X	X	X

8.3.3 Collection of use cases

8.3.3.1 General

The list of use cases in this section presents a variety of public transport activities that could benefit by using the raw data exchange. Therefore, in most cases the focus lies on the data contents that are to be exchanged (content use cases). Eighteen use cases have been identified, as depicted in the following table:

Table 3 — OpRa use cases

No.	OpRa use case
1	Public Transport Service dimensions
2	Public Transport Service Commercial speed
3	Public Transport Service spatial coverage
4	Public Transport Service temporal coverage
5	Public Transport Service Inter-change nodes
6	Origin/Destination zones connections
7	Origin/Destination Matrix
8	O/D zones moving cause
9	Public Transport Fleet dimensions
10	Transport Demand Dimensions
11	Public Transport Service Load factor
12	Public Transport Service safety
13	Public Transport Service pollution emission

No.	OpRa use case
14	Public Transport Service incoming funds
15	Public Transport Service costs
16	Public Transport Operator dimensions
17	Service delays and inefficiencies
18	Public Transport Service performed distance

For the purpose of the work and to be able to identify the raw data needed to exchange and store, each use case has been specialised in a series of elementary use cases. The following Traceability Matrix highlights the relations among use cases and elementary use cases.

Table 4 — Use cases, traceability Matrix

			Macro-category	OFFER		DEMAND		EXTERNALITY	ECONOMY	EFFICIENCY
No.	Use cases	Elementary use cases	Sub-category (if any)	Planned	Actual (measured)	Expected	Actual (measured usage)			
1	Public Transport Service dimensions	13		X	X					
2	Public Transport Service Commercial speed	2		X	X					
3	Public Transport Service spatial coverage	1		X	X					
4	Public Transport Service temporal coverage	1		X	X					
5	Public Transport Service Inter-change nodes	2		X	X					
6	Origin/Destination zones connections	2		X	X		X			
7	Origin/Destination Matrix	6				X	X			
8	O/D zones moving cause	2				X	X			
9	Public Transport Fleet dimensions	4				X	X			
10	Transport Demand Dimensions	10								
11	Public Transport Service Load factor	2		X	X					
12	Public Transport Service safety	6						X		
13	Public Transport Service pollution emission	1						X		
14	Public Transport Service incoming funds	16							X	
15	Public Transport Service costs	5							X	

			Macro-category	OFFER		DEMAND		EXTERNALITY	ECONOMY	EFFICIENCY
No.	Use cases	Elementary use cases	Sub-category (if any)	Planned	Actual (measured)	Expected	Actual (measured usage)			
16	Public Transport Operator dimensions	4								X
17	Service delays and inefficiencies	4								X
18	Public Transport Service performed distance	6			X					X

In the following sections, elementary use cases are described for each category. The following histogram reports the number of use cases (UCs) per category:

Table 5 — Use cases for each category

Category	# UCs
0 — Generic High Level	9
1 — Service Offer	18
2 — Service Demand	17
3 — Service Externality	7
4 — Service Economy	21
5 — Service Efficiency	16
Total	88

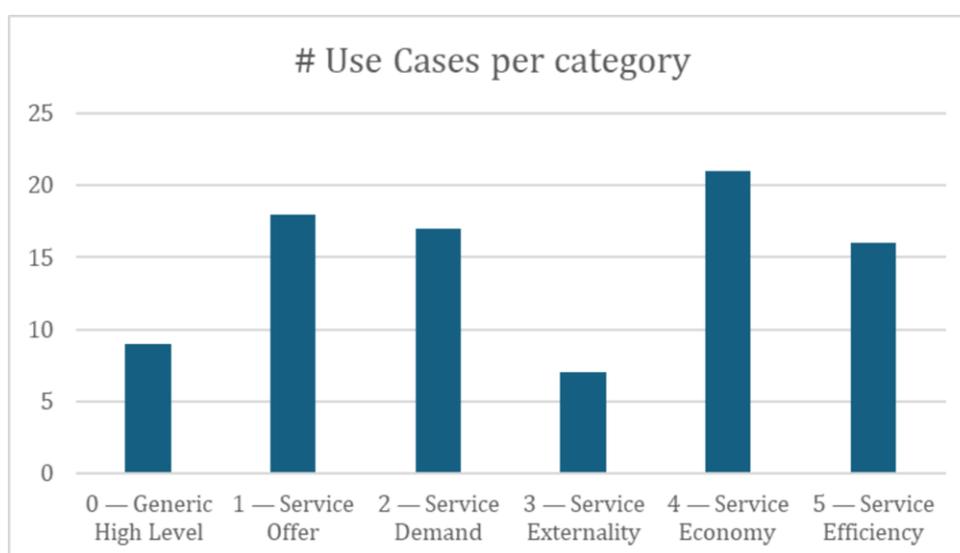


Figure 3 — Visualisation of the number of use case per category

8.4 Prioritisation of Use Cases for Modelling

The design of XSD schema for the exchange of raw data needed to enable KPI calculations required a prioritisation of Use Cases, primarily to ensure that the requirements of the MMTIS regulation are being met, and secondly to facilitate the governance activities for the PT Services.

The below tables indicate for each OpRa use case the priority determined according to specific criteria:

- P1: if the use case has HIGH relevance for MMTIS regulation
- P2: if the use case is relevant for contracts management and so for the PTA governance of the Service
- P3: others

8.4.1 Service Offer

Table 6 — Prioritisation of use cases planned service offer

Use Case	Description	Main actors	Member States	MMTIS relevance	Notes	Priority
Use Case: OFFER-001-001	Total length of planned distance to be run by PT Vehicle.	PTO, PTA	4	LOW	High importance for PTA in charge to manage contracts.	P3
Use Case: OFFER-001-002	Planned number of offered Fleet Vehicle seats	PTO, PTA	3	HIGH	Improve multimodal travel information services and facilitate passengers' travel.	P1
Use Case: OFFER-001-003	Mean Fleet Vehicle seats per offered planned distance	PTO, PTA	1	HIGH	Observed data on delays and cancellations, linked to rules on passengers' rights and obligations, can also enable service providers to inform passengers about their rights for compensation.	P1
Use Case: OFFER-001-004	Planned VEHICLE JOURNEYS	PTO, PTA	2	MEDIUM		P2
Use Case: OFFER-001-005	Planned commercial speed	PTO, PTA	1	MEDIUM		P2
Use Case: OFFER-001-006	Total number of STOP POINTs in the Network	PTO, PTA	1	LOW		P3
Use Case: OFFER-001-007	Total distance of ROUTEs	PTO, PTA	1	LOW		P3
Use Case:	Provide rescheduling	PTO, PTA	2	MEDIUM	Covered by NeTEx - change due to control	P2

OFFER-001-008	information				action or rescheduling the latter is relevant to this UC.	
Use Case: OFFER-001-009	The planned service (i.e. network description and timetable description) shall be provided Shall be provided under a well-documented standard and profile (as NeTEx).	PTO, PTA	2	HIGH	Not OpRa KPI.	P1
Use Case: OFFER-001-010	Report the planned vehicle type per VEHICLE JOURNEY	PTO, PTA	2	MEDIUM	Can be provided via NeTEx, can be Country based, can be of interest for Accessibility.	P2
Use Case: OFFER-001-011	Report the number of vehicles in the fleet (global and per vehicle type)	PTO, PTA	1	LOW	Can be provided via NeTEx.	P3

Table 7 — Prioritisation of use cases actual service offer

Use Case	Description	Main actors	Member States	MMTIS relevance	Notes	Priority
Use Case: OFFER-002-001	Total Fleet Vehicle performed distance	PTO, PTA	4	MEDIUM	Only data on the arrival time or the departure time, or both, and, where possible the reasons for delays or cancellations should be made accessible via the national access point as specified in Article 3, not any operational data collected during the journey. * Interest of Agencies	P2
Use Case: OFFER-	Measured number of offered Fleet Vehicle seats	PTO, PTA	3	HIGH		P1

002-002						
Use Case: OFFER-002-003	Mean Fleet Vehicle seats per offered performed distance	PTO, PTA	1	HIGH		P1
Use Case: OFFER-002-004	Actual VEHICLE JOURNEYS	PTO, PTA	2	LOW		P3
Use Case: OFFER-002-005	Actual commercial speed	PTO, PTA	1	MEDIUM	* Interest of Agencies	P2
Use Case: OFFER-002-006	Report the actual vehicle type per VEHICLE JOURNEY (should be different from planed information as soon as the number of seats, and number of seats for disabled persons or accessibility services are different from the planed version),	PTO, PTA	2	MEDIUM		P2
Use Case: OFFER-002-007	Report the number vehicle used during operation (global and per vehicle type)	PTO, PTA	1	LOW		P3

8.4.2 Service Demand

Table 8 — Prioritisation of use cases planned service demand

Use Case	Description	Main actors	Member States	MMTIS relevance	Notes	Priority
Use Case: DEMAND-001-001	Expected number of passengers	PTO, PTA	4	MEDIUM	It considers the total expected number of passengers by the Public Transport Service in the reference period (typically the year). Goal: Evaluate the expected demand respect the reference Service contract. Formulae: Expected number of transported	P1

					passengers Possibly on a per line and per day base	
Use Case: DEMAND-001-002	Expected number of passengers on the total length of Vehicle path	PTO, PTA	1	LOW	It considers the total expected number of passengers and the total planned distance by the PTO to interrogate the Public Transport Service in the reference period (typically the year). Goal: Evaluate the expected demand respect the planned distance of each PTO respect the reference Service contract. Formulae: Expected number of transported passengers / Total planned distance	P3
Use Case: DEMAND-001-003	Expected number of passengers multiplied by total length of Vehicle path.	PTO, PTA	1	LOW	It considers the total expected number of transported passengers and the total planned distance to be performed by the PTO to interrogate the Public Transport Service in the reference period (typically the year). Goal: Evaluate the expected demand respect the planned distance of each PTO respect the reference Service contract. Formulae: (Expected number of transported passengers * total planned distance)	P3
Use Case: DEMAND-001-004	Expected number of passengers multiplied total planned distance on the total number of vehicles	PTO, PTA	1	LOW	It considers the total expected number of transported passengers, the total planned distance and the total number of to be used vehicles by the PTO to interrogate the Public Transport Service in the reference period (typically the year). Goal: Evaluate the expected demand respect the planned distance and vehicle fleet of each PTO respect the reference Service contract. Formulae: (Expected number of transported passengers * total planned distance) / Total number of vehicles	P3
Use Case: DEMAND-001-005	Expected reason for traveling	PTO, PTA, studies, city planning	1	LOW	Provision of expected reasons for traveling, segmented by day, period of the day and line.	P3

Use Case: DEMAND-001-006	Expected origin and destination of travels	PTO, PTA, studies, city planning	1	LOW	Provision of expected distribution of origin and destination of travels, segmented by day and period of the day.	P3
Use Case: DEMAND-001-007	Expected mode of travel	PTO, PTA, studies, city planning	1	LOW	Provision of expected modes of travel on a per trip basis (may involve several modes), segmented by day and period of the day.	P3

Table 9 — Prioritisation of use cases actual service demand

Use Case	Description	Main actors	Member States	MMTIS relevance	Notes	Priority
Use Case: DEMAND-002-001	Measured number of passengers	PTO, PTA	2	HIGH		P1
Use Case: DEMAND-002-002	Measured number of passengers on the total performed distance		2	MEDIUM		P2
Use Case: DEMAND-002-003	Measured number of passengers multiplied by total performed distance		2	MEDIUM		P2
Use Case: DEMAND-002-004	Measured number of passengers multiplied total performed distance on the total number of vehicles	PTO, PTA	2	MEDIUM		P2
Use Case: DEMAND-002-005	Actual reason for traveling	PTO, PTA, studies, city planning	1	LOW	Provision of actual reasons for traveling, segmented by day, period of the day and line.	P3
Use Case: DEMAND-002-006	Actual origin and destination of travels	PTO, PTA, studies, city planning	1	LOW		P3
Use Case: DEMAND-002-007	Actual mode of travel	PTO, PTA, studies, city planning		LOW	Provision of actual modes of travel on a per journey basis (may involve several modes), segmented by day and period of the day.	P3
Use Case: DEMAND-002-008	Actual connections on during travels	PTO, PTA, studies, city planning	1	LOW	Provision of actual connexions of passenger during their travel: - number of connections per trip - number of times a	P3

					connection link is used On a day-type/period basis	
Use Case: DEMAND- 002-009	Actual use of journey planners	PTO, PTA, studies, city planning	1	LOW	Provision of actual use of journey planner: - number of request (time period based, unique user based) - origine/destination distribution - number of "no route found" (and report associated OD) Goal: Evaluate the difference between requests and usage Evaluate unserved needs ("no route found")	P3
Use Case: DEMAND- 002-010	Actual total travel time	PTO, PTA, studies, city planning	1	LOW	Provision of the total travel time, in PT and outside PT, for a period	P3

8.4.3 Service Efficiency

Table 10 — Prioritisation of use cases planned service offer

Use Case	Description	Main actors	Member States	MMTIS relevance	Notes	Priority
Use Case: EFFICIENCY- 001	It considers the number of vehicles on a per 5 years age range (e.g. older than 5 years,10 years, etc.) used by the PTO to provide the Public Transport Service in the reference period (typically the year). Goal: Evaluate the technical fleet efficiency of each PTO respect the reference Service contract. Formulae: N.A.	PTO, PTA	3	LOW		P3
Use Case: EFFICIENCY- 002	It considers the number of vehicles on a per 5 years age range (e.g. older than 5 years,10 years, etc) respect the total number of fleet vehicles, used by the PTO to provide the Public Transport Service in the reference period (typically the	PTO, PTA	2	LOW		P3

	<p>year).</p> <p>Goal: Evaluate the technical fleet efficiency of each PTO respect the reference Service contract.</p> <p>Formulae: Number of vehicles in a specific age range / total number of vehicles</p>					
Use Case: EFFICIENCY-003	<p>It considers the total performed distance normalized by total number of employees of each PTO to provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the efficiency of each PTO considering the number of total employees and in the respect the reference Service contract.</p> <p>Formulae: Performed distance / number of employees</p>	PTO, PTA	1	LOW	* Interest of Agencies	P3
Use Case: EFFICIENCY-004	<p>It considers the total performed distance normalized by total number of drivers of each PTO to provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the efficiency of each PTO considering the number of total drivers and in the respect the reference Service contract.</p> <p>Formulae: Performed distance / number of drivers</p>	PTO, PTA	1	LOW	* Interest of Agencies	P3
Use Case: EFFICIENCY-005	<p>It considers the mean performed distance of each vehicle of each PTO to provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the efficiency in the usage of vehicles of each PTO respect the reference Service contract.</p> <p>Formulae: Performed distance / number of vehicles on a per contract basis.</p>		1	LOW	* Interest of Agencies	P3
Use Case: EFFICIENCY-	<p>It considers the category of the employees for each vehicle of each PTO to provide the Public</p>		2	LOW	* Interest of Agencies	P3

006	<p>Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the employees distribution in the different categories (drivers, administrative, engineering, etc.) of each PTO respect the reference Service contract.</p> <p>Formulae: Percentage and absolute distribution of employees for all contractual categories</p>					
Use Case: EFFICIENCY-007	<p>It considers the total number of employees and drivers for each vehicle of each PTO to provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the percentage of drivers on total number of employees of each PTO respect the reference Service contract.</p> <p>Formulae: Total number of drivers / Total number of employees</p>	PTA	1	LOW		P3
Use Case: EFFICIENCY-008	<p>It considers the total performed distance by each vehicle to provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the distance performed by each vehicle for maintenance purposes.</p> <p>Formulae: Total number of km for each fleet vehicle</p>	PTO	1	MEDIUM	* Interest of Agencies	P2
Use Case: EFFICIENCY-009	<p>It considers the occurred and resolve date of each Failure for each vehicle used to provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the maintenance efficiency of each vehicle to monitor the maintenance purposes.</p> <p>Formulae: MTBF</p>	PTO	3	LOW		P3
Use Case:	It considers the occurred delay at	PTO		HIGH	As regards	P1

EFFICIENCY-010	<p>a specific point (STOP_POINT) for each line of the Public Transport Service in the reference period (day, month). Goal: Evaluate the efficiency of the Service in terms of delays. Formulae: Mean (Arrival Time - Scheduled Time)</p>				<p>the historic and observed travel and traffic data, only data which are relevant to enhance multimodal travel information services should be shared.</p>	
Use Case: EFFICIENCY-011	<p>Should be available :</p> <ul style="list-style-type: none"> - on a per stop base (especially first and last stops) - on an origine/destination base - on a peak/of-peak base - on an area (i.e. region or city) base <p>The definition of late shall be very clearly stated:</p> <ul style="list-style-type: none"> - maximum delay to be classified as on time (may depend on mode) - there may be a specific definition for headway services - cancelled services shall be clearly identified - additional services shall be clearly identified 	PTO, PTA	2	HIGH	Number of late services	P1
Use Case: EFFICIENCY-012	<p>Delay and late service shall be categorized depending on their cause:</p> <ul style="list-style-type: none"> - incidents due to the infrastructure, - due to external events (weather, passenger, traffic congestion, etc), - due to the operator's management and decision, - to strikes or social event, 	Service delays and inefficiencies	2	HIGH	As regards the historic and observed travel and traffic data, only data which are relevant to enhance multimodal travel information services should be shared.	P1
Use Case: EFFICIENCY-013	<p>Number of full or nearly full, and empty or nearly empty vehicles. On a per line (or</p>	PTO, PTA	1	HIGH	It concerns occupancy.	P1

	origin/destination), per peak/off-peak, and vehicle type base.					
Use Case: EFFICIENCY-014	Report the number of cancelled journeys, and associated reasons (number of cancelations per reason type), on a predefined period base.	PTO, PTA	1	HIGH		P1
Use Case: EFFICIENCY-015	It considers the age category of the employees for each PTO to provide the Public Transport Service in the reference period (typically the year). Goal: Evaluate the employees age distribution in the different categories (drivers, administrative, engineering, etc.) of each PTO respect the reference Service contract. Formulae: Percentage and absolute distribution of employees age for all contractual categories.	PTO, PTA	1	LOW		P3
Use Case: EFFICIENCY-016	It considers the maximal number of drivers needed for daily operation of the Public Transport Service in the reference period (typically the year). Goal: Evaluate the measured demand respect the reference Service contract. Formulae: Measured number of maximal needed daily drivers	PTO, PTA	1	LOW		P3

9 Conceptual Data Model

9.1 Concepts introduced in OpRa

9.1.1 Data preparedness

Many use cases in OpRa use data sources that can have various levels of reliability. Informing the consumer of a dataset about the reliability of the raw data and other indicators being exchanged an important block of metadata. For example, if an onboard device data aggregator acting as a producer states that the data has been fully processed and cleaned, the consumer side is not required to develop and maintain data cleansing functionality.

To be able to calculate the metrics transferred, using OpRa will require, depending on the data about a metric being transferred, access to the raw unprocessed data for both planned and unplanned service operations as well as processed clean data.

For example, in the scenario where data is being exchanged between a PTO unit and a PTO server there are three stages of preparedness of the data:

- **Raw**
- **Cleansed**
- **Processed**

9.1.1.1 Concept overview

To illustrate the differences between the preparedness states, the following table sums the roles of a producer and a consumer system, and lists some consequences which would trigger some architecture planning considerations in an OpRa-using environment:

Table 11 — Data preparedness Concepts

	Preparedness: Raw	Preparedness: Cleansed	Preparedness: Processed
Producer / Consumer roles	Producer states that the data contains the whole measurement, but it may contain erroneous data. It is up to the Consumer to get rid of measurement errors if needed	Producer states that erroneous data was removed from the measurement. Consumer may further investigate errors if required, but that's not mandatory	Producer states that validating the data had been finished and shall not be modified. Consumer can use the data without further examination of quality
Consequences	Consumer needs to develop data cleansing methods Consumer may not be aware of measurement characteristics (error probabilities) of Producer's equipment Consumer is responsible for calculating statistics	Producer can hide measurement equipment's characteristics (error probabilities) Producer has larger responsibilities depending on regulatory environment Quality of the Producer system might affect overall performance, since it filters out data Consumer software can be simpler, focusing more on higher-level tasks Consumer is responsible for calculating statistics	

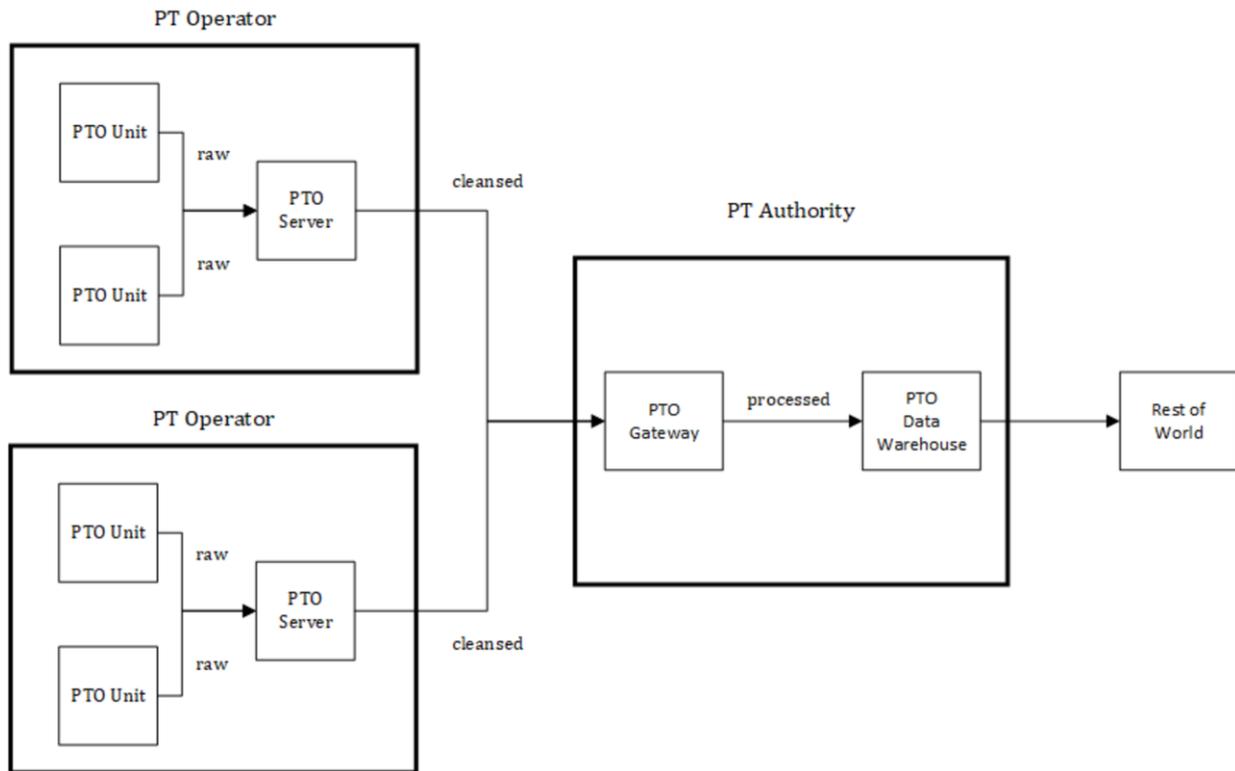


Figure 4 — Data preparedness stages for a PTO – PTA environment

9.1.1.2 Preparedness level raw

In the scenario where data is being exchanged between a PTO unit and PTO server data for the use case EFFICIENCY-014: Number of cancelled journeys, the following data may be needed:

- planned service
- Requires actual service: control actions (/PARTIAL/ JOURNEY CANCELLATION)
- DATED VEHICLE JOURNEY reference
- Cancellation type (PARTIAL / full)
- Control action metadata

This data may contain partial or unchecked data which originates from a given person on duty at a given place.

No checking or processing of data has occurred. This data can be referred to as raw.

9.1.1.3 Preparedness level cleansed

Where data goes through some cross-checking before transferring, for example it may go through some integration process combining from multiple sources (e.g. workstations).

Where there is an obvious error, or the data value is outside of expected and or allowed values this data element may have been removed.

This data is the same as the raw data values except false data has been removed. This data can be referred to as cleansed.

9.1.1.4 Preparedness level processed

Once data has been received in the PTA data warehouse by aggregating data from multiple PTOs it may undergo processing. This processing may have corrected a value where there is an obvious error or the data value is outside of expected and or allowed values.

This data is the same as the raw data except where false data has been corrected. This data can be referred to as processed.

9.1.2 Statistics

9.1.2.1 Distribution analysis support

OCCURRENCE represents the distribution of indicators, comparing the number of a subset to the total number of indicators within a statistical domain.

The **Interval Occurrences Aggregation MODEL** defines a generic structure for statistical aggregation based on intervals and occurrences, serving as a reusable foundation for domain-specific models like late journeys or cancellations. It introduces two core classes: **DURATION INTERVAL**, which represents a time-based category with attributes such as *StartOfInterval*, *EndOfInterval*, and inclusivity flags, and **OCCURRENCE**, which captures aggregated metrics including *NumberOfOccurrences* and *PercentageOfOccurrences*. These classes are linked so that occurrences can be grouped within defined intervals, enabling flexible reporting by time bands or ranges. The model also references **LOGGABLE OBJECT** from the generic logging framework, ensuring that aggregation results can be traced back to recorded events. By abstracting these concepts, this model provides a common statistical layer that other packages, such as those for late journeys, cancelled journeys, or expected passenger counts can specialize, ensuring consistency and interoperability across all performance and KPI calculations in OpRa.

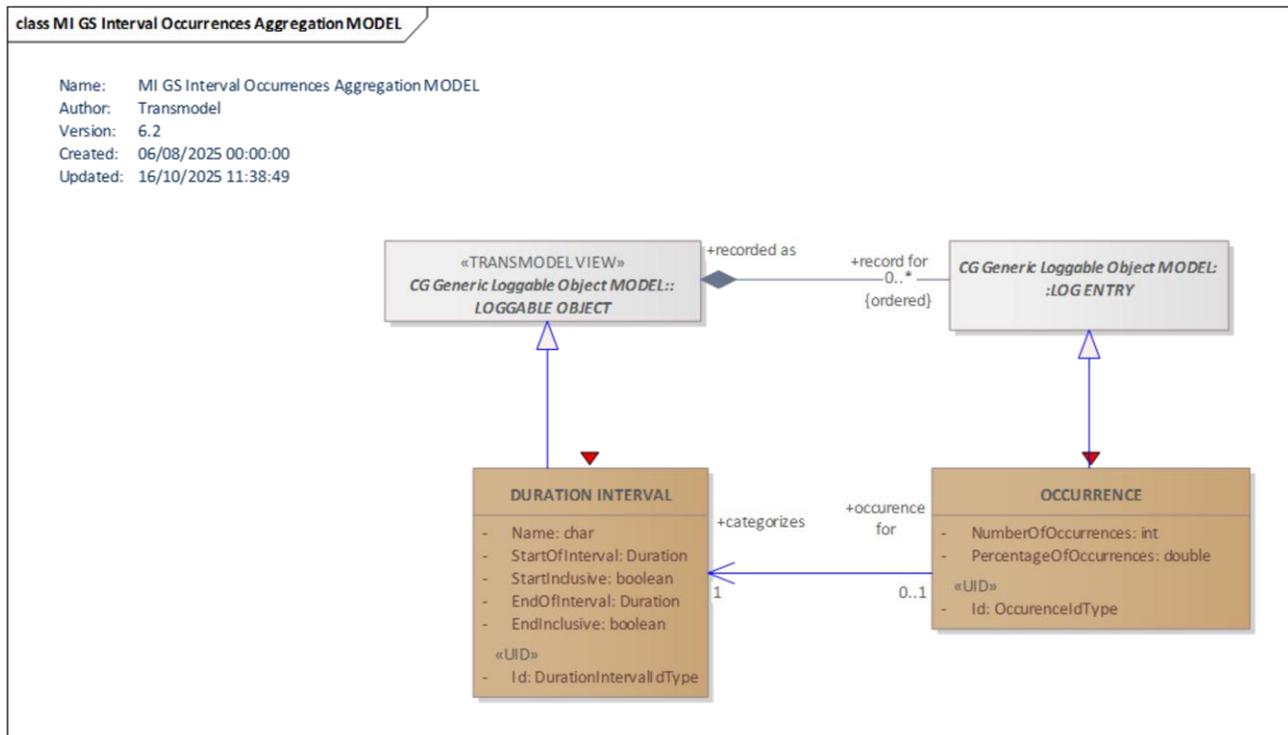


Figure 5 — Interval Occurrences Aggregation MODEL

9.1.2.2 Distribution of duration type indicators

In the case of the subset being a one-dimensional range of duration value defined by a lower and an upper limit (allowing infinity), DURATION INTERVALs define these intervals, also adding the capability of excluding or including the limit in the interval.

For example, when classifying DELAYs in the public transport service, there might be different consequences (refunds toward passengers, bonus-malus systems of operators, etc.) of the DELAYs, with the limits being defined in the regulatory context of public transport operations. Thus, these limits are not pre-defined in OpRa.

Once data has been received in the PTA data warehouse and has been processed it can be used to provide statistical data for various requests. The data can be aggregated for further processing or reported on at a granular level.

For example, once process data can be used in an aggregated manner to create:

- Aggregation metadata
- Aggregated values (calculated)
- Number of total journeys
- Number of cancelled journeys
- Number of partially cancelled journeys
- Number of partially or fully cancelled journeys

To be able to report some statistics it will be necessary to combine multiple aggregated data. For example, to calculate use case DEMAND-001-002 Expected number of passengers on the total length of Vehicle path it is necessary to divide DEMAND-001-001 by OFFER-001-001.

This data can be referred to as statistics.

9.2 UML models

The OpRa framework is split into six main groups of conceptual models, each defined as an UML package.

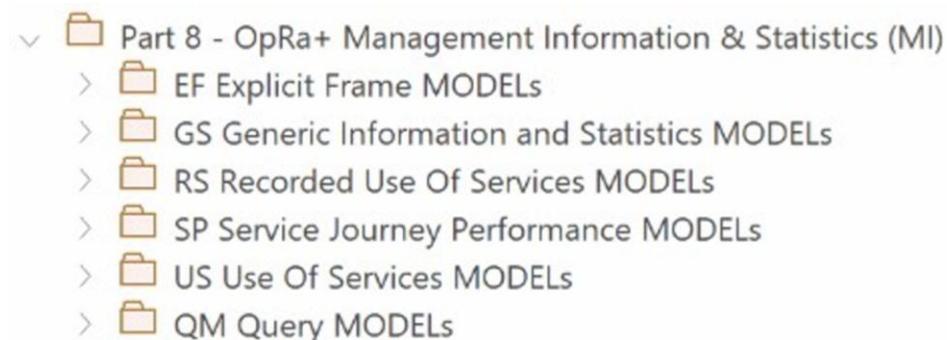


Figure 6 — OpRa Framework packages

The UML package structure organizes the OpRa data model into six major conceptual packages, each grouping related models for clarity and modularity.

- **EF** Explicit Frame MODELS provides the foundational service frames, including RAW DATA FRAME, ACTUAL SERVICE FRAME, PLANNED SERVICE FRAME, and KPI FRAME, which enclose the core data contexts for management information and statistics.
- **GS** Generic Information and Statistics MODELS introduce reusable components for aggregations and calendar-aware scopes, enabling time-based and statistical analysis across the system.
- **RS** Recorded Use Of Services MODELS focus on capturing actual service demand and passenger usage, with detailed sub-models for counting methods and aggregated validation entries.
- **SP** Service Journey Performance MODELS addresses performance indicators, such as the number of cancelled journeys, while SP Number of Late Journeys MODELS specifically model late journeys detection and aggregation, including classes like LATE DATED VEHICLE JOURNEY ENTRY and LATE DATED VEHICLE JOURNEY COUNT.
- **US** Use Of Services MODELS complement these by modelling expected service demand rather than observed usage; it includes US Service Demand MODELS, which further specialize into US Expected Number of Passengers MODELS, containing elements such as EXPECTED NUMBER OF PASSENGERS indicators, EXPECTED PASSENGERS COUNT, and EXTERNAL EXPECTED PASSENGER COUNT to support forecasting and planning.
- Finally, a complementary **QM** Query Models section provides structures for querying and reporting. Together, these packages create a layered architecture that supports monitoring, analysis, forecasting, and control of public transport services in alignment with Transmodel standards.

OpRa by default only provides references to the original data in a LOG ENTRY. This use of references has been used to help manage the size of data being transferred by OpRa. If the consuming system does not have access to the details of the original data, for example a PT SCOPE then this data will need to be transferred separately.

When using OpRa the system participants will need to agree how and when concrete log objects or raw data are used.

Colour is used systematically in OpRa to help make the diagrams easier to understand. Elements from the same functional domain are mostly given the same default colour in all diagrams across all the different levels of abstraction and are based on those used in Transmodel.

Generic framework elements are mostly shown in grey.

Lines in red identify connections new in Transmodel v6.2

Lines in blue identify connections created by the requirements identified by OpRa (to be completed in final version of the TS).

9.3 Modelling Explicit Frames

The diagram of the Explicit Frame Hierarchy models the hierarchical structure of frame-based data containers used in OpRa to organize and exchange public transport information. At its core is the COMPOSITE FRAME (from RC Composite Frame MODEL), which aggregates multiple specialized frames to provide a unified context for data exchange. These include the RAW DATA FRAME (from RF Raw Data Frame MODEL), PLANNED SERVICE FRAME (from PF Planned Service Frame MODEL), ACTUAL SERVICE FRAME (from AF Actual Service Frame MODEL), and KPI FRAME (from KF Kpi Frame MODEL), each carrying domain-specific content such as planned timetables, real-time operational data, and performance indicators. Supporting this structure are generic components like VERSION FRAME (from CC Generic Version Frame MODEL) for version control and VALIDITY CONDITION (from CC Generic

Validity MODEL) to define temporal applicability. This layered design ensures interoperability and consistency across planning, monitoring, and reporting processes in public transport systems.

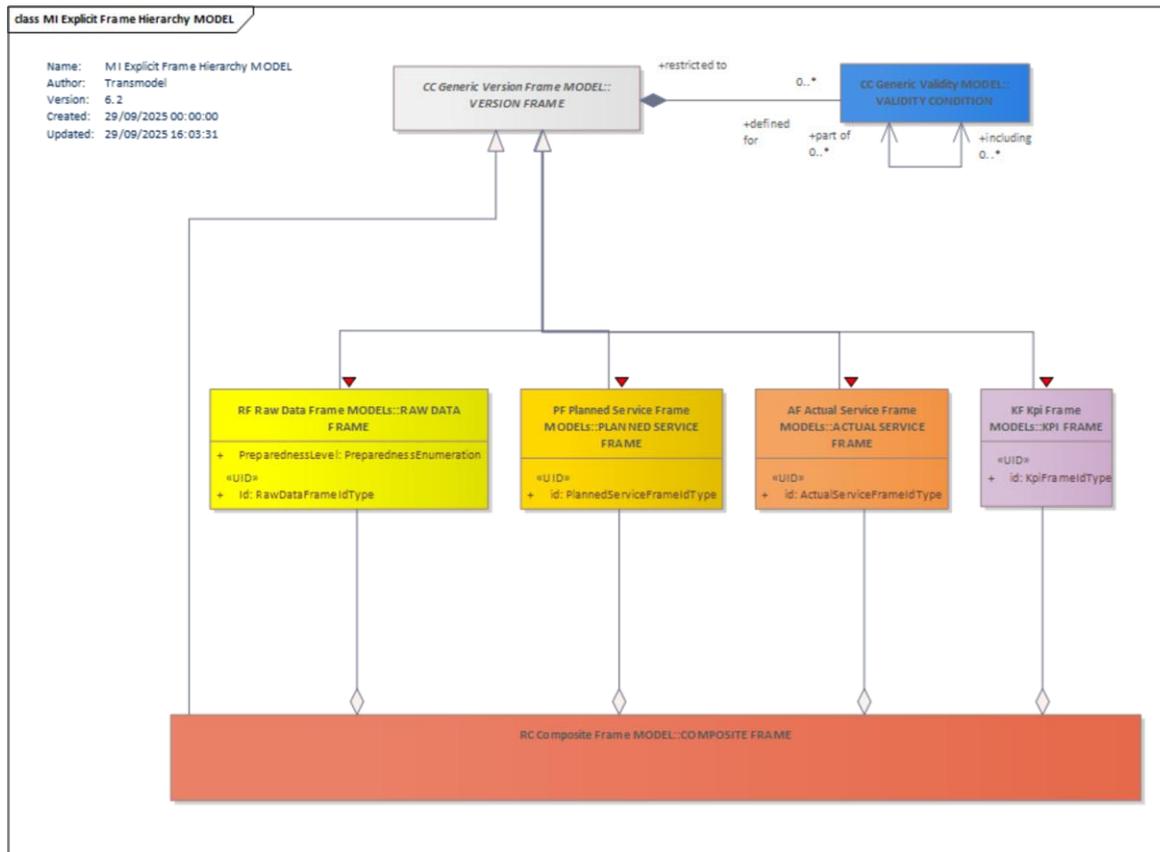


Figure 7 — Explicit Frame Hierarchy MODEL

9.4 Modelling Generic Information and Statistics

The model defines a set of generic, reusable classes that provide temporal and contextual structures for other models, such as those handling late journeys, cancellations, and expected passenger counts. At its core is **CALENDAR AWARE PT SCOPE**, which anchors statistical or operational indicators to a defined time frame and service context. This scope is enriched by classes from the **RC Service Calendar MODEL**, including **SERVICE CALENDAR**, **OPERATING PERIOD**, **OPERATING DAY**, **DAY TYPE**, and **TIME BAND**, which together describe when services are planned or active. These elements allow models to express KPIs or aggregations in relation to specific dates, periods, or recurring patterns (e.g., weekdays, peak hours). The model also references **PT SCOPE** from the publication scope model, ensuring that aggregated or reported data can be filtered by organizational or geographic boundaries. By centralizing these temporal and scope definitions, this model enables consistent interpretation of time-based indicators across all related models, supporting interoperability and accurate performance analysis.

— on an area (i.e. region or city) base,

The definition of late shall be very clearly stated:

- maximum delay to be paid on time (may depend on mode)
- there may be a specific definition for headway services
- additional services shall be clearly identified.

The following figure shows conceptual classes and attributes for the number of late journeys model elements.

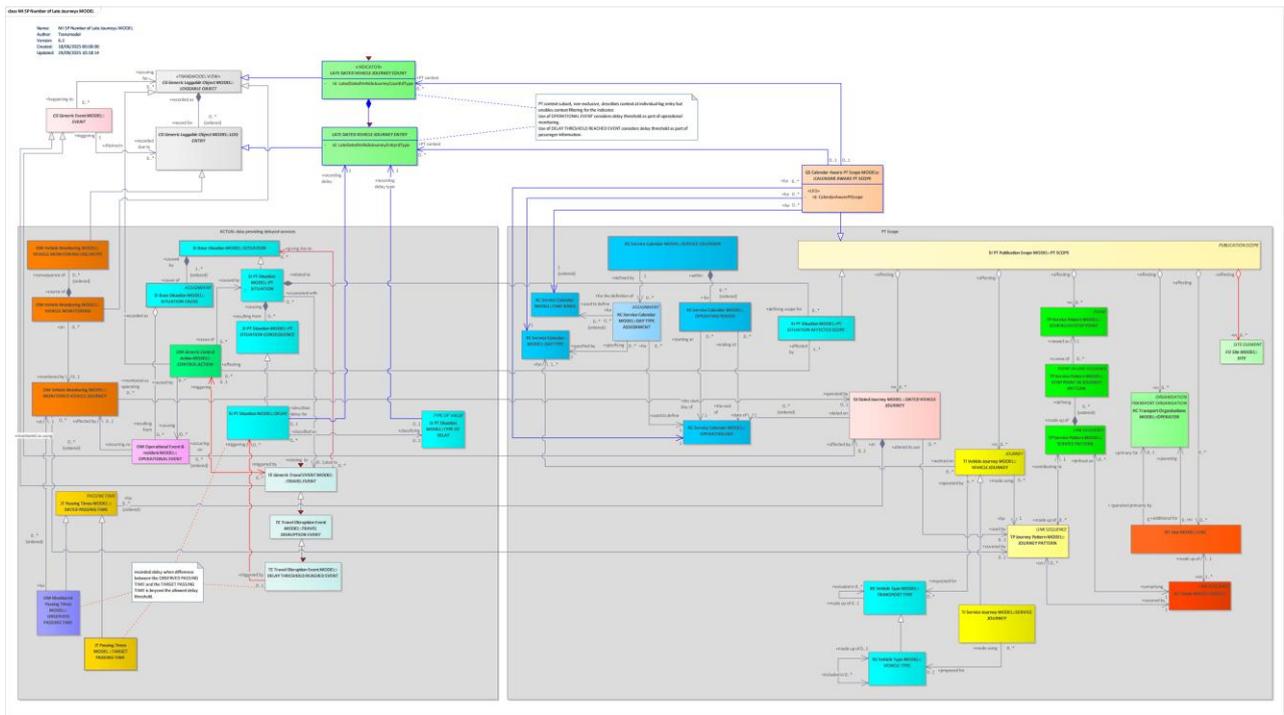


Figure 9 — Number of Late Journeys MODEL Overview

class MI SP Number of Late Journeys MODEL

Name: MI SP Number of Late Journeys MODEL
 Author: Transmodel
 Version: 6.2
 Created: 18/06/2025 00:00:00
 Updated: 29/09/2025 10:18:14

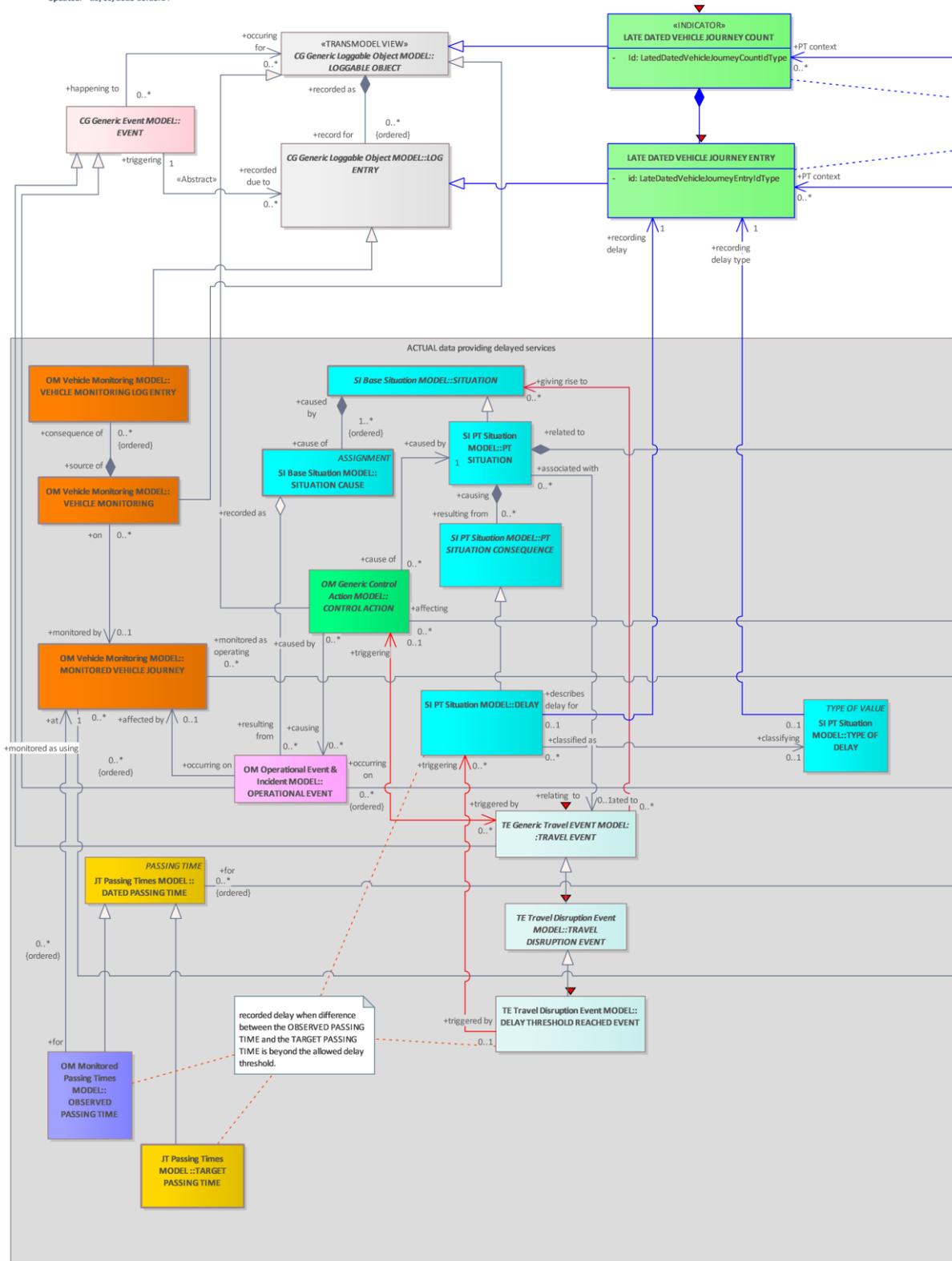


Figure 10 — Number of Late Journeys MODEL ACTUAL data providing delayed services

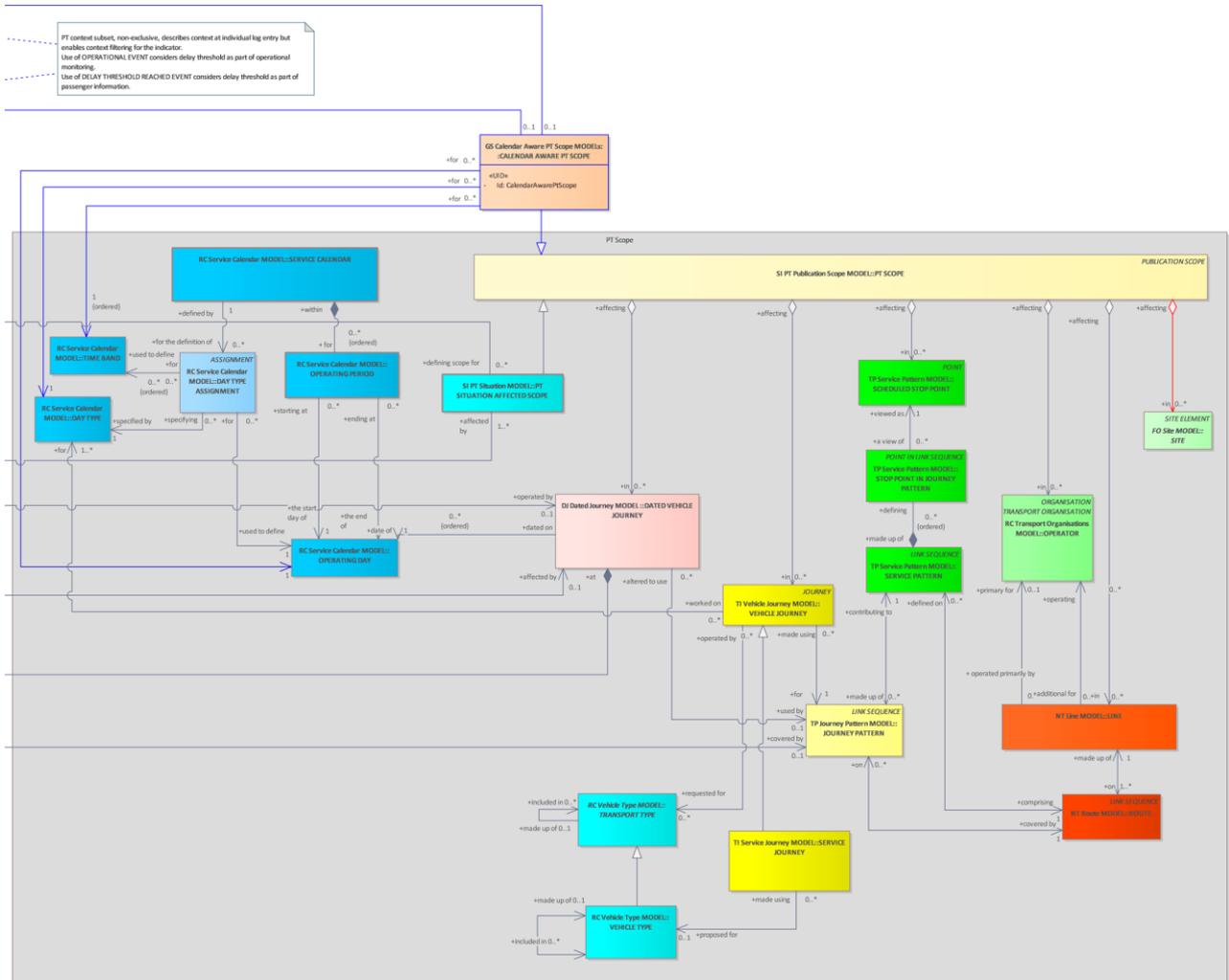


Figure 11 — Number of Late Journeys MODEL PT Scope

The next MI SP Late Journeys Aggregations MODEL builds on the foundation established by the Number of Late Journeys MODEL, which captures individual lateness events through LATE DATED VEHICLE JOURNEY ENTRY and aggregates them into LATE DATED VEHICLE JOURNEY COUNT. While the first model focuses on recording and counting late journeys, this aggregation model introduces a statistical layer that enables deeper analysis and reporting. It leverages generic components like DURATION INTERVAL and OCCURRENCE from the GS Generic Aggregations MODELS to classify delays into time-based intervals and calculate both absolute and relative frequencies. These are specialized into LATE JOURNEY INTERVAL and LATE JOURNEY OCCURRENCE, which group late journeys by delay duration and compute metrics such as counts and percentages. At the top level, the INDICATOR LATE DATED VEHICLE JOURNEY COUNT consolidates these occurrences into KPI-style indicators, contextualized by PT SCOPE to define the operational or reporting domain. Together, these two models form a complete framework: the first ensures accurate capture of lateness events, and the second transforms this raw data into meaningful statistical insights for performance monitoring and service quality evaluation.

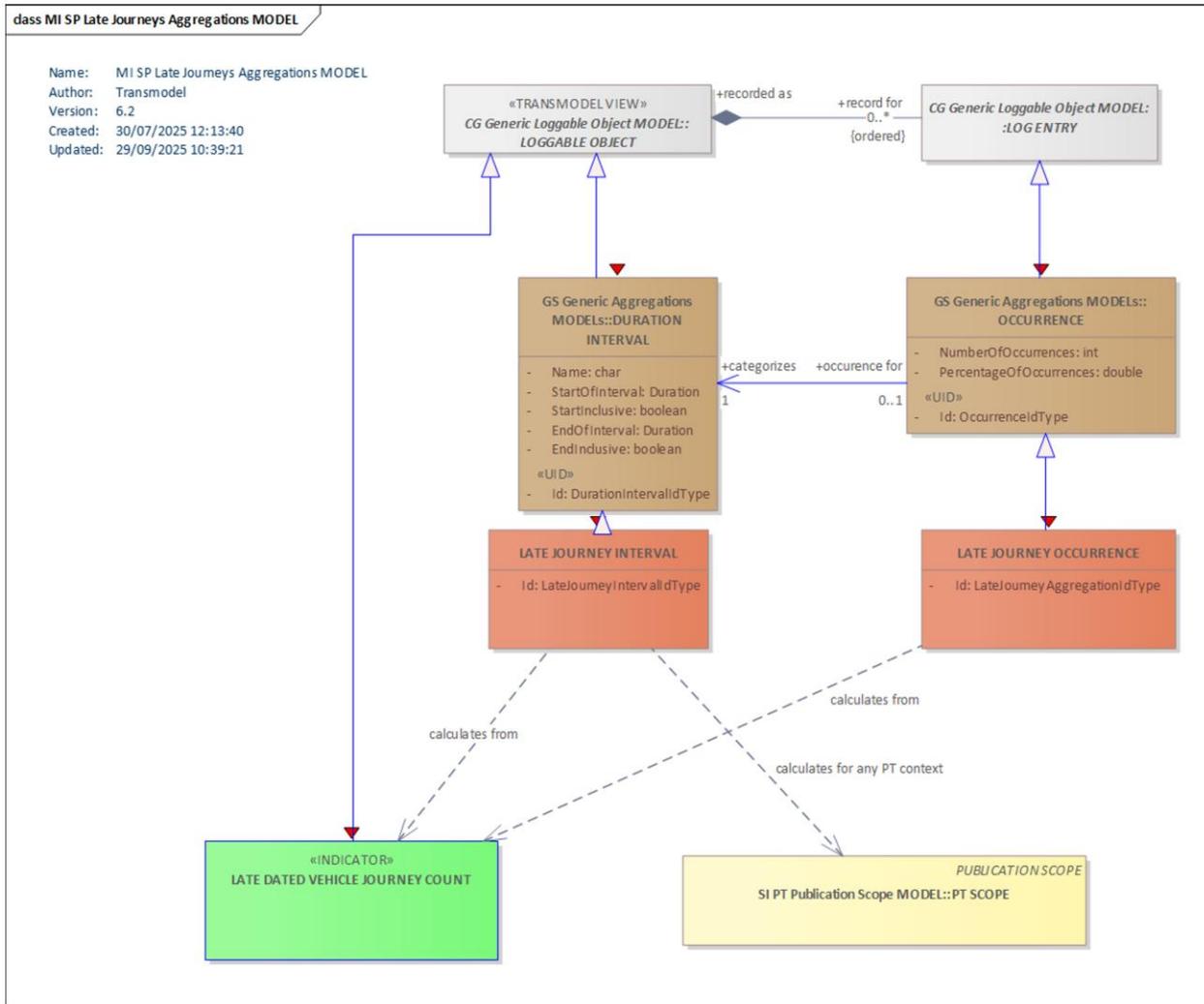


Figure 12 — Late Journeys Aggregations MODEL

9.6 Use case EFFICIENCY-014: Number of cancelled journeys

9.6.1 Use case modelling

This use case reports the number of cancelled journeys, and associated reasons (number of cancellations per reason type), on a predefined period base. It focuses on how public transport cancellations are recorded, aggregated, and published for planned and dated vehicle journeys. It connects cancellation data to operational events, journey monitoring, service calendars, vehicle types, and operators.

The model defines the conceptual structure for capturing and aggregating journey cancellations. At its core is the CANCELLED DATED VEHICLE JOURNEY ENTRY, which records individual cancellations with identifiers and contextual details such as the associated DATED VEHICLE JOURNEY, the reason for cancellation, and its operational scope. These entries are aggregated into CANCELLED DATED VEHICLE JOURNEY COUNT, enabling performance indicators for service reliability. The model links cancellations to PUBLIC TRANSPORT SITUATION, allowing each cancellation to be classified as a situation with a cause and potential operational consequences. It also references SERVICE CALENDAR and OPERATOR, ensuring that cancellations are analysed within the correct temporal and organizational context. This structure supports both operational monitoring and KPI reporting by providing a standardized way to log, classify, and summarize cancellations across the network.

The following figure shows conceptual attributes for the number of cancelled journeys model elements:

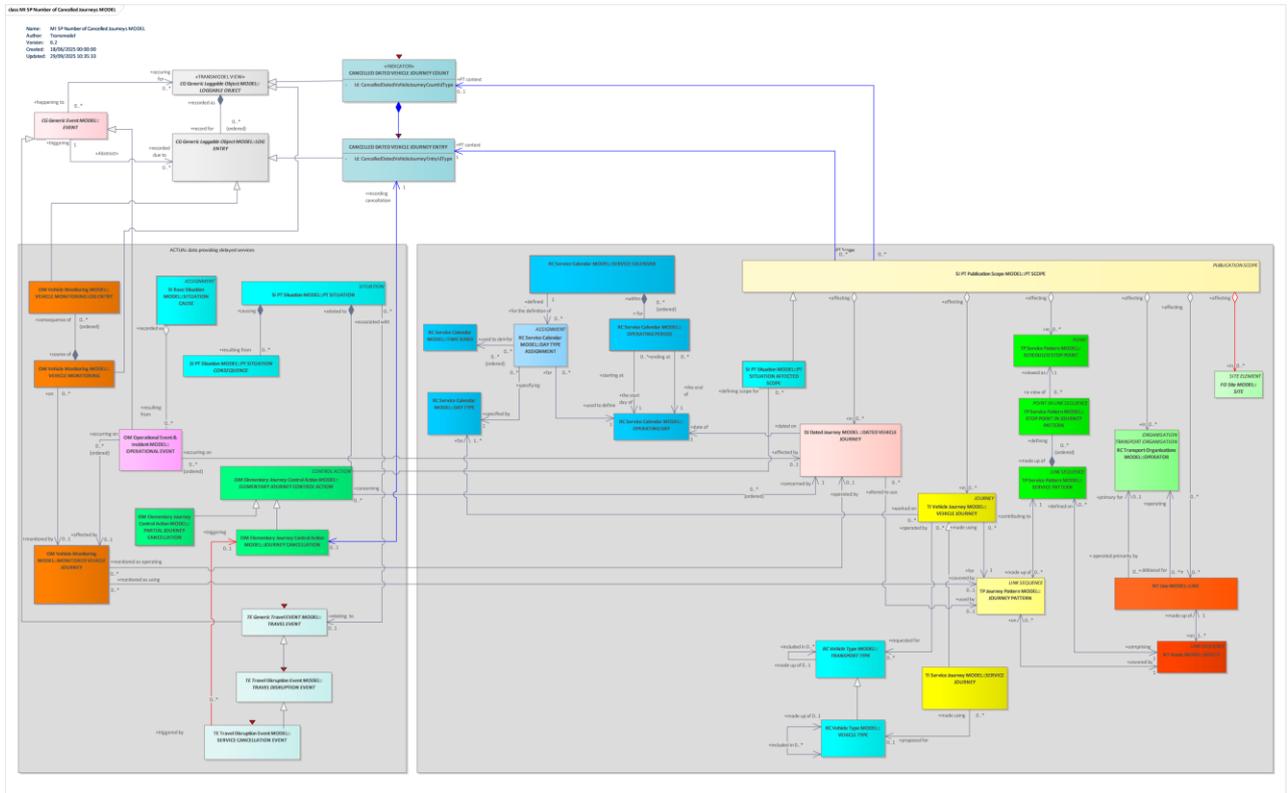


Figure 13 — Number of Cancelled Journeys MODEL

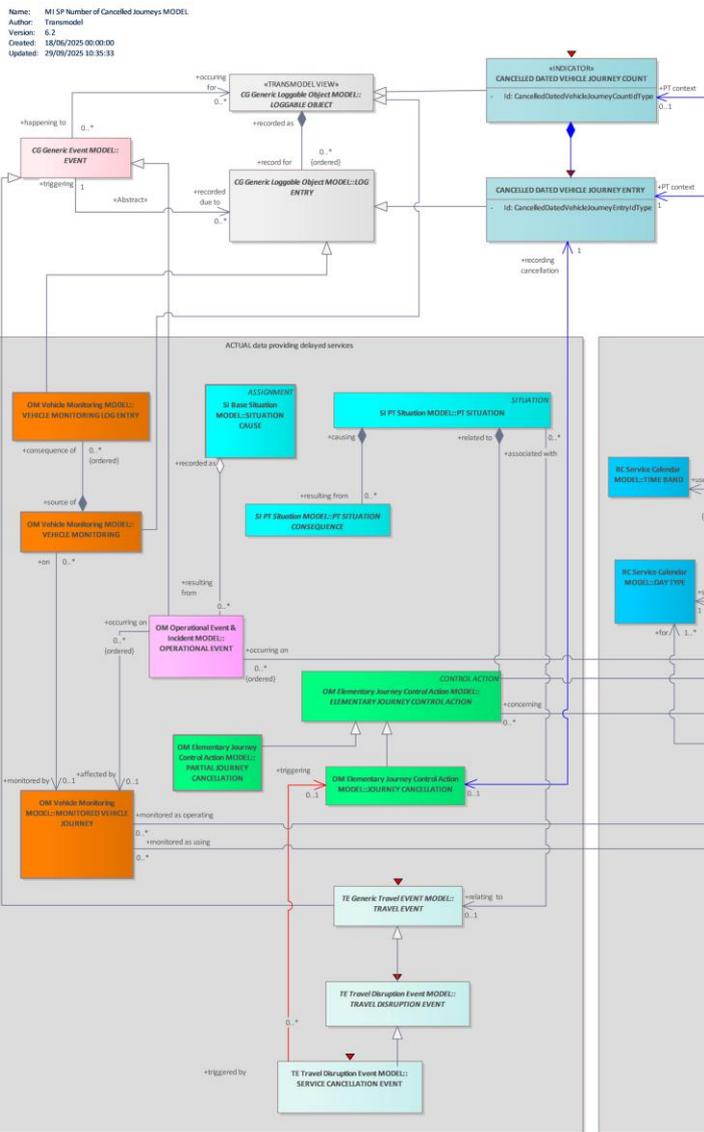


Figure 14 — Number of Cancelled Journeys MODEL ACTUAL data providing delayed services

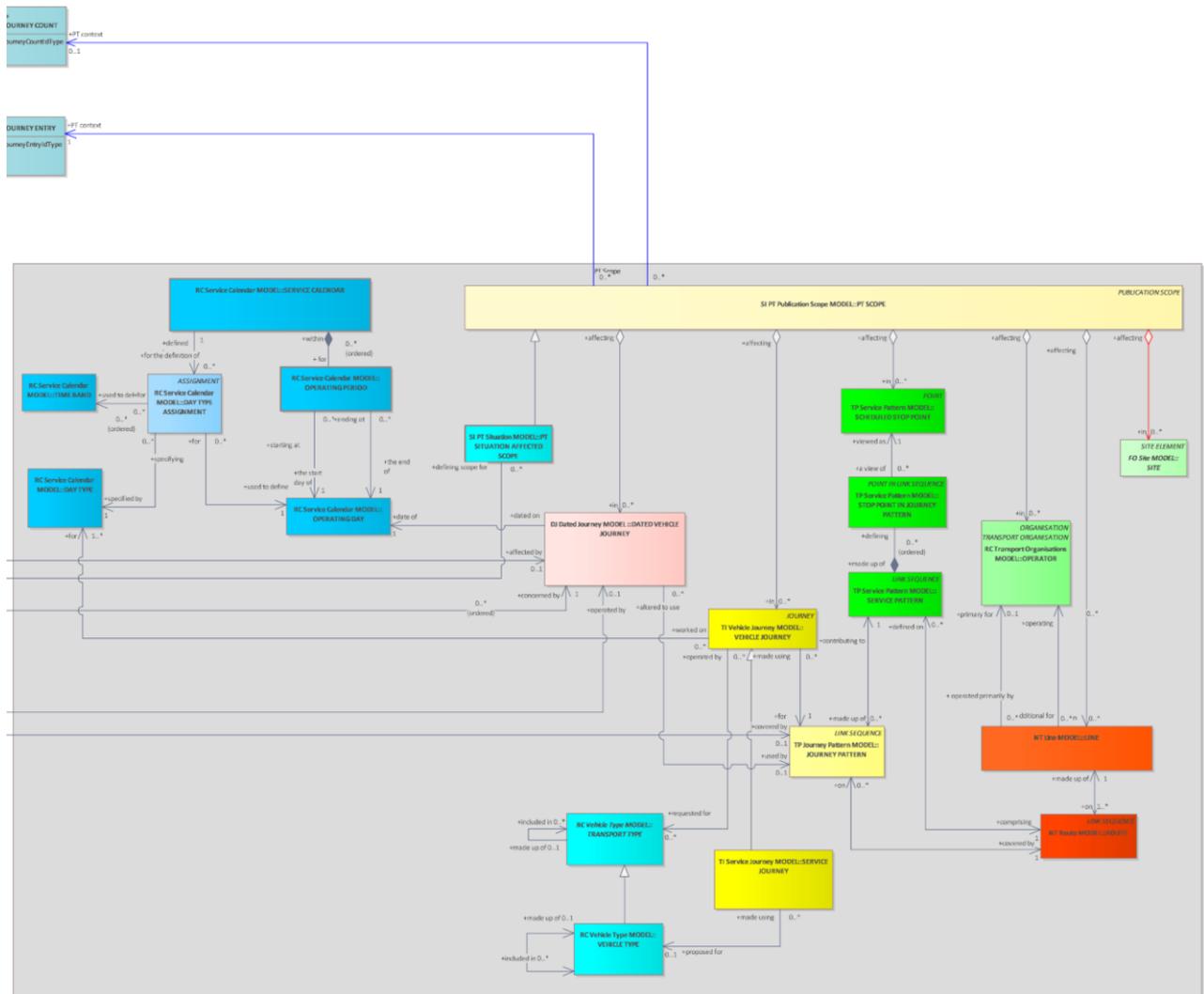


Figure 15 — Number of Cancelled Journeys MODEL PT Scope

The MI SP Cancelled Journeys Aggregations MODEL extends the earlier MI SP Number of Cancelled Journeys MODEL, which records each CANCELLED DATED VEHICLE JOURNEY ENTRY and totals them in CANCELLED DATED VEHICLE JOURNEY COUNT by adding a statistical layer that groups, summarizes, and contextualizes those raw cancellations. Using the generic building blocks from GS GENERIC AGGREGATIONS MODELS, it represents cancellations as OCCURRENCE data points (e.g., with NumberOfOccurrences and PercentageOfOccurrences) and buckets them into analysis periods via an INTERVAL concept (e.g., time buckets such as hour/day/week or other reporting windows). These generic types are specialized for the domain into CANCELLED JOURNEY OCCURRENCE (counts/rates of cancellations for a scope) and, where applicable, a CANCELLED JOURNEY INTERVAL to explicitly define the aggregation window. At the KPI level, an INDICATOR CANCELLED DATED VEHICLE JOURNEY COUNT consolidates the occurrences into publishable metrics, parameterized by PT SCOPE (publication/operational scope) so the same statistics can be sliced by OPERATOR, LINE, SERVICE CALENDAR, or other public transport contexts. In short, the first model captures cancellations; this aggregation model analyses them—turning raw entries into interval-based counts and percentages that drive reliability KPIs and performance dashboards.

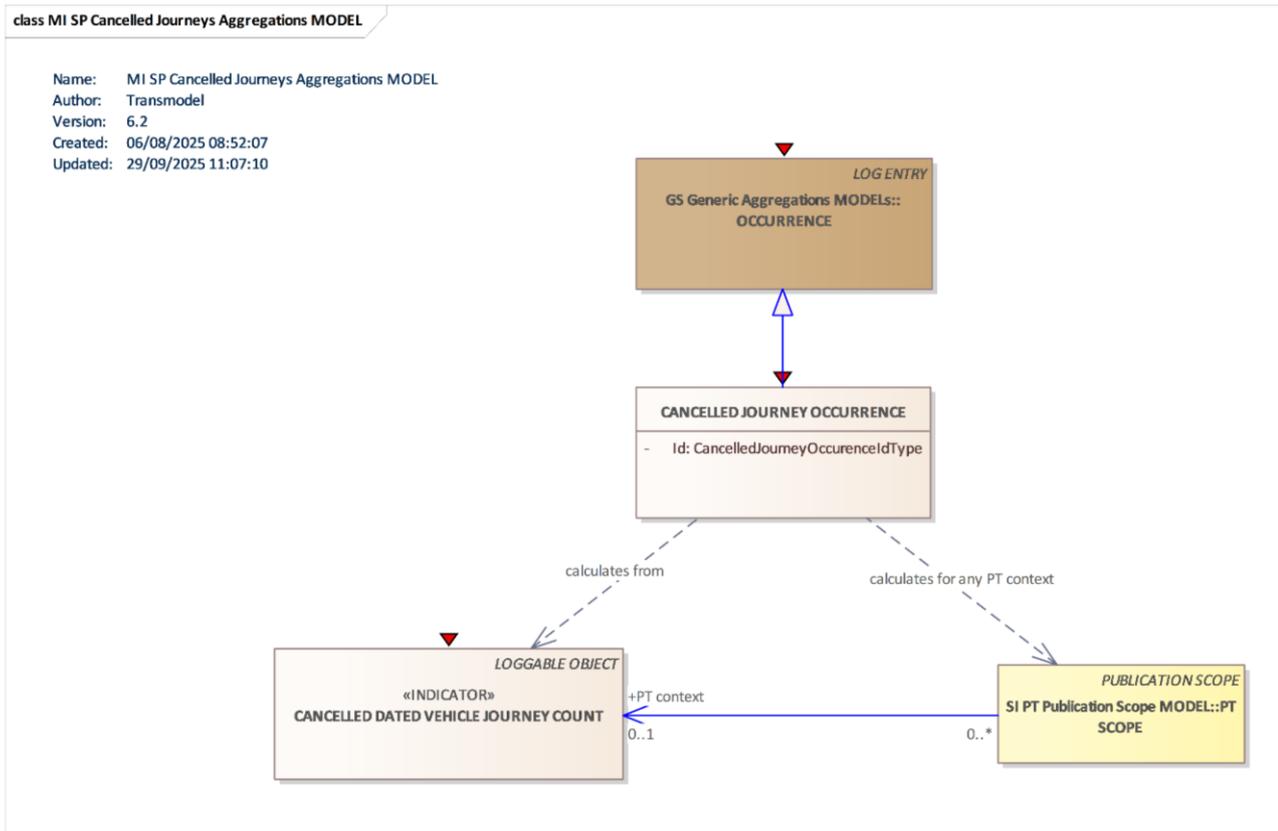


Figure 16 — Cancelled Journeys Aggregations MODEL

9.7 Use case DEMAND-001-001: Expected number of passengers

9.7.1 Use case modelling

This use case reports the expected number of passengers, for a predefined period, typically a year. It is used to evaluate the expected demand for the reference service contract. It focuses on how expected occupancy and passenger count data are structured, exchanged, and associated with operational and planning entities. The use case uses a comprehensive data structure for modelling, predicting, and reporting public transport passenger volumes. It links vehicle types, routes, journey patterns, schedules, organisations, and occupancy data to support planning, analysis, and operational reporting for public transit systems. Relationships clarify how expected passenger numbers are calculated and contextualized within broader transport management and publication frameworks.

The model defines how anticipated passenger demand is represented and structured within the OpRa based framework. At its core, the model introduces the EXPECTED NUMBER OF PASSENGERS indicator, which quantifies forecasted demand for a given service scope and time frame. Supporting this are classes such as EXPECTED PASSENGERS COUNT and EXTERNAL EXPECTED PASSENGER COUNT, which allow differentiation between internally calculated forecasts and externally sourced estimates. These elements are grouped under US Service Demand MODELS, ensuring alignment with broader usage and demand concepts. The model captures essential attributes like expected passenger volumes, temporal applicability (e.g., day, period, or interval), and contextual dimensions such as SERVICE JOURNEY, OPERATOR, or network segment. By structuring expected demand in this way, the model enables integration with planning processes, KPI frameworks, and comparative analysis against actual recorded usage, supporting both operational planning and strategic decision-making.

is manual but guided by a set of transformation rules and experiences from NeTEx and SIRI, still ensuring traceability between the conceptual specification and the physical schema.

This approach guarantees that any evolution or correction in the business semantics is reflected unambiguously in the technical implementation. Furthermore, model-driven design supports tool-assisted validation, code generation, and documentation, and facilitates future harmonization with other standards in the Transmodel family.

10.3 Open Implementation and Technology Use

Data encoded as OpRa can be transferred either with a XML file, and/or with API calls in a similar approach to that taken with the implementation of SIRI in EN 15531.

It is recommended that the exchange protocol defined in EN 15531:2 is used, as this is common to other Transmodel based standards.

10.4 Summary of Modelling Approach

The OpRa conceptual modeling strategy is to provide a systematic, reusable, and interoperable set of data structures reflecting the operational reality of public transport services. The top-down modeling approach starts with domain analysis and identification of information needs (via modelled use cases), formalizes these requirements in a platform-agnostic UML conceptual model, and then transforms and constrains the model according to XSD's capabilities and validation mechanisms.

A key principle is maximizing reuse: wherever possible, OpRa leverages and reuses definitions from Transmodel, NeTEx, and SIRI, ensuring semantic equivalence and compatibility. Physical mapping choices (e.g., use of container elements, reference patterns, and naming conventions) are standardized across the schema and documented. Additionally, the modelling approach explicitly addresses the handling of data versioning, validity conditions, and responsibility data, which are vital for operational use cases.

Mapping from the OpRa conceptual model to its physical XSD representation follows a defined set of transformation guidelines. Each UML class is mapped to a complexType or element in XSD, with inheritance resolved according to XSD capabilities. Relationships and associations are realized via containment or referenced elements, depending on cardinality and business semantics. Enumerations, constraints, and metadata are translated to their XSD equivalents, and shared data types (e.g., identifiers, timestamps, multilingual strings) are reused or imported from common modules (such as NeTEx or SIRI schema libraries).

10.5 Common OpRa Framework

This section describes the common OpRa Framework that is shared by all OpRa functional parts. The framework has two different aspects.

Common framework mechanisms: The framework provides mechanisms for common aspects of all OpRa objects that are needed for effective data exchange, such as versioning, validity, grouping, and responsibility tracking. The mechanisms, implemented through common supertypes and containers, and specialised in the various OpRa functional modules, can be understood and implemented uniformly for all OpRa components, rather than on an ad-hoc basis.

Reusable components: Certain common components, for example explicit frames (i.e. RAW DATA FRAME), generic information and statistics (i.e. DURATION INTERVAL, OCCURRENCE), that are used in many functional parts of OpRa are defined in separated Opra packages. OpRa also reuses components from the public transport network topology, timing information and vehicle scheduling, and operations monitoring and control.

The Versions & Validity model: describes the successive versions of data elements and the conditions to be attached to elements to precisely know when they should be used these are re-used from NeTEx.

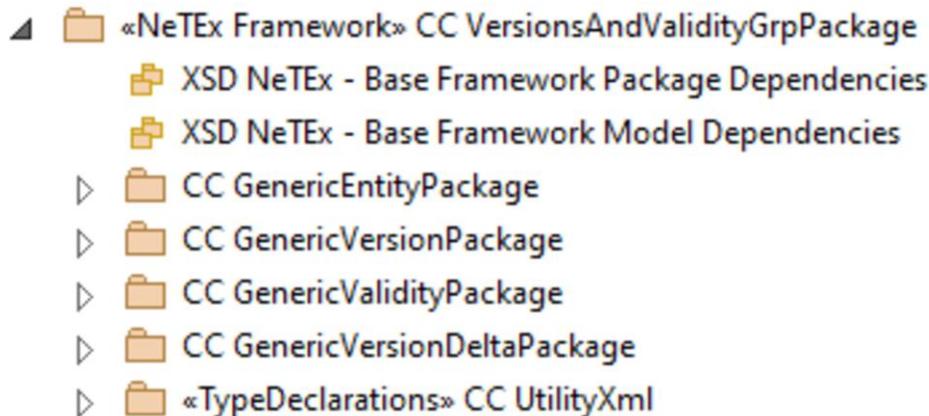


Figure 18 — NeTEx Versions & Validity MODEL packages

The Responsibility model: describes the stakeholder organisations related to the objects that possibly can be exchanged using OpRa and expresses the type of responsibility or role they have with respect to these objects.

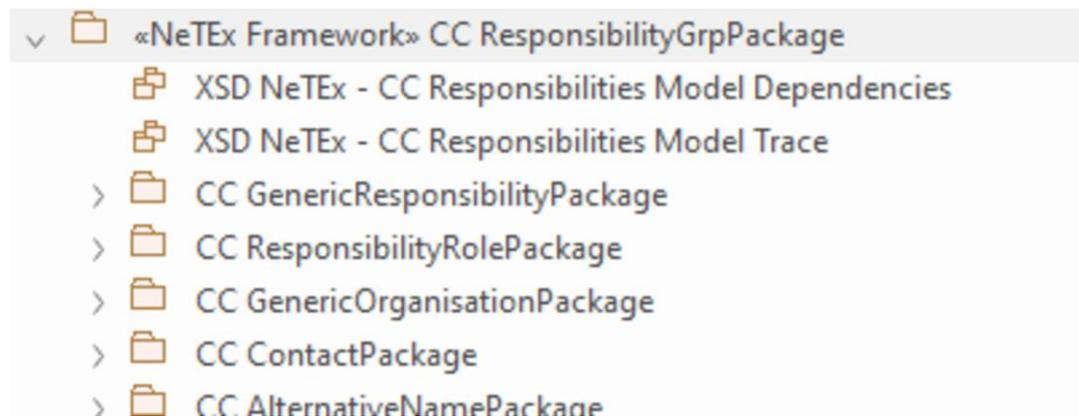


Figure 19 — NeTEx Responsibility MODEL packages

10.6 Implementing Transmodel framework features in OpRa

Transmodel includes a generic mechanism for several commonly required features such as versioning and grouping. Because OpRa describes a physical implementation of Transmodel in XML, defined using a W3C XSD schema, which requires strong static typing to have the benefits of the built in validation, it makes a specific concrete implementation of certain of the mechanisms which are represented as generic types in Transmodel. This can be seen in the following section on VALIDITY and VERSION FRAMES. The Physical implementation is thus more specific and restrictive than the generalised conceptual formation but has the advantage of being easier to implement and can be described more concretely, making it in some ways easier to understand. Thus, rather than having only a general-purpose frame to organise data elements for exchange, OpRa by preference uses several explicit specialisations of the Frame, each of which is restricted to specific types of content.

10.7 Versions & Validity

10.7.1 Introduction

Information systems for public transport operation typically require the exchange of many different types of data, produced by different organisations or operating divisions, and are subject to a multistage lifecycle from planning through to production and realisation in real-time. These data are continuously evolving and are subject to a variety of different validity conditions as to when they are current, and as to which data is needed for a particular purpose. OpRa includes uniform version and validity mechanism to address these requirements; the mechanisms are part of the OpRa framework and that can be applied to all data elements throughout their various lifecycles.

The versioning model allows successive versions of data elements to be identified, allowing the fine-grained identification of just those elements that have changed, and the auditing of changes. All references can also be versioned so that for composite datasets that comprise several related elements it is possible to be precise as to which version of each element is required. The versioning model also allows schemes where the responsibility for maintaining different parts of the data is split among several organisations and systems, each providing its partial data separately. In this case, references to external data are not explicitly versioned, but instead the correct version of the different referenced entities are deduced from validity conditions when combining the data.

A version frame mechanism provides a versionable container that allows a coherent set of related elements to be exchanged. Since pragmatically actual systems that contain data to be exchanged differ in the sophistication of their support for versioning, the mechanisms are designed so that they may be used either just in a course grained manner at the level of the whole dataset, or if support is available, in a more powerful way at the level of the individual data element.

The validity model allows conditions to be attached to elements as to when they are current or the circumstances in which they should be used. Validity conditions can be attached to specific elements and, through version frames, to whole sets of objects so that it is possible to be explicit about the exact conditions governing the coherence and relevance of data. This makes it possible for systems to express the currency conditions for data they require and to describe the validity of data that is returned by a system.

10.7.2 Version & Validity – Model Dependencies

The versioning mechanisms are part of the core OpRa framework and are provided by common set of modules that are referenced by all other OpRa modules. The fundamental models are shown in the following diagram and described in detail in the following sections.

The ENTITY model describes the OpRa basic object structure.

The VERSIONING model adds in version control elements and attributes.

The VERSION FRAME model provides VERSION FRAMES with which to group multiple instances of versions of entities that make up a coherent version set.

The RESPONSIBILITY model adds in metadata for ENTITY ownership and roles for data management.

The VALIDITY package defines generic validity conditions for use in the framework.

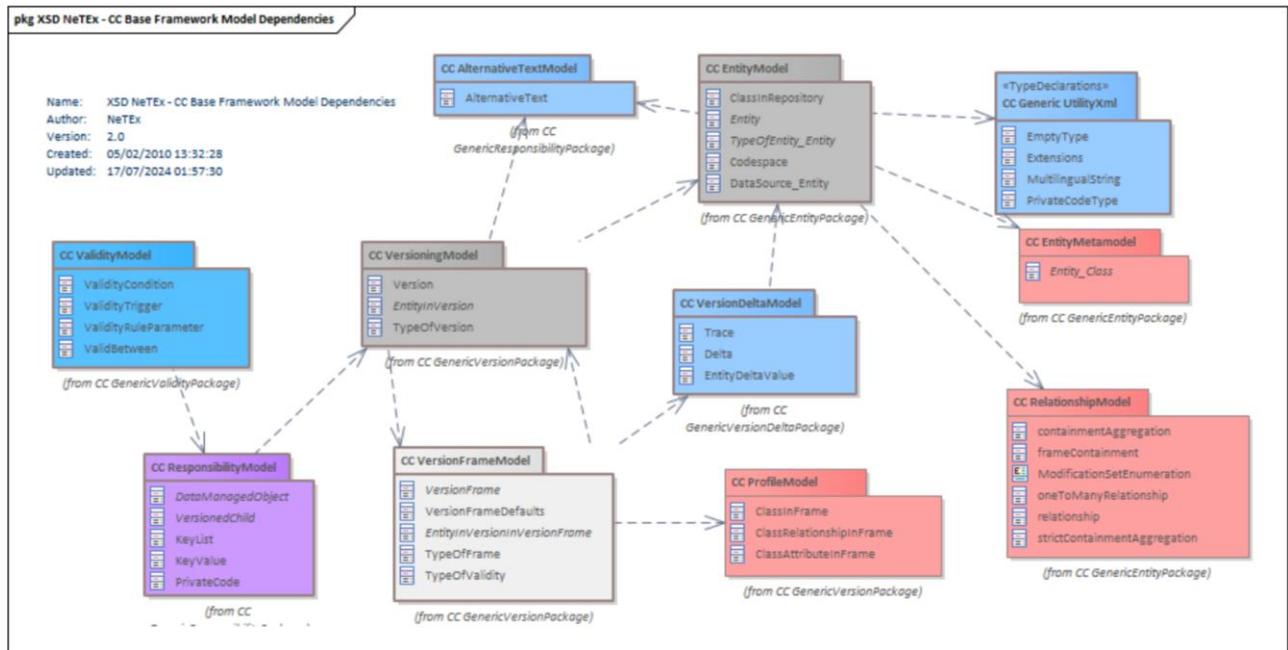


Figure 20 — Version & Validity – Model Dependencies (UML)

The sections below describe the common base structure of the objects that can be exchanged with OpRa.

10.7.2.1 Generic ENTITY – Conceptual MODEL

NOTE The following explanations use excerpts from Transmodel.

A OpRa document containing exchanged data will comprise specific versions of objects that are passed between different systems. Each system shall hold its own representation of the object and may also retain a history of different versions.

The ENTITY represents an actual object instance of data present in an exchanged dataset. An ENTITY may represent any instance of a CLASS IN REPOSITORY, corresponding to an instance of the object as stored in a specific database. All NeTex derived objects are formal descendants of ENTITY.

An ENTITY is linked to a DATA SOURCE which is used to identify the system having produced the version of the ENTITY in the exchanged dataset. Each ENTITY may have its own DATA SOURCE.

CLASSES IN REPOSITORY can be grouped into sets of coherent versions using a CLASS IN FRAME. CLASS IN REPOSITORY and CLASS IN FRAME are part of the Transmodel conceptual model and help to make clear the difference between an instance of an object for exchange (which is in effect always a particular version of an ENTITY) and an instance in a specific database (which may include multiple versions of an ENTITY). However, since NeTex is only concerned with the exchange model as serialised into an XML schema, and not with the repository structures, only the ENTITY and DATA SOURCE are realised in the OpRa XSD schema.

The TYPE of ENTITY defines a set of subcategories that can be used to make arbitrary classifications of a specific ENTITY (thus it is really a “category of ENTITY” rather than a class or type). TYPE OF ENTITY is an abstract mechanism that is present in Transmodel to indicate the possibility of categorisation. Actual OpRa objects generally have a more specific categorisation, e.g. TYPE OF POINT, TYPE OF SERVICE, etc. that specifies a category that is specific to the ENTITY type. TYPE OF ENTITY may be implemented in two different ways.

As a closed list of values as specified by an enumeration. The values are fixed and intended to be strictly standardised in the XML. In this case the XML element will be named xxxType, for example, **StopPlaceType, QuayType**.

As an open list of values to which an implementer may add additional values. For example, TYPE OF POINT, TYPE OF PLACE, TYPE OF SERVICE, etc. In this case the XML element will be named TypeOfxxxx for example **TypeOfPoint, TypeOfPlace, TypeOfService**.

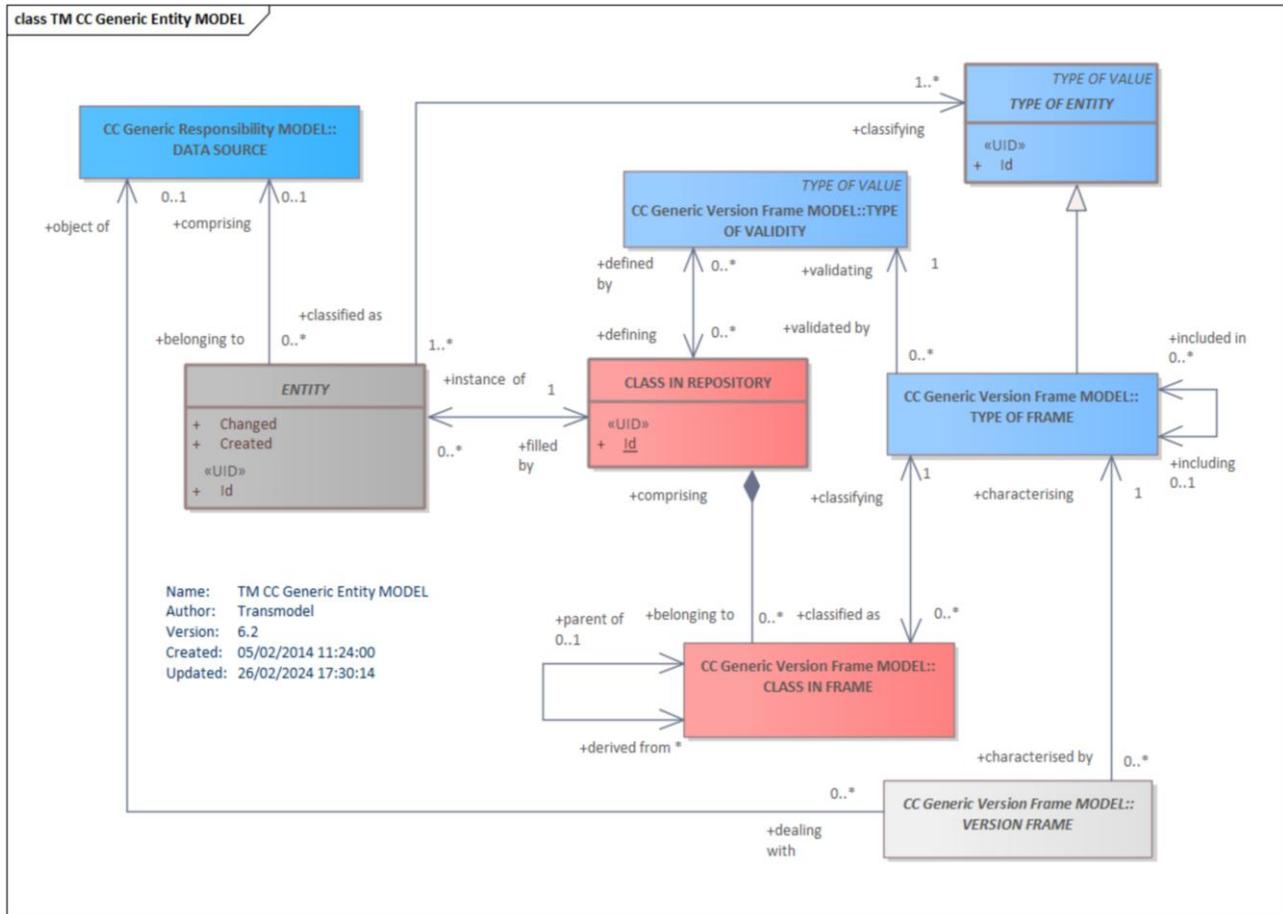


Figure 21 — Generic Entity – Conceptual MODEL (UML)

10.7.2.2 Generic Entity – Physical Model

10.7.2.2.1 General

The physical Generic Entity Model has classes corresponding to those of the conceptual model, but only the ENTITY and DATA SOURCE and CODESPACE elements are implemented in the XSD, since CLASS IN REPOSITORY etc. relate to the database of the participating system, not the exchange format.

An extra object is added in the physical model: the Codespace. Codespace is similar to XML concept of a Namespace that enables the names of elements and attributes in an XML schema to be unique, using the generic W3C domain mechanism. We use it also to ensure that the names of instances of elements in an XML document to be unique – see examples later below. This allows data elements from different countries, different regions, and different organisations to be combined without clashes. Having an explicit Codespace object makes it possible for applications to reflect on and compute over the Codespaces in a generic manner. The Codespace is used for two purposes: (i) to give a unique context within which to identify objects uniquely; (ii) to manage the distributed allocation of new identifiers to new instances of objects. See ADMINISTRATIVE ZONE later.

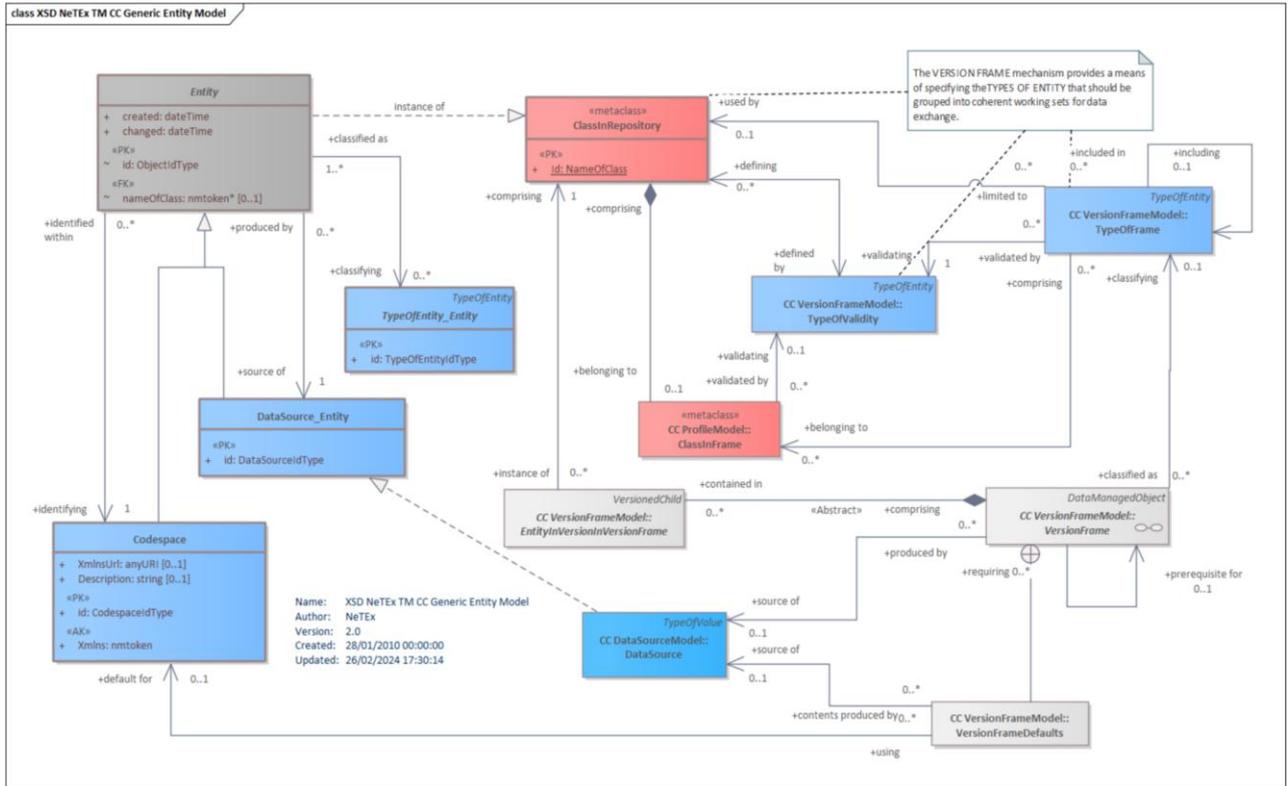


Figure 22 — Generic Entity – Physical Model (UML)

10.7.2.2.2 Codespace – Physical Model

The following diagram shows how a code space may be assigned to a specific ADMINISTRATIVE ZONE.

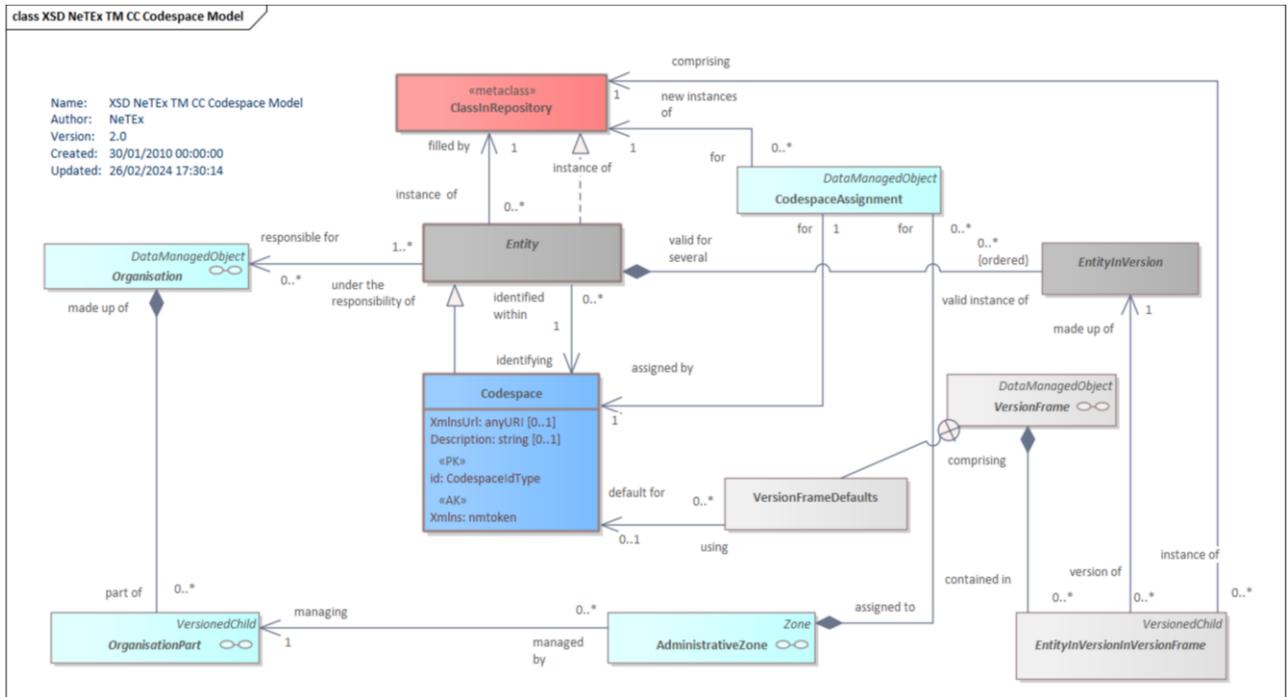


Figure 23 — Codespace – Physical Model (UML)

10.7.2.3 Generic Entity – Attributes and XSD

10.7.2.3.1 Entity – Model Element

An ENTITY is any data instance to be managed in an operational Version Management System. When several data sources coexist (multimodality and/or interoperability), an ENTITY has to be related to a given DATA SYSTEM in which it is defined

Table 12 — Entity - Element

Classification	Name	Type	Cardinality	Description
«atr»	nameOfClass	<i>xsd:NMTOKE</i> <i>N</i>	0:1	Name of class of which this is an instance. Used for reflection.
«PK»	id	<i>ObjectIdType</i>	1:1	Identifier of ENTITY. Unique within CODESPACE.

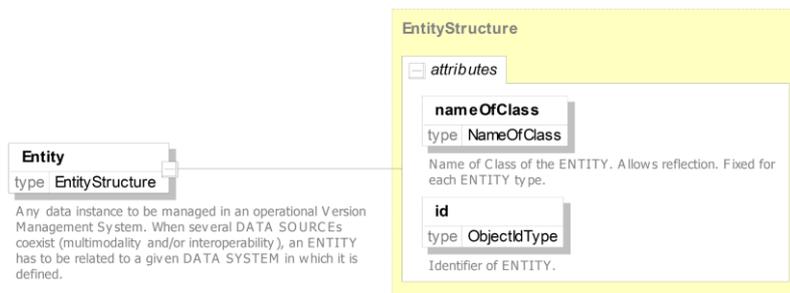


Figure 24 — Entity - XSD

10.7.2.3.2 Codespace – Model Element

A CODESPACE defines the context within which an identifier of an object in a document is unique. The use of CODESPACES in OpRa also allows an economical encoding of identifiers in an XML document so that a common prefix does not have to be repeated on each element in the document.

One or more CODESPACES can also be associated with each ADMINISTRATIVE ZONE to reserve their use of that namespace and a prefix or range value for the allocation of identifiers to new entities within that zone.

A CODESPACE is specified as a path expression by using internet domains, through IANA (Internet Assigned Numbers Authority), which provides a mechanism for registering global uniqueness. For example *tfl.gov.uk*, *bahn.de*, *ratp.fr*, *foo.com*, or *sbb.de*. These can be declared in a Codespace element **:xxx="path"**

Each Codespace has an identifier which may be used in the document using a **Codespace** declaration on a VERSION FRAME.

EXAMPLE CODESPACE declaration:

```
<CompositeFrame version="any" id="mybus:CompositeFrame:CF1">
  <!-- ===== CODESPACES===== -->
  <codespaces>
    <Codespace id=" napt ">
      <Xmlns> napt </Xmlns>
      <XmlnsUrl>http://naptan.org.uk/naptan</XmlnsUrl>
      <Description>UK NaPTAN </Description>
    </Codespace>
  </codespaces>
</CompositeFrame>
```

```

<Codespace id=" era ">
  <Xmlns> era </Xmlns>
  <XmlnsUrl>http://era.org.eu/</XmlnsUrl>
  <Description>European Rail Authority</Description>
</Codespace>
</codespaces>

```

CODESPACE use:

```

id=napt:4701234567, ref= napt:4701234567
id=era:4501234345, etc.

```

An application processing a document is expected to understand any rules peculiar to interpreting the identifiers in a specific **Codespace**. For example NaPTAN stop identifiers in the naptan codespace have the structure '999 0 XXXXXXXXXX' where '999' is an area prefix, '0' is fixed and 'XXXXXXXXXX' is a number unique within the area. The XML will not enforce the rule but an implementation that is aware of the codespace is able to do so.

Table 13 — Codespace - Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>Entity</i>	::>	CODESPACE inherits from ENTITY.
«PK»	id	<i>CodespaceIdType</i>	1:1	Identifier of CODESPACE. Unique within document.
«AK»	Xmlns	<i>CodespaceIdType</i>	1:1	Codespace prefix, unique within a given XML document e.g. 'napt'
	XmlnsUrl	<i>xsd:anyURI</i>	0:1	Codespace path. Globally unique. For example, http://naptan.org.uk/naptan or http://vdv.de/vdv/haltstelle/
	Description		0:1	Description of CODESPACE.

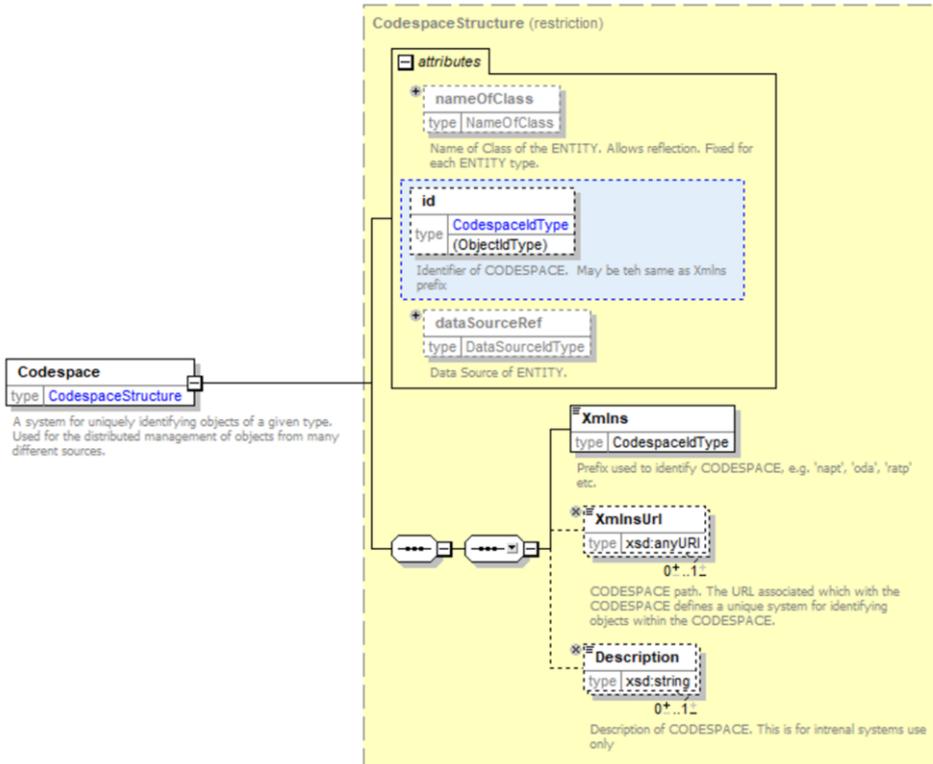


Figure 25 — Codespace - XSD

10.7.2.4 TypeOfEntity - Model Element

Classification of ENTITIES, for instance according to the domain in which they are defined or used.

In the OpRa physical model where there is a need to categorise an element specifically according to its type, a specialized "type of type" is provided, for example, *TypeOfPlace* implements the TYPE OF PLACE entity from the Conceptual model; *TypeOfPoint* implements the TYPE OF POINT entity from the Conceptual model, etc. In some cases, the *TypeOfXxx* elementType is implemented as a closed list of enumerated values, in which case it will be named *xxxType*, for example by *StopPlaceType*.

Table 14 — TypeOfEntity - Element

Classification	Name	Type	Cardinality	Description
::>	::>	<u>TypeOfValue</u>	::>	TYPE OF ENTITY inherits from TYPE OF VALUE.
«PK»	<i>id</i>	TypeOfEntityIdType	1:1	Identifier of TYPE OF ENTITY.

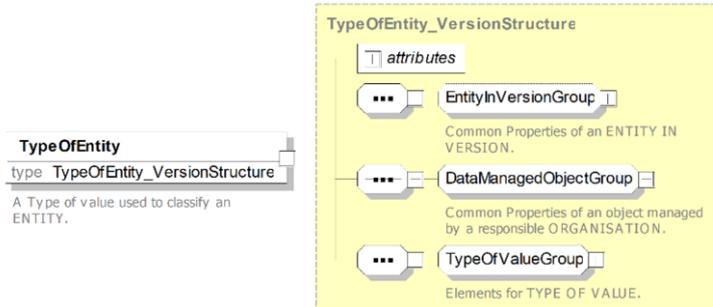


Figure 27 — TypeOfEntity - XSD

10.7.2.4.1 ClassInRepository - Model Element

Any ENTITY name belonging to the repository, e.g. DAY TYPE, PROPERTY OF DAY, TIME BAND, VEHICLE TYPE, DUTY, etc. are relevant instances of CLASS IN REPOSITORY in the context of Version Management.

This is a meta class that can be used to reference the available entity types.

Table 15 — ClassInRepository - Element

Classification	Name	Type	Cardinality	Description
«PK»	<i>id</i>	<i>NameOfClass</i>	1:1	Unique identifier of class of ENTITY. There will only be a single instance, which shall be the class name, e.g. <i>JourneyPattern, StopPlace</i> , etc.

The ClassInRepository is not implemented in the OpRa XSD Schema, however nameOfClass may be specified as an attribute in certain places to facilitate mapping to repositories.

10.7.2.5 XML Examples for generic Entities

10.7.2.5.1 Codespaces & Identifiers - XML Example Fragment

The following example shows the use of CODESPACES to declare uniqueness. In the example the ROUTE LINKS come from one CODESPACE (http://www.foo.eu/stuff, using local prefix 'foo'), and the ROUTE POINTS from two others (http://www.bar.de, using local prefix 'bar', and 'http://www.fum.ch, with local prefix 'fum,'). These can be used to give unique CODESPACES for each identifier. Note that two of the points (A and C) have the same number '01', but are rendered unique by different CODESPACES. The prefixes are arbitrary and local to the document. The urls are persistent and fixed.

Example

```
<PublicationDelivery version="1.0" xsi:schemaLocation="http://www.netex.org.uk/netex
/xml/NetEx_publication.xsd" xmlns="http://www.netex.org.uk/netex"
xmlns:siri="http://www.siri.org.uk/siri" xmlns="http://transmodel-cen.eu/opra/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:foo="http://www.foo.eu/stuff" xmlns:bar="http://www.bar.de/"
xmlns:fum="http://www.fum.ch/">
  <PublicationTimestamp>2010-12-17T09:30:47.0Z</PublicationTimestamp>
  <dataObjects>
    <CompositeFrame version="1" id="bar:CF01">
      <ServiceFrame version="1" id="bar:SF01">
        <Name>Stops for Winter timetable for route 24 </Name>
        <routePoints>
```

```

<RoutePoint version="any" id="bar:01">
  <Name>Point in Road near Stop A</Name>
  <Location>
    <Longitude>53.0000</Longitude>
    <Latitude>0.1000</Latitude>
  </Location>
</RoutePoint>
<RoutePoint version="any" id="bar:02">
  <Name>Point in Road near Stop B</Name>
  <Location>
    <Longitude>53.0000</Longitude>
    <Latitude>0.2000</Latitude>
  </Location>
</RoutePoint>
<RoutePoint version="any" id="fum:01">
  <Name>Point in Road near Stop C</Name>
  <Location>
    <Longitude>53.0000</Longitude>
    <Latitude>0.3000</Latitude>
  </Location>
</RoutePoint>

</routePoints>
  <routeLinks>
    <!-- ===== ROUTE LINKs ==OUTBOUND===== -->
    <RouteLink version="any" id="foo:01">
      <FromPointRef version="any" ref="bar:01"/>
      <ToPointRef version="any" ref="bar:02"/>
    </RouteLink>
    <RouteLink version="any" id="foo:02">
      <FromPointRef version="any" ref="bar:02"/>
      <ToPointRef version="any" ref="fum:01"/>
    </RouteLink>
  </routeLinks>
::>
<PublicationDelivery version="1.0" xsi:schemaLocation="http://www.netex.org.uk/netex
/xml/NeTEx_publication.xsd" xmlns="http://www.netex.org.uk/netex"
xmlns:siri="http://www.siri.org.uk/siri" xmlns="http://transmodel-cen.eu/opra/1.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:foo="http://www.foo.eu/stuff"
  xmlns:bar="http://www.bar.de/"
  xmlns:fum="http://www.fum.ch/">
  <PublicationTimestamp>2010-12-17T09:30:47.0Z</PublicationTimestamp>
  <dataObjects>
    <CompositeFrame version="1" id="bar:CF01">
      <ServiceFrame version="1" id="bar:SF01">
        <Name>Stops for Winter timetable for route 24 </Name>
        <routePoints>
          <RoutePoint version="any" id="bar:01">
            <Name>Point in Road near Stop A</Name>
            <Location>
              <Longitude>53.0000</Longitude>
              <Latitude>0.1000</Latitude>
            </Location>
          </RoutePoint>
          <RoutePoint version="any" id="bar:02">
            <Name>Point in Road near Stop B</Name>
            <Location>
              <Longitude>53.0000</Longitude>
              <Latitude>0.2000</Latitude>
            </Location>
          </RoutePoint>
          <RoutePoint version="any" id="fum:01">

```

```

        <Name>Point in Road near Stop C</Name>
        <Location>
            <Longitude>53.0000</Longitude>
            <Latitude>0.3000</Latitude>
        </Location>
    </RoutePoint>
</routePoints>
    <routeLinks>
<!-- ===== ROUTE LINKs ==OUTBOUND===== -->
    <RouteLink version="any" id="foo:01">
        <FromPointRef version="any" ref="bar:01"/>
        <ToPointRef version="any" ref="bar:02"/>
    </RouteLink>
    <RouteLink version="any" id="foo:02">
        <FromPointRef version="any" ref="bar:02"/>
        <ToPointRef version="any" ref="fum:01"/>
    </RouteLink>
::>

```

Example:

```

<CompositeFrame version="any" id="tbd:cf01" dataSourceRef="mydata:DB770">
    <ResourceFrame version="any" id="tbd:RS01" >
        <codespaces>
            <Codespace id=" napt ">
                <Xmlns>napt</Xmlns>
                <XmlnsUrl>http://www.naptan.org.uk/naptan</XmlnsUrl>
                <Description>UK NaPTAN Stop Place codes</Description>
            </Codespace>
            <Codespace id="tbd">
                <Xmlns>tbd</Xmlns>
                <XmlnsUrl>http://www.intdata.co.uk/data</XmlnsUrl>
                <Description>Other interchange DATA SOURCE </Description>
            </Codespace>
        </codespaces>
        <dataSources>
            <DataSource version="any" id="nptg:NaPTAN" dataSourceRef="nptg:NaPTAN"
>
                <Name>DfT Naptan database </Name>
            </DataSource>
            <DataSource version="any" id="tbd:Xtra_Acc_Data" dataSourceRef="
tbd:Xtra_Acc_Data " >
                <Name>Additional accessibility Data </Name>
            </DataSource>
            <DataSource version="any" id="mydata:DB770" dataSourceRef="
mydata:DB770">>
                <Name>Wigan local data base </Name>
            </DataSource>
        </dataSources>
    </ResourceFrame>
::>...
    <SiteFrame version="any" id="tbd:SiteFrame:SVC01">
        <stopPlaces>
            <StopPlace version="001" created="2006-09-11T15:42:00" changed="2009-
02-26T15:47:00 modification="revise"
                dataSourceRef="nptg:NaPTAN" id="napt:910GWIMBLDN">
                <Name>Wimbledon Rail Station</Name>
                <ShortName>Wimbledon Station</ShortName>
                <types><TypeOfZoneRef ref="napt:GRLS"/></types>
                <AccessibilityAssessment version="any" id="tbd:910GWIMBLDN_Ac1"

```

```

                dataSourceRef=" tbd:Xtra_Acc_Data " >
        <MobilityImpairedAccess>true</MobilityImpairedAccess>
        <limitations>
                <AccessibilityLimitation modification="new"
dataSourceRef="tbd:Xtra_Acc_Data"
                created="2010-05-17T09:30:47Z">
                <WheelchairAccess>true</WheelchairAccess>
                <StepFreeAccess>true</StepFreeAccess>
                <EscalatorFreeAccess>true</EscalatorFreeAccess>
                <LiftFreeAccess>true</LiftFreeAccess>
                </AccessibilityLimitation>
        </limitations>
        </StopPlace>
::>.
        <responsibilitySets>
                <!-- Generalisation of NPTG ability to associate data with an area and
Equivalent to NPTG areas = -->
                <ResponsibilitySet version="001" created="2010-05-17T09:30:47Z"
modification="revise" changed="2010-05-17T09:30:47Z" responsibilitySetRef="napt:RS_nptg"
id="napt:RS_82">
                        <roles>
                                <ResponsibilityRoleAssignment version="001"
created="2010-05-17T09:30:47Z" id="napt:RS_82_01">
                                        <Description>490 London - Managed
centrally</Description>
                                        <DataRoleType>collects</DataRoleType>
                                        <ResponsibleOrganisationRef version="any"
ref="tbd:Org_TL001"/>
                                        <ResponsibleAreaRef version="any" ref="napt:82"/>
                                </ResponsibilityRoleAssignment>
                        </roles>
                </ResponsibilitySet>
                <ResponsibilitySet version="001" created="2010-05-17T09:30:47Z"
responsibilitySetRef="napt:RS_nptg" id="napt:RS_110">
                        <roles>
                                <ResponsibilityRoleAssignment version="001"
created="2010-05-17T09:30:47Z" id="napt:RS_110_01">
                                        <Description>910 UK Rail - Managed
centrally</Description>
                                        <DataRoleType>collects</DataRoleType>
                                        <ResponsibleOrganisationRef version="any"
ref="tbd:Org_TD002"/>
                                        <ResponsibleAreaRef version="any" ref="nptg:110"/>
                                </ResponsibilityRoleAssignment>
                        </roles>
                </ResponsibilitySet>
                <ResponsibilitySet version="001" created="2010-05-17T09:30:47Z"
responsibilitySetRef="napt:RS_147" id="napt:RS_147">
                        <roles>
                                <ResponsibilityRoleAssignment version="001"
created="2010-05-17T09:30:47Z" id="napt:RS_147_01">
                                        <Description>940 UK Tram - Managed
centrally</Description>
                                        <DataRoleType>collects</DataRoleType>
                                        <ResponsibleOrganisationRef version="any"
ref="tbd:Org_TL001"/>
                                        <ResponsibleAreaRef version="any" ref="nptg:147"/>
                                </ResponsibilityRoleAssignment>
                        </roles>
                </ResponsibilitySet>
                <ResponsibilitySet version="001" created="2010-05-17T09:30:47Z"
responsibilitySetRef="napt:RS_nptg" id="napt:RS_nptg">

```

```

        <roles>
            <ResponsibilityRoleAssignment version="001"
created="2010-05-17T09:30:47Z" id="napt:RS_nptg">
                <Description>NPTG Coordination of area codes UK
London</Description>
                <DataRoleType>collects validates
distributes</DataRoleType>
                <StakeholderRoleType>operation</StakeholderRoleType>
                <ResponsibleOrganisationRef version="any"
ref="tbd:Org_TD002"/>
                <ResponsibleAreaRef version="any" ref="nptg:UK"/>
            </ResponsibilityRoleAssignment>
        </roles>
    </ResponsibilitySet>
    <ResponsibilitySet version="001" created="2010-05-17T09:30:47Z"
responsibilitySetRef="tbd:RS_123" id="napt:RS_de">
        <roles>
            <ResponsibilityRoleAssignment version="001"
created="2010-05-17T09:30:47Z" id="napt:RS_123_01">
                <Description>Extra STOP PLACE data </Description>
                <DataRoleType>all</DataRoleType>
                <ResponsibleOrganisationRef version="any"
ref="tbd:Org_TD002"/>
            </ResponsibilityRoleAssignment>
        </roles>
    </ResponsibilitySet>
</responsibilitySets>

```

10.7.3 Generic Version

10.7.3.1 General

The sections below describe the versioning mechanism for objects exchanged with OpRa.

NOTE The modelling of versions in OpRa is designed to fulfil the versioning needs of a data exchange mechanism exchanging complex linked objects. It is in effect a version description model, not a model of a version management system. However, it allows for fine grained versioning, and uses a uniform and generic approach that can be used for any kind of complex data object. This versioning mechanism is available on all OpRa elements, but not mandatory, thus allowing legacy systems without any versioning mechanism to use OpRa simply by omitting the versioning attributes. In practice, versioning will often just be done at an aggregate level (e.g. the whole export) and not at the level of the individual data instance.

Public transport data are in a permanent process of evolution; schedule and operational data typically undergo a regular cycle of planning, distribution and execution, whilst reference data describing the network, such as stop and line data, will change if network or physical environment is modified. It is therefore necessary to be able to organise data elements to support such a lifecycle, with multiple versions of a given element being in use concurrently, and different assemblies of data referencing different versions for different purposes. This is achieved in OpRa with VERSIONs and VERSION FRAMES.

10.7.3.2 Generic VERSION – Conceptual MODEL

NOTE The following explanations use excerpts from Transmodel.

Each state of an object, or a set of objects, is called a VERSION. VERSIONs of an object may be consecutive or competitive. Consecutive VERSIONs describe the successive states of an object, whilst

competitive VERSIONs describe an alternative version to use in particular circumstances (as say described by a VALIDITY CONDITION). For example, there may be for a single LINE at the same time competitive versions of the officially scheduled LINE; a simulated LINE (for planning work or for study), and the LINE with alternative ROUTEs (for disruption management for example).

The VERSION describes the identifier and purpose of a version state. The actual version state is described by an instance of ENTITY IN VERSION. Thus, in a given repository or documents there will be a single instance of each Transmodel ENTITY and one or multiple instances of ENTITY IN VERSIONs for that ENTITY; these will be tied together by a common identifier and differentiated by distinct VERSION identifiers. For example an instance of the entity VEHICLE JOURNEY may have multiple VEHICLE JOURNEY IN VERSION instances, a STOP PLACE may have multiple STOP PLACE IN VERSION instances, etc.

The purpose of the VERSION may be categorised with an arbitrary classification using a TYPE OF VERSION, for example planning, scheduled, operational, etc.

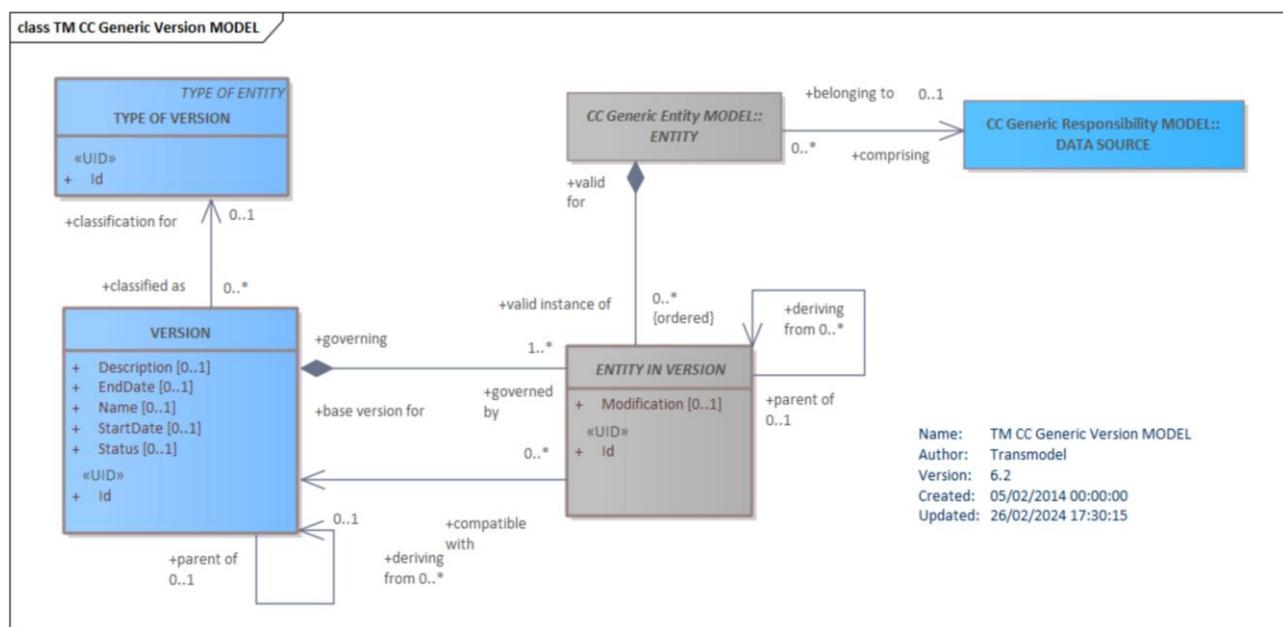


Figure 27 — Generic Version – Conceptual MODEL (UML)

10.7.3.3 Generic Version – Physical Model

10.7.3.3.1 General

The physical Generic Version Model has classes corresponding to those of the conceptual model. In the physical model ENTITY IN VERSION is abstract; no instance of *EntityInVersion* is possible, instead there is a specific subclass of it for each type of concrete NeTEx ENTITY that is exchanged. For example STOP PLACE IN VERSION, VEHICLE JOURNEY IN VERSION, etc. For brevity the 'IN VERSION' is omitted from the names of model elements. Thus Place is short for *PlaceInVersion*, *VehicleJourney* is short for *VehicleJourneyInVersion*, etc.

NOTE W3C XSD schemas do not support generic types so the Entity_Entity is a workaround used to achieve the equivalent mechanism of a container for ENTITY IN VERSION instances of a specific ENTITY type. Although the OpRa physical model and XSD schema do not include specific elements for each specialisation of ENTITY, but if they did the names of such classes would by convention be qualified with “_ENTITY” in order to distinguish them from the corresponding ENTITY IN VERSION. For example, *VehicleJourneyEntity*, *VehicleJourney_Entity*, *TimingLink_Entity*, etc. as shown in the following table.

Table 16 — Examples of ENTITY & ENTITY IN VERSION Names

ENTITY		ENTITY IN VERSION	
Name in Conceptual Model	Name in Physical Model **	Name in Conceptual Model	Name in Physical Model
ENTITY	Entity_Entity	ENTITY IN VERSION	EntityInVersion
[XXX]	Xxx_Entity	[XXX] IN VERSION	Xxx
LINE	Line_Entity	LINE IN VERSION	Line
VEHICLE JOURNEY	VehicleJourney_Entity	VEHICLE JOURNEY IN VERSION	VehicleJourney
TIMING LINK	TimingLinkEntity	TIMING LINK IN VERSION	TimingLink
TIMETABLE FRAME	TimetableFrame_Entity	TIMETABLE FRAME IN VERSION	TimetableFrame
Etc.			

10.7.3.3.3 Version States and the Edit / Version / Release Cycle

The OpRa implementation assumes a conventional Edit / Release / Version / Release lifecycle for versioning each data element within a repository.

When an element is created or modified a new open version of it is created (status = draft). Further changes may be made to the state of the object within the local repository, but it may not be exported.

If it is part of a containing aggregation, for example, a Version Frame, then a new open edition of the containing aggregation also needs to be created (status = draft) and the new version of the contained element shall be released to the new version of the container.

Once all changes are complete the element is versioned (status = versioned), after which no further changes can be made. Only versioned elements can be exchanged. Both contained and container elements are versioned. Contained elements shall be versioned before their container is versioned.

Data instances exported as OpRa data elements should have states consistent with these rules.

10.7.3.4 Version – Attributes and XSD

10.7.3.4.1 Version – Model Element

10.7.3.4.1.1 General

IMPORTANT: The Version Element is DEPRECATED from NeTeX 2.0 onwards. This, is only for the XML Version element (<Version ... >) and not the XML version attribute (as in <Tag id="xxx" version="yyy" ... >) whose usage is now emphasised in the XSD.

Details of the deprecated Version element is provided in Annex C.

10.7.3.4.2 EntityInVersion – Model Element

10.7.3.4.2.1 General

An ENTITY IN VERSION is an ENTITY associated with a given VERSION. ENTITY IN VERSION is restricted by ENTITY IN FRAME.

Table 17 — EntityInVersion – Element

Classification	Name	Type	Cardinality	Description	
::>	::>	<i>Entity</i>	::>	ENTITY ON VERSION inherits from ENTITY.	
«FK»	dataSourceRef	<i>DataSourceIdType</i>	0:1	Data system which originated data instance.	
«atr»	created	<i>xsd:dateTime</i>	0:1	Date and time of creation of ENTITY.	
«atr»	changed	<i>xsd:dateTime</i>	0:1	Date and time of last change to ENTITY.	
«atr», «enum»	modification	<i>ModificationEnum</i>	0:1	Nature of modification. Enumerated value: new revise delete unchanged delta.	
«atr», «FK»	version	<i>VersionIdType</i>	0:1	VERSION number of this instance of the ENTITY IN VERSION. The value may be "any" or a VERSION number adhering to agreed-upon formatting and interpretation rules (representing, e.g., a number or a date). Successive VERSION numbers should be strictly increasing.	
«atr», «enum»	status	<i>VersionStatusEnum</i>	0:1	Status of Entity in VERSION. Enumerated value.	
«atr», «FK»	derivedFromVersionRef	<i>VersionIdType</i>	0:1	Reference to VERSION from which this VERSION of the ENTITY was derived. This is a <i>VersionIdType</i> , the same type as version, thus being a version number.	
«atr», «FK»	compatibleWithVersionFrameVersionRef	<i>VersionIdType</i>	0:1	Version of frame from with which this version of ENTITY is compatible. Assumes Frame of the same Id as current frame.	
«atr», «FK»	derivedFromObjectRef	<i>ObjectIdType</i>	0:1	Identity of object from which this object of ENTITY was derived. Normally the same.	
		CHOICE	0:1	Common Properties of an ENTITY IN VERSION.	
«cntd»	a	validityConditions	<i>ValidityCondition</i>	1:1	VALIDITY CONDITIONs conditioning entity.

					Note: ValidityCondition can be combined and ANDED (all the conditions must be fulfilled at the same time)
«cntd»	b	ValidBetween	<u>ValidBetween</u>	1:*	OPTIMISATION. Simple version of a VALIDITY CONDITION. Comprises a simple period. NO UNIQUENESS CONSTRAINT.
«cntd»		alternativeTexts	<i>AlternativeText</i>	0:*	ALTERNATIVE TEXTs associated with ENTITY.

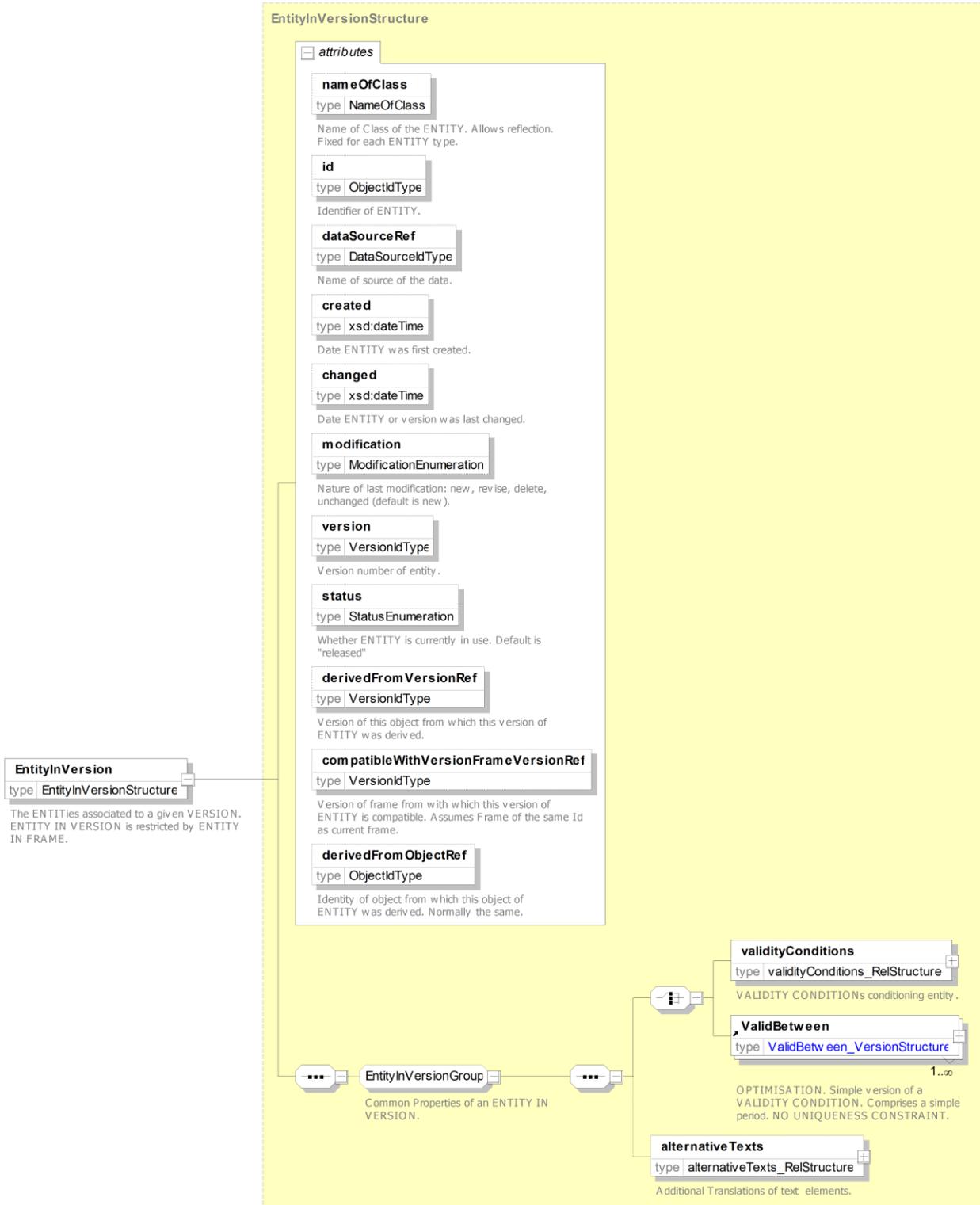


Figure 29 — EntityInVersion – XSD

10.7.3.4.2.2 #Modification – Allowed Values

Modification explains why this ENTITY version was distributed. The following table shows the allowed values for Modification (ModificationEnumeration).

Table 18 — Modification - Allowed Values

Value	Description
new	This is the first version of a new object.
revise	This is an update of a previously defined object.
delete	This version is to point out that the object was deleted.
unchanged	This object has not changed since the previous version.
delta	This version contains only the changes to the previous version of the object.

10.7.3.4.2.3 VersionStatus - Allowed Values

The following table shows the allowed values for *Version / Status (StatusEnumeration)*.

Table 19 — VersionStatus - Allowed Values

Value	Description
draft	This version is a draft: it is currently under construction and should not be used for operational purposes.
proposed	This version is comprehensive but not yet validated.
versioned	This is a finalised version frozen from further modifications.
deprecated	This is an old version: it should not be used for operational purposes anymore.
other	Like draft, proposed, and deprecated, this version should not be used for operational purposes (for other reasons, however).

10.7.3.4.3 EntityInVersionReferences (VersionOfObjectRef)- Model Element

A relationship to another an ENTITY IN VERSION is represented by a *VersionOfObjectRef*, that is a reference to a specific version of an entity, e.g. *LineRef*, *ServiceJourneyRef*. Where the same type of entity is referenced by more than one relationship from an ENTITY, the names of the attributes used to represent the relationship need to be qualified; for example, a LINK may have both a *FromPointRef*, and a *ToPointRef* for its start and end POINTS.

Note that the *created* and *changed* attributes of a *VersionOfObjectRef* are for the *reference* instance (that is, the *VersionOfObjectRef* element itself), **not** for the *referenced* object. If a reference is created, or changed, these should be updated.

Table 20 — VersionOfObjectRef - Element

Classification	Name	Type	Cardinality	Description
	NameOfRefClass	NameOfClass	0:1	Reference to object type of referenced ENTITY that is in the specified VERSION FRAME
«atr»	created	xsd:dateTime	0:1	Date and time of creation of reference to ENTITY, i.e. when it was added to frame.
«atr»	changed	xsd:dateTime	0:1	Date and time of last change to reference to

		me		ENTITY.
«FK»	version	VersionRef	0:1	Reference to version of ENTITY that is in the specified VERSION FRAME. Will be checked by XML validity checker.
«FK»	ref	ObjectIdType	1:1	Identifier of a referenced ENTITY.
«FK»	versionRef	VersionRef	0:1	Alternate Reference to version of ENTITY that is in the specified VERSION FRAME. Will not be checked by XML validity checker. +v1.1
	uri	xsd:anyURI	0:1	Location of the external entity. +v2.0

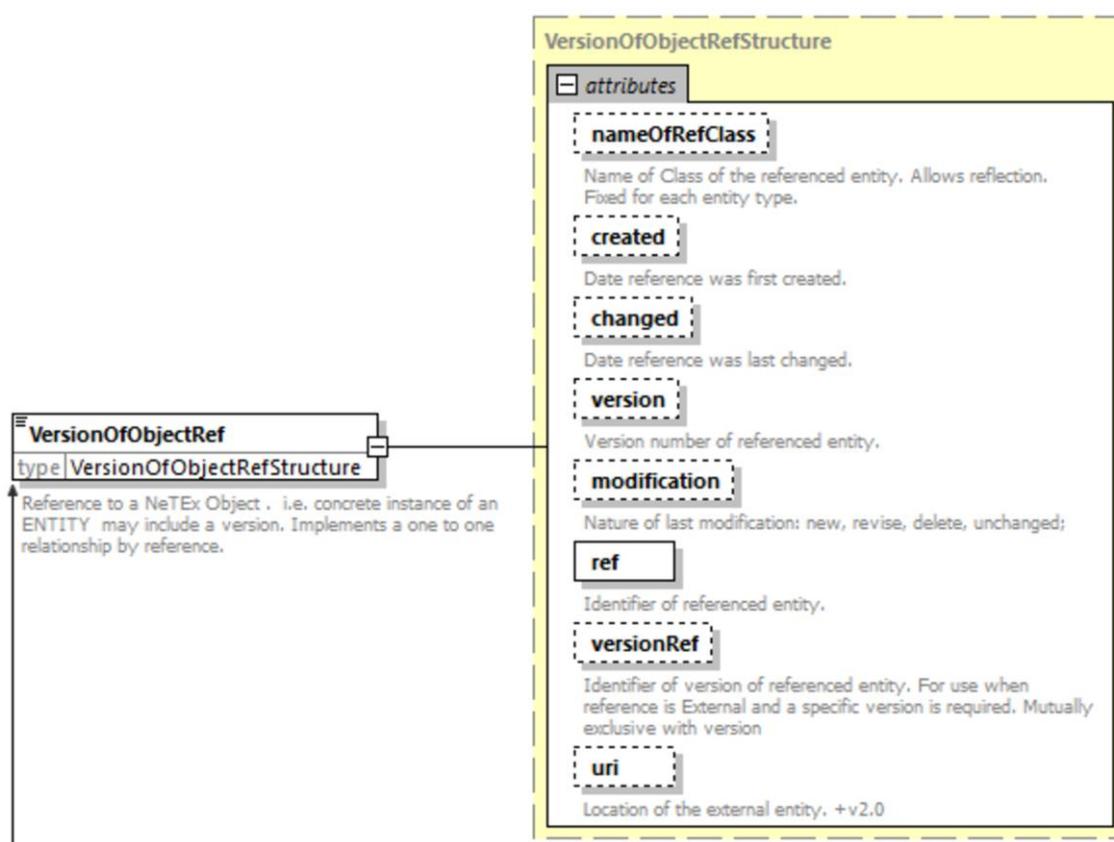


Figure 30 — EntityInVersionInVersionFrame (VersionOfObjectRef) - XSD

10.7.3.4.4 Entity_Entity - Model Element

The Entity_Entity element provides a general purpose implementation of the ENTITY entity that can be used to group the versions (i.e. ENTITY in VERSIONs) of an ENTITY. All the child versions should be of the same class as the ENTITY. See below for an example.

Table 21 — Entity_Entity - Element

Classification	Name	Type	Cardinality	Description
«atr»	nameOfClass	xsd:NMTOKEN	0:1	Name of class of which this is an instance. Used for reflection.

«PK»	id	ObjectIdType	1:1	Identifier of ENTITY. Unique within CODESPACE.
«atr»	created	xsd:dateTime	0:1	Date and time of creation of ENTITY.
«atr»	changed	xsd:dateTime	0:1	Date and time of last change to ENTITY.
«cntd»	versions	(EntityInVersion)	0:*	<p>Versions of the ENTITY.</p> <p>In a given instance of Entity_Entity, the EntityInVersion instances shall all be of the same type as the value of the nameOfClass attribute. For example, if the nameOfClass is 'ServiceLink', then the contained versions shall all be instances of ServiceLink.</p>

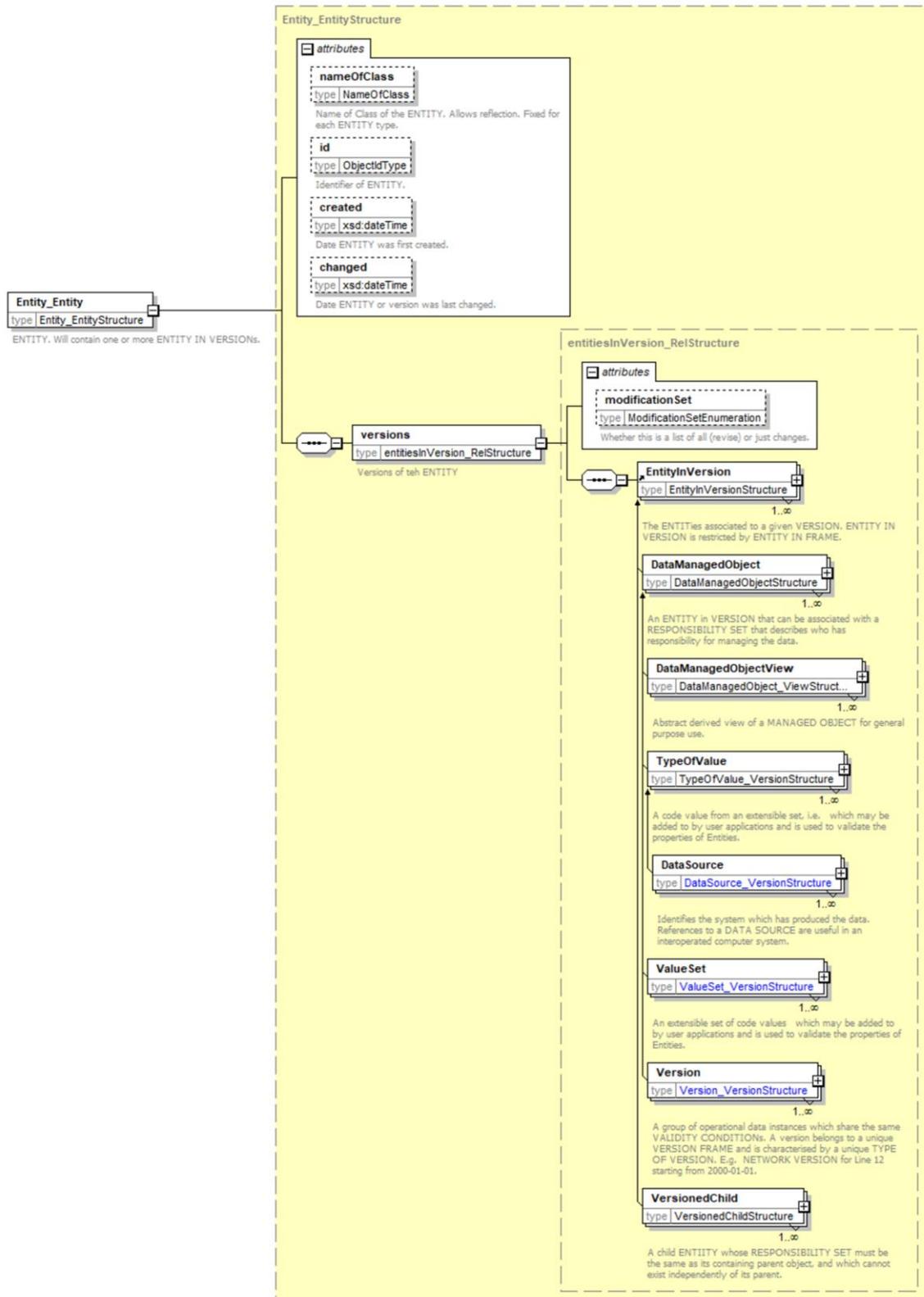


Figure 31 — Entity_Entity - XSD

10.7.3.4.5 TypeOfVersion- Model Element

Classification of a VERSION, according to purpose.

Table 22 — TypeOfVersion - Element

Classification	Name	Type	Cardinality	Description
::>	::>	TypeOfValue	::>	TYPE OF VERSION inherits from TYPE OF VALUE.
«PK»	id	TypeOfVersionIdType	1:1	Identifier of TYPE OF VERSION.

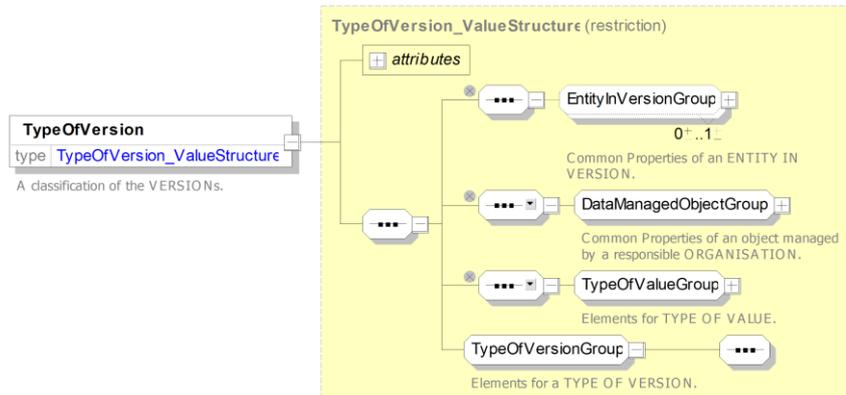


Figure 32 — TypeOfVersion - XSD

10.7.3.5 XML Examples of use of Versions

10.7.3.5.1 Version Instances

The following example XML fragment shows the use of explicit version numbers. The example uses separate version numbers for the SCHEDULED STOP POINT (version="22") and the SERVICE FRAME (version="1").

Example:

```
<ServiceFrame version="1" created="2010-05-17T09:30:47.0Z" id="mybus:ntwkf001">
  <Name>My Network (V1) </Name>
  <scheduledStopPoints>
    <ScheduledStopPoint version="22" created="2010-05-17T09:30:47.0Z"
changed="2011-06-17T09:30:27.0Z" id="mybus:SSP0001A">
      <Name>Haltstelle A</Name>
      <Description>Stop A</Description>
    </ScheduledStopPoint>
  </scheduledStopPoints>
</ServiceFrame>
```

10.7.3.5.2 Including Version Instances

Important: The Version Element is DEPRECATED from NeTEx 2.0. See VERSION for more details.

Information about the VERSIONs can be included along with the actual ENTITY in VERSION instance, for example the following example XML fragment shows the use of VERSION descriptions for the versions of the SCHEDULED STOP POINT and SERVICE FRAME shown in the previous example.

Example:

```
<ServiceFrame version="1" created="2010-05-17T09:30:47.0Z" id="mybus:ntwkf001">
  <Name>My Network (V1) </Name>
  <versions> <!-- DEPRECATED -->
```

```

        <Version created="2010-05-17T09:30:47.0Z" id="mybus:ntwkf001"
version="1">
            <Status>versioned</Status>
            <Description>Version 1 of Service Frame </Description>
            <VersionType>baseline </VersionType>
        </Version>
        <Version created="2011-06-17T09:30:27.0Z " version="022">
            <Status>versioned</Status>
            <Description>Version 22 of Stop Point SSP001A </Description>
        </Version>
    </versions>
    <scheduledStopPoints>
        <ScheduledStopPoint version="22" created="2010-05-17T09:30:47.0Z"
changed="2011-06-17T09:30:27.0Z" id="mybus:SSP0001A">
            <Name>Haltstelle A</Name>
            <Description>Stop A</Description>
        </ScheduledStopPoint>
    </scheduledStopPoints>
</ServiceFrame>

```

10.7.3.5.3 Use of an Entity with Versions of that Entity In Version – XML Example fragment

Important: The Version Element is DEPRECATED from NeTeX 2.0. See VERSION for more details.

Note that only the Version element is deprecated, not the “version” attribute of ENTITIES: thus, two versions of an entity can still be provided in the same exchange by providing multiple records for the same entity, thus having same Id attribute but different version attribute.

It is possible for the same document to hold more than one version of the same ENTITY, that is to say multiple instances of a specific ENTITY IN VERSION. The following example XML fragment shows the use of an Entity_Entity to group two sets of successive versions (i.e. ENTITY IN VERSIONS) of an ENTITY, one for two versions (‘001’ and ‘002’) of a Scheduled Stop Point (‘mybus:SSP0001A’), and the other for two versions of a Service Pattern (‘007’ and ‘012’).

Example:

```

<GeneralFrame version="any" id="mybus:ServiceFrame:SF1">
    <members>
        <Entity_Entity created="2010-05-17T09:30:47.0Z" id="mybus:SSP0001A"
nameOfClass="ScheduledStopPoint">
            <versions>
                <ScheduledStopPoint version="1"
created="2010-05-17T09:30:47.0Z" modification="new" " id="mybus:SSP0001A">
                    <Name>Haltstelle A</Name>
                    <Description>Version one of stop A</Description>
                </ScheduledStopPoint>
                <ScheduledStopPoint version="2" created="2010-05-
17T09:30:47.0Z" changed="2010-05-18T09:30:47.0Z" modification="revise"
derivedFromVersionRef="1" id="mybus:SSP0001A">
                    <Name>Haltstelle A - Museum</Name>
                    <Description>Version two of stop A. Name is Changed and code
added</Description>
                    <PrivateCode>mycodeA</PrivateCode>
                </ScheduledStopPoint>
            </versions>
        </Entity_Entity>

        ::>::::
        <!-- ===ServicePattern_Entity id="mybus:ServicePattern:SP_001" -->
        <Entity_Entity created="2010-05-21T10:30:51.0Z" changed="2010-05-
22T10:30:51.0Z" id="mybus:SP_001" nameOfClass=" ServicePattern ">
            <versions>

```

```

        <ServicePattern version="7" created="2010-05-21T10:30:51.0Z"
modification="new" id="mybus:SP_001">
  <Name>From A to C, version 1</Name>
  <pointsInSequence>
    <StopPointInJourneyPattern version="001" id="mybus:SP_001_01-1">
      <ScheduledStopPointRef version="002" ref="mybus:SSP0001A"/>
    </StopPointInJourneyPattern>
    <StopPointInJourneyPattern version="001" id="mybus:SP_001_02-2">
      <ScheduledStopPointRef version="002" ref="mybus:SSP0002B"/>
    </StopPointInJourneyPattern>
    <StopPointInJourneyPattern version="001" id="mybus:SP_001_03-3">
      <ScheduledStopPointRef version="001" ref="mybus::SSP0003C"/>
    </StopPointInJourneyPattern>
  </pointsInSequence>
</ServicePattern>
  <ServicePattern version="12" created="2010-05-21T10:30:51.0Z"
changed="2010-05-22T10:30:51.0Z" modification="new" id="mybus:SP_001">
  <Name>From A to D, version 2</Name>
  <pointsInSequence>
    <StopPointInJourneyPattern version="2" id="mybus:SP_001_01-1">
      <ScheduledStopPointRef version="1" ref="mybus:SSP0001A"/>
    </StopPointInJourneyPattern>
    <StopPointInJourneyPattern version="2" id="mybus:SP_001_02-2">
      <ScheduledStopPointRef version="2" ref="mybus:SSP0002B"/>
      <ForAlighting>true</ForAlighting>
      <ForBoarding>>false</ForBoarding>
    </StopPointInJourneyPattern>
    <StopPointInJourneyPattern version="2" id="mybus:SP_001_03-3">
      <ScheduledStopPointRef version="1" ref="mybus:SSP0003C"/>
    </StopPointInJourneyPattern>
  </pointsInSequence>
</ServicePattern>
</versions>
</Entity_Entity>
</members>
</GeneralFrame>

```

10.7.4 Implementing relationships in NeTEx

10.7.4.1 Entity identifiers and uniqueness

All NeTEx entities can be given a unique identifier, the `id` attribute. Identifiers are of type *ObjectIdType*.

Specific entities will have a specific subtype, for example *PointIdType* is a subtype of *ObjectIdType*.

Identifier values should be unique within object type within a document. The NeTEx main schema includes integrity constraints to enforce uniqueness.

10.7.4.2 Associations and Entity References

Where an association between two entities in the conceptual model needs to be implemented, normally an element at one end will be chosen in the physical model to hold the reference as a foreign key, i.e. concrete attribute that holds the identifier for the associated object. The choice of end will be according to the semantics of the model. For example, *GroupOfEntities* has an association with *PurposeOfGrouping* to specify a purpose for the group and so has an attribute *PurposeOfGroupingRef* which holds a value of *PurposeOfGroupingIdType*.

Most relationships can be versioned, that is a specific version of the referenced entity may be specified on the reference. To implement versioned references a *VersionOfObjectRef* is used as the attribute type, specialised to the specific type of the referenced entity, for example *PointRef*, *DayTypeRef*, etc.

Thus, instead of a simple reference to simple type (e.g. *PointIdType*) being held as the referencing attribute, a complex type (a *PointRefStructure*) is used that also specifies a version. For example, the following references version '001' of point 'SSP0001A';

```
<FromPointRef ref="mybus:SSP0001A" version="001"/>
```

See below for further XML examples.

Versioned object references are provided as reusable components for all NeTeX DATA MANAGED OBJECTS. For example, in the following figure **DataSourceRef** is a type of **VersionOfObjectRef** that holds both a version id and a reference value of type *DataSourceIdType*.

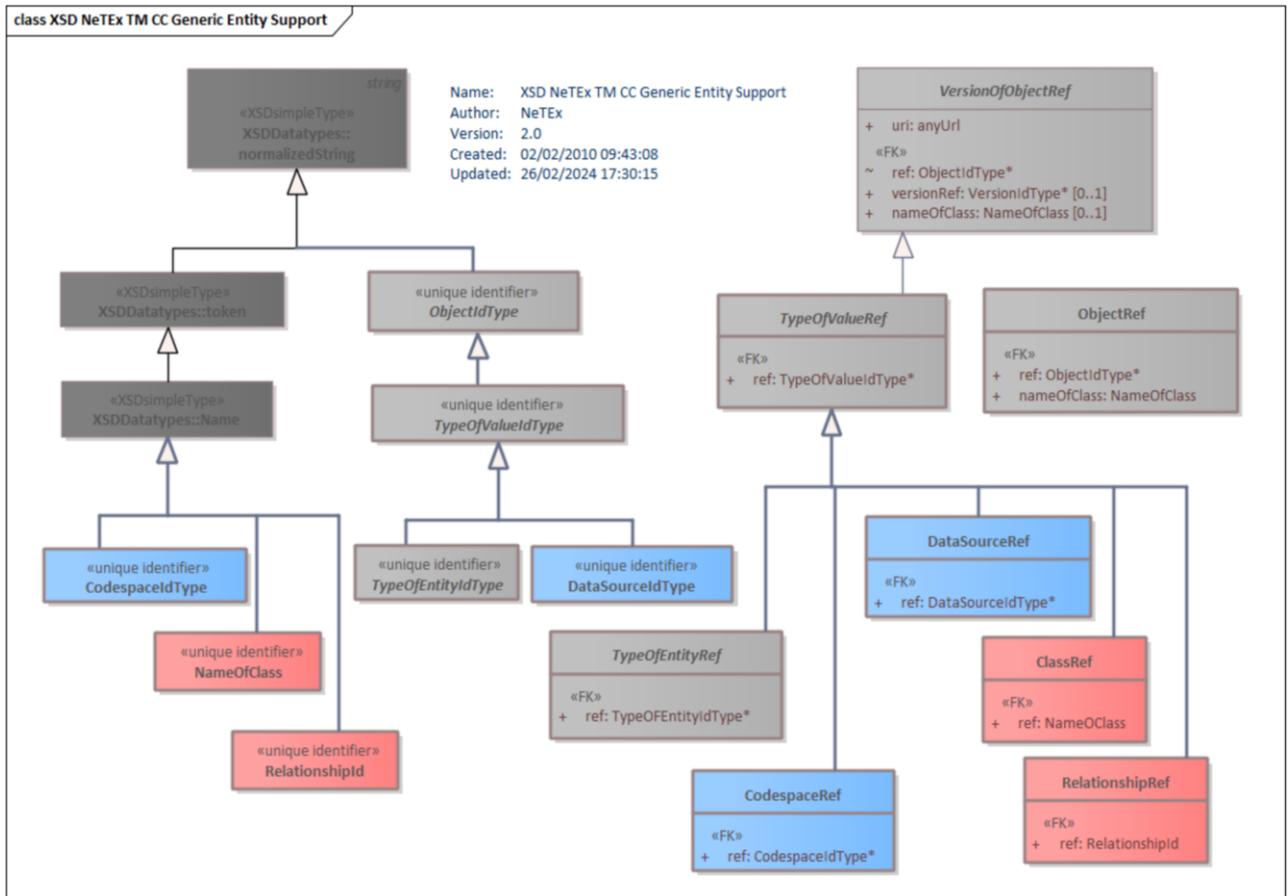


Figure 33 — Entity References - Physical Model (UML)

The version on a reference is optional - if no version is specified integrity checking will be omitted in the XML. A dummy version number can be used to invoke integrity checking against an indeterminate version – this mechanism can be used to ensure that the referenced entity is present in the document regardless of version number. The three levels of enforcement possible are summarised with examples in the following table.

Table 23 — Versioned References

Reference Nature	Example Reference	Example id & version number of referenced entity in same document.	Referential Integrity Checking
Exact version	<PointRef version="22" ref="foo:xxx" />	<Point version="22" id="foo:xxx">	Yes

Indeterminate version	<PointRef version="any" ref="foo:xxx"/>	<Point version="any" id="foo:xxx"	Yes
No version	<PointRef ref="foo:xxx"/>	An instance of Point with identifier foo does not need to be present in the same XML document. If it is present any value is allowed for version.	No

For an n-to-one relationship a simple version reference will be used.

For an n-to-many relationship a set of simple version references will be used, contained within a relationship class. See below.

Generally, the physical attribute used to hold the reference on the referencing entity will by convention be named xxxRef where xxx is the name of the referenced element, or in some cases a qualified name of the referenced element. For example, **PointRef** refers to a POINT and uses an instance of *PointIdType*; **ScheduledStopPoint** refers to a SCHEDULED STOP POINT and uses an instance of *ScheduledStopPointIdType*. Qualifiers are used when it is necessary or helpful to distinguish references. For example FromPointRef and ToPointRef on link; or on **Zone**, **ParentZoneRef** refers to another ZONE The attribute name is qualified with "Parent" to distinguish it from the identifier of the zone itself and has a type of *ZoneRefStructure*.

Certain metadata relationships that are implemented as attributes are not versionable.

Entity →dataSourceRef →**DataSource**

Entity →responsibilitySetRef →**ResponsibilitySet**

Entity →version →Version

Entity →derivedFromVersionRef →**Version**

VersionFrame →compatibleWithBaseVersionRef →**Version**

10.7.4.3 Relationship metaclasses

The NeTeX relationship classes provide a concrete implementation of the metamodel in NeTeX, making it possible for implementations to reflect over the elements in order to provide automated validation and mapping, etc.

There is a specific relationship class for each one-to-many relationship implemented in NeTeX; all of them are concrete specialisations of the NeTeX framework **oneToMany** relationship and by convention are named xxxx_RelStructure. For example, for the relationship 'STOP PLACE contains QUAYS' the **StopPlace** element will have a quays attribute of type *quays_RelStructure*.

Table 24 — NeTeX schema relationship types

Relationship Type	Description	Example
<i>oneToMany_Rel</i>	The child elements are by reference only.	Organisation : 0:* →typeOfOrganisationRefs_Rel: → TypeOfOrganisationRef
<i>containmentAggregation_Rel</i>	The child elements may variously be by reference or inline.	StopPlace : →quays_Rel: → 0:* Quay QuaysRef
<i>strictContainmentAggregation_Rel</i>	The child elements shall be included	CompoundTrain

	in line and there is a compositional semantic.	→ <i>components_Rel</i> : → 1:* TrainInCompoundTrain
<i>frameRelationship_Rel</i>	The child elements shall be present inline but there is no compositional semantic.	SiteFrame : → <i>stopPlacesInFrame_Rel</i> : → 0:* StopPlace

The following figure shows the abstract relationship types used in NeTeX.

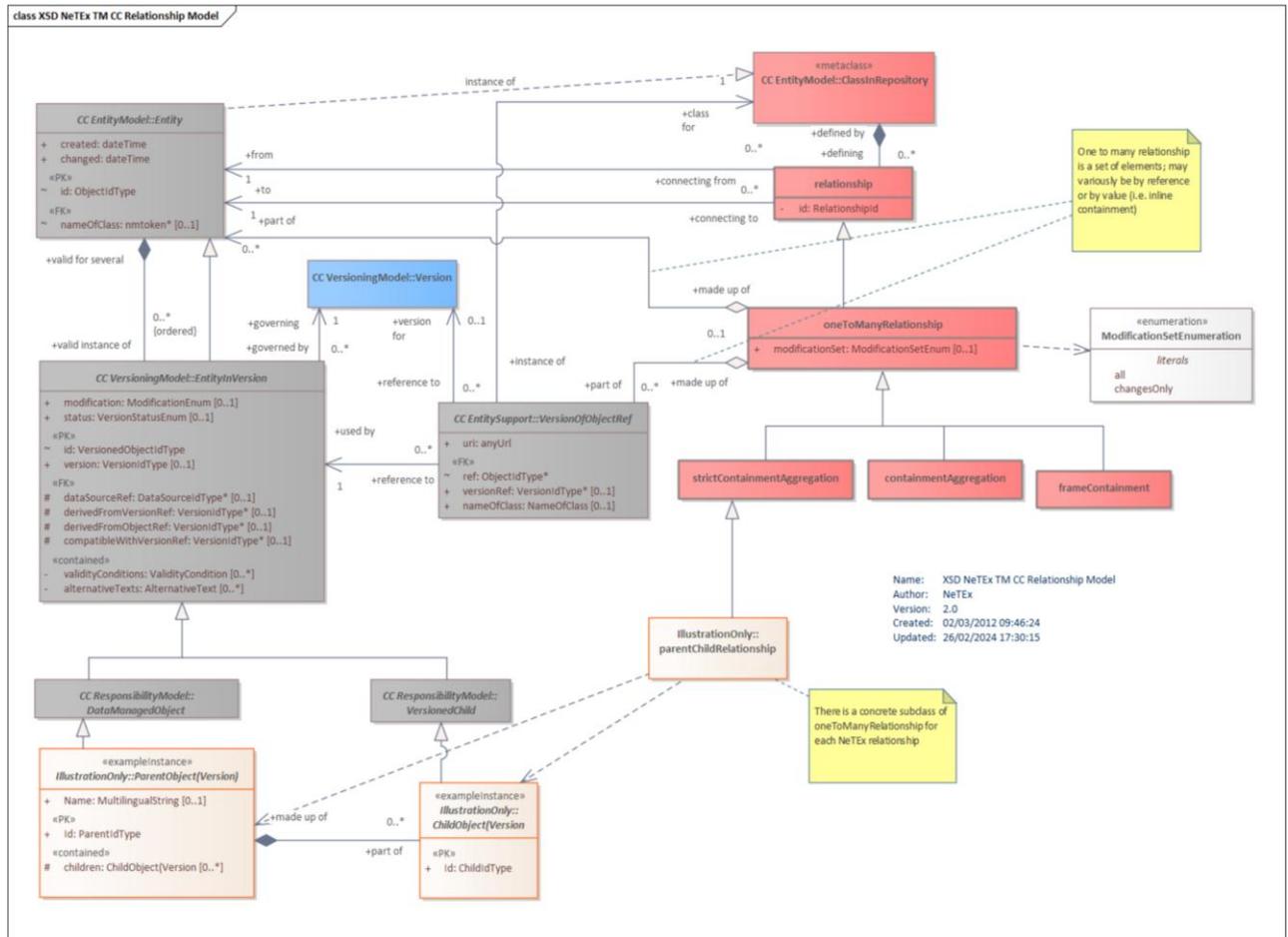


Figure 34 — NeTeX Relationships – Physical Model (UML)

10.7.4.4 XML Examples of Relationships

Depending on the use case, an application may need to exchange a complete dataset as a single coherent document, or it may exchange static and reference data as separate documents as part of a more complex workflow. In the former case, it is desirable to be able to perform automatic integrity checking to ensure that all references are satisfied; in the latter case, only certain references should be checked. The NeTeX schema supports both approaches by means of the version and versionRef attributes. Four different scenarios are found.

- Full integrity checking of a reference using an actual version.
- Integrity checking of a reference using a dummy version.

- No Integrity checking.
- No Integrity checking but version cited.

10.7.4.4.1 Use of versions & Version references with Integrity checking – XML fragment

The following example shows the use of versions and references. The SERVICE LINK is versioned, and connects two SCHEDULED STOP POINTS. It explicitly references which version of each point should be used. The XML will be validated to check that the referenced elements are present in the document.

Example:

```
<scheduledStopPoints>
  <ScheduledStopPoint version="1" created="2010-05-17T09:30:47.0Z"
changed="2010-05-18T09:30:47.0Z" modification="revise" id="mybus:SSP0001A">
  <Name>Haltstelle A - Museum</Name>
  <Description>Version one of stop A. Name is Changed</Description>
  <privateCodes>
    <PrivateCode>mycodeA</PrivateCode>
  </privateCodes>
</ScheduledStopPoint>
  <ScheduledStopPoint version="2" created="2010-05-18T09:30:47.0Z"
changed="2010-06-18T09:30:47.0Z" modification="revise" id="mybus:SSP0002B">
  <Name>Haltstelle B</Name>
  <Description>Version two of stop B</Description>
  <privateCodes>
<PrivateCode>mycodeB</PrivateCode>
  </privateCodes>
</ScheduledStopPoint>
</scheduledStopPoints>
<serviceLinks>
  <ServiceLink version="2" created="2010-05-19T10:30:47.0Z" changed="2010-05-
21T10:30:51.0Z" modification="revise" id="mybus:SL_AtoB01">
  <Name>Version one of Link from A to B</Name>
  <Distance>1.12</Distance>
  <FromPointRef version="1" ref="mybus:SSP0001A"/>
  <ToPointRef version="2" ref="mybus:SSP0002B"/>
</ServiceLink>
</serviceLinks>
```

10.7.4.4.2 Use of dummy versions & references with Integrity checking – XML fragment

The following example shows the use of dummy versions with references. The SERVICE LINK connects two SCHEDULED STOP POINTS. It explicitly references which version of each point should be used. However a dummy value (by convention 'any') is used as the version number. The XML will be validated to check that the referenced elements are present in the document with the dummy version number - but no specific version is actually intended.

Example:

```
<scheduledStopPoints>
  <ScheduledStopPoint version="any" created="2010-05-17T09:30:47.0Z"
changed="2010-05-18T09:30:47.0Z" modification="revise" id="mybus:SSP0001A">
  <Name>Haltstelle A - Museum</Name>
  <Description>Version two of stop A. Name is Changed</Description>
  <privateCodes>
<PrivateCode>mycodeA</PrivateCode>
  </privateCodes>
</ScheduledStopPoint>
  <ScheduledStopPoint version="any" created="2010-05-18T09:30:47.0Z"
changed="2010-06-18T09:30:47.0Z" modification="revise" id="mybus:SSP0002B">
```

```

        <Name>Haltstelle B</Name>
        <Description>Version two of stop B</Description>
        <privateCodes>
<PrivateCode>mycodeB</PrivateCode>
</privateCodes>
    </ScheduledStopPoint>
</scheduledStopPoints>
<serviceLinks>
    <ServiceLink version="002" created="2010-05-19T10:30:47.0Z" changed="2010-
05-21T10:30:51.0Z" modification="revise" id="mybus:SL_AtoB01">
        <Name>Version one of Link from A to B</Name>
        <Distance>1.12</Distance>
        <FromPointRef version="any" ref="mybus:SSP0001A"/>
        <ToPointRef version="any" ref="mybus:SSP0002B"/>
    </ServiceLink>
</serviceLinks>

```

10.7.4.4.3 Use of unversioned references - XML fragment

The following example shows the use of references without a version. The SERVICE LINK is versioned and connects two SCHEDULED STOP POINTS. However, it does not specify which version of each point should be used. The XML will not be validated to check that the referenced elements are present in the document.

Example:

```

<scheduledStopPoints>
    <ScheduledStopPoint version="001" created="2010-05-17T09:30:47.0Z"
changed="2010-05-18T09:30:47.0Z" modification="revise" id="mybus:SSP0001A">
        <Name>Haltstelle A - Museum</Name>
        <Description>Version one of stop A. Name is Changed</Description>
        <privateCodes>
            <PrivateCode>mycodeA</PrivateCode>
        </privateCodes>
    </ScheduledStopPoint>
    <ScheduledStopPoint version="002" created="2010-05-18T09:30:47.0Z"
changed="2010-06-18T09:30:47.0Z" modification="revise" id="mybus:SSP0002B">
        <Name>Haltstelle B</Name>
        <Description>Version two of stop B</Description>
        <privateCodes>
            <PrivateCode>mycodeB</PrivateCode>
        </privateCodes>
    </ScheduledStopPoint>
</scheduledStopPoints>
<serviceLinks>
    <ServiceLink version="002" created="2010-05-19T10:30:47.0Z" changed="2010-
05-21T10:30:51.0Z" modification="revise" id="mybus:SL_AtoB01">
        <Name>Version one of Link from A to B</Name>
        <Distance>1.12</Distance>
        <FromPointRef ref="mybus:SSP0001A"/>
        <ToPointRef ref="mybus:SSP0002B"/>
    </ServiceLink>

```

10.7.4.4.4 Use of version references without integrity checking - XML fragment

The following example shows the use of references with a version, however the version is merely cited (the versionRef keyword rather than version is used) and will not be checked by the XML validator (but might still be used by application program code). The SERVICE connects two SCHEDULED STOP

POINTS. The XML will not be validated to check that the referenced elements are present in the document.

Example:

```
<scheduledStopPoints>
  <ScheduledStopPoint version="001" created="2010-05-17T09:30:47.0Z"
changed="2010-05-18T09:30:47.0Z" modification="revise" id="mybus:SSP0001A">
  <Name>Haltstelle A - Museum</Name>
  <Description>Version two of stop A. Name is Changed</Description>
</privateCodes>
  <PrivateCode>mycodeA</PrivateCode>
</privateCodes>
  <ScheduledStopPoint version="002" created="2010-05-18T09:30:47.0Z"
changed="2010-06-18T09:30:47.0Z" modification="revise" id="mybus:SSP0002B">
  <Name>Haltstelle B</Name>
  <Description>Version two of stop B</Description>
</privateCodes>
  <PrivateCode>mycodeB</PrivateCode>
</privateCodes>
</ScheduledStopPoint>
</scheduledStopPoints>
<serviceLinks>
  <ServiceLink version="002" created="2010-05-19T10:30:47.0Z" changed="2010-
05-21T10:30:51.0Z" modification="revise" id="mybus:SL_AtoB01">
  <Name>Version one of Link from A to B</Name>
  <Distance>1.12</Distance>
  <FromPointRef versionRef="001" ref="mybus:SSP0001A"/>
  <ToPointRef versionRef="002" ref="mybus:SSP0002B"/>
</ServiceLink>
</serviceLinks>
```

10.7.5 Generic Version Frame

10.7.5.1 Generic VERSION FRAME - Conceptual MODEL

NOTE The following explanations use excerpts from Transmodel.

Usually, when exchanging data, only a subset of all the possibly available OpRa classes are present in an exchanged dataset and only a subset of the available data instances for those classes (for example, one specific version of the class, and possibly only a subset of its data attributes). This subset is defined by the VERSION FRAME mechanism. The TYPE OF FRAME can indicate the expected contents. For example, there might be a TIMETABLE FRAME containing all vehicle schedules of 'Line 12' services, grouping together the LINE, ROUTES, JOURNEY PATTERNS, SCHEDULED STOP POINTS, VEHICLE JOURNEYS, etc. for a specific validity period. There might be an TYPE OF FRAME TYPE agreed between the exchanging parties of 'Bavarian Line Timetable' for the TYPE OF FRAME that indicates the contents.

VERSION FRAMEs allow data to be managed and exchanged as a coherent version, that is a set of instances of different entity types that are consistent and correct as to referential integrity and other business semantics and so are suitable for use without extensive consistency checking by the importing application. A VERSION FRAME contains a list of specific versions of an entity, that is, instances of ENTITY IN VERSION.

To be useful a dataset must be both *correct* (satisfy integrity checks that all required references are satisfied and all types correct) and *complete* (all necessary elements shall be present). Note that this does not imply that a single VERSION FRAME shall contain all the elements that are referenced in that VERSION FRAME. A VERSION FRAME can refer to external elements defined in other VERSION FRAMEs. If *versioned* references are used then these other VERSION FRAMEs shall be present in the same XML document. However, if *unversioned* references are used then the other VERSION FRAMEs do not have to

be present at all and it is assumed they are exchanged separately. In practice different use cases require different sets of data to be assembled and have different criteria for completeness.

A VERSION FRAME may be a prerequisite for another VERSION FRAME.

10.7.5.1.1 Types of Frame – Conceptual MODEL

The possibilities for including specific types of ENTITIES in VERSION in a FRAME are limited by the generic rules set by a corresponding CLASS IN FRAME. All the classes that are allowed to be present in the frame are defined by the CLASS IN FRAME, and each frame is defined by its TYPE OF FRAME. Note that validation to check that only permitted classes are present has to be carried out by the importing application and cannot be done by XML validation alone.

VERSION FRAMEs may have common properties as regards to validity. This is described by the TYPE OF FRAME entity (e.g. vehicle schedules, network description for line versions, etc.). The main property of a TYPE OF FRAME is the purpose it is designed for.

A TYPE OF FRAME may be associated with a particular TYPE OF VALIDITY, which expresses a general validity environment. The TYPE OF VALIDITY will apply to any VERSION FRAMEs of that type. For instance, if the schedules designed for day types are to be distinguished from schedules planned for a particular operating day, different TYPES OF VALIDITY, which will serve as a basis to select general validity rules, may specify this difference. Similarly, certain VERSION FRAMEs may be designed only for simulation purposes and be distinguished from production data, this classification being expressed with a different TYPE OF VALIDITY.

A particular rule (filter) for selecting VERSIONs may be attached to a TYPE OF FRAME, in particular in situations where several VERSIONs of the same VERSION FRAME may be in competition at the same time. The valid VERSION in a given context will be chosen by application of a specific request filters, comparing this context with parameter values. For instance, production vehicle schedules will be selected using the filters which state the PROPERTIES OF DAY to be considered by an algorithm, whereas simulation schedules may be selected using another filter. This aspect is useful when rules for selecting versions are more complex than the application of a simple criterion such as “the most recent valid version”.

A TYPE OF FRAME may include other TYPES OF FRAME, for which the validity rules and processes may be different. This is represented by a circular relationship on TYPE OF FRAME.

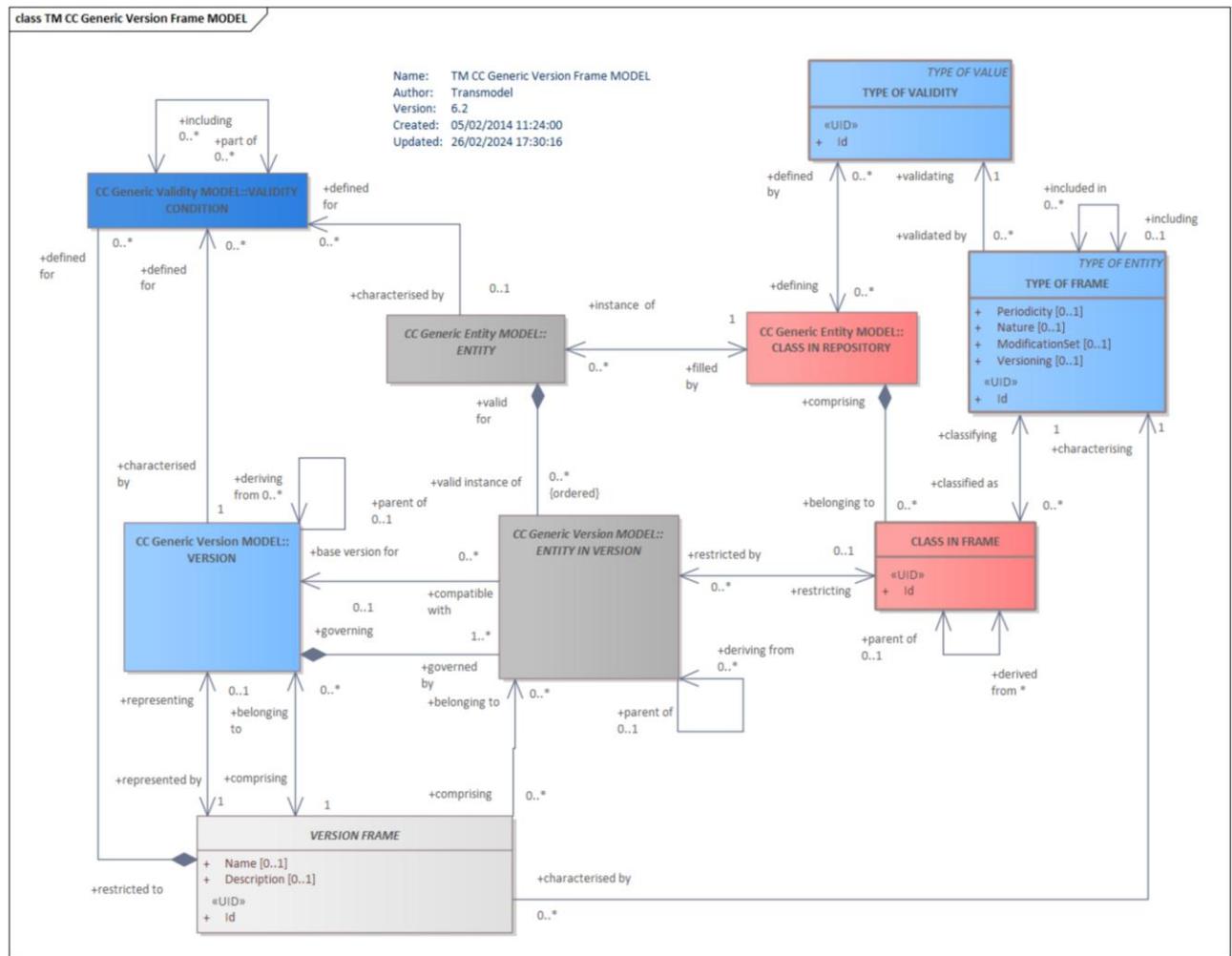


Figure 35 — Generic Version Frame – Conceptual MODEL (UML)

A particular VERSION FRAME, defined according to a TYPE OF FRAME, is usually limited by operational parameters. For example, VERSION FRAME for network description of “*area West*”, for vehicle schedule parameters in “*depot D*”, for fare versions on “*tramway lines*”, etc. When these limiting parameters are actual instances of entities stored in the database, this may be described by instances of entity VALIDITY CONDITION, related to the VERSION FRAME. For instance, a VALIDITY CONDITION may represent a DEPOT, a LINE or a GROUP OF LINES, a particular OPERATOR, etc.

10.7.5.1.2 Version Frames and Versioning

The VERSION FRAME itself is versioned, so that if any change is made to the content of a frame to add, change or delete its entities, then a new version of the frame shall be created; see 'Version States' above. Only versioned Frames (i.e. with status of ‘*version*’) may be exchanged. Typically, in a given repository a new version will be created with a status of ‘*draft*’, be worked on locally till complete and consistent, and then be ‘*frozen*’ and versioned for export.

For a defined group of object instances, there may be several (consecutive or competitive) VERSIONS of a VERSION FRAME. For example, the content of a frame containing the SCHEDULED STOP POINTs for a town will change as they are added, updated or deleted, so there will be successive versions of the same frame), or there may be successive instances of a given TIMETABLE FRAME reflecting successive changes to a given schedule. In Summary:

A given aggregation may undergo successive versions as the data evolves through its lifecycle, so there may be several *consecutive* VERSIONS of a VERSION FRAME.

A given aggregation may represent an alternative to be used in particular conditions, so there may be several *competitive* VERSIONS of a VERSION FRAME in which case a VALIDITY condition shall be attached to the frame to discriminate the conditions for use.

10.7.5.1.3 General Frames and Explicit Frames

In the original Transmodel model the required content of a given frame can be specified dynamically (i.e. at run time) using the CLASS IN FRAME mechanism which in effect provides metadata that an importing program can use to check that all of the necessary elements are present. Such a dynamic validation mechanism cannot be enforced in XML and so to obtain the benefits of XML's built in type checking in OpRa, the general frame mechanism is complemented by a more specific set of "Explicit" VERSION FRAMES that specify sets of data elements appropriate for a particular use case or set of related use cases; for example, INFRASTRUCTURE FRAME, SITE FRAME, TIMETABLE FRAME, etc. Each of these represents a predefined combination of data types that are commonly exchanged together as part of the data management processes of organisations concerned with transport data.

Sometimes data elements from more than one type of explicit frame is needed; a COMPOSITE FRAME can be used to group a coherent set of explicit frames. A given frame may indicate which other frames are prerequisites.

The explicit frames correspond to various parts of OpRa and in most cases are described in the appropriate section along with their contents. In most cases a given OpRa element appears only in one explicit frame. For example, INFRASTRUCTURE LINK is found only in the INFRASTRUCTURE FRAME. The following table summarises the frames.

Table 25 — Explicit Frames

	Name	Primary contents
Framework	CompositeFrame	Frame to group other VERSION FRAMES.
	GeneralFrame	Any ENTITY or ENTITY IN VERSION.
	ResourceFrame	General purpose components such as ORGANISATIONs, VEHICLE TYPEs and code values.
Part1	InfrastructureFrame	INFRASTRUCTURE POINTs, LINKs & RESTRICTIONs.
	SiteFrame	SITEs, STOP PLACEs, POINTS OF INTEREST and other fixed objects.
	ServiceFrame	Network description elements such as LINEs, ROUTEs, etc. Tactical Planning elements such as SCHEDULED STOP POINTs, JOURNEY PATTERNs, etc. pattern.
Part2	TimetableFrame	Timetable elements: SERVICE JOURNEYs with timings.
	VehicleScheduleFrame	VEHICLE SCHEDULEs: BLOCKs and BLOCK PARTs.
	DriverScheduleFrame	DRIVER SCHEDULEs: DUTYs and DUTY plans. (not yet available)
Part3	FareFrame	Fare related elements: TARIFF STRUCTUREs, FARE PRODUCTs, FARE PRICEs, SALES OFFER PACKAGEs, etc.
	SalesTransactionFrame	Data generated by the sale of fare products: CUSTOMERs, CUSTOMER ACCOUNTs, FARE CONTRACTs, SALES TRANSACTIONs, TRAVEL DOCUMENTs, etc.

VersionFrame itself is abstract and cannot be used in an XML document, but a concrete general purpose frame **GeneralFrame** is provided that can contain any type of ENTITY or ENTITY in VERSION. This can be used to handle any ad-hoc assembly of data.

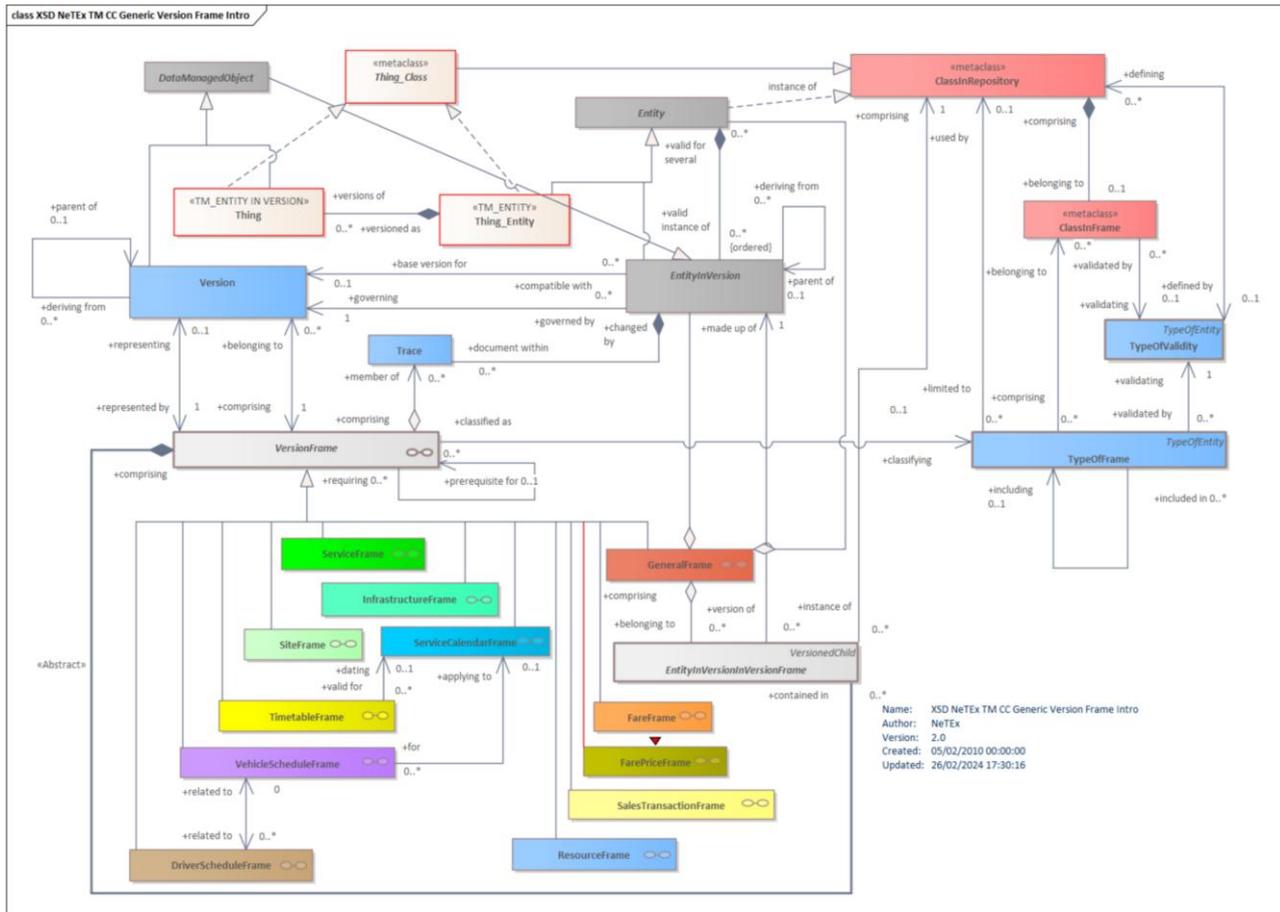


Figure 36 — Explicit Version Frames - Overview (UML)

10.7.5.1.4 Using Version Frames to Requests

In OpRa a request and filtering mechanism (at XSD level) is used to specify the data to be returned. The contents are then returned in a VERSION FRAME. The type of frame depends on the request. The data to be requested can be specified in several different ways as shown in the following table

The filtering mechanism takes into account the VALIDITY CONDITIONS attached to the data. This means that, for example, the same NeTeX request may provide different results, depending on when the request is submitted and what filter criteria are specified.

Table 26 — Requests & Version Frames

Request Type	Description	Results Returned
Version Frame + Version	Specific version of a specific known version frame. E.g. Version 002 of TimetableFrame ABC001.	Exact requested version of Version frame returned.
Version Frame + Validity Criteria	Specific known version frame that meets some validity criteria. E.g. TimetableFrame ABC001 for	Instance of version frame that meets validity condition.

	Summer 2013.	
Scope Data criteria _ Validity Criteria	Criteria for selecting data. E.g. "Stops for the Munich Area, now"".	Ad hoc version frames returned to hold matching data.

10.7.5.1.5 Types of Version frame

VERSION FRAMEs may have common properties as regards validity. This is described by the TYPE OF FRAME entity (e.g. vehicle schedules, network description for line versions, etc.). The main property of a TYPE OF FRAME is the purpose for which it is designed.

A TYPE OF FRAME may be associated with a particular TYPE OF VALIDITY, which expresses a general validity environment. The TYPE OF VALIDITY will apply to any VERSION FRAMEs of that type. For instance, if the schedules designed for day types are to be distinguished from schedules planned for a particular operating day, different TYPEs OF VALIDITY, which will serve as a basis to select general validity rules, may specify this difference. Similarly, certain VERSION FRAMEs may be designed only for simulation purposes and be distinguished from production data, this classification being expressed with a different TYPE OF VALIDITY.

A particular rule (filter) for selecting VERSIONs may be attached to a TYPE OF FRAME, in particular in situations where several VERSIONs of the same VERSION FRAME may be in competition at the same time. The valid VERSION in a given context will be chosen by application of a specific request filters, comparing this context with parameter values. For instance, production vehicle schedules will be selected using the filters which state the PROPERTies OF DAY to be considered by an algorithm, whereas simulation schedules may be selected using another filter. This aspect is useful when rules for selecting versions are more complex than the application of a simple criterion such as "the most recent valid version".

A TYPE OF FRAME may include other TYPEs OF FRAME, for which the validity rules and processes may be different. This is represented by a circular relationship on TYPE OF FRAME.

A particular VERSION FRAME, defined according to a TYPE OF FRAME, is usually limited by operational parameters. For example, VERSION FRAME instances; for network description of "area West"; for vehicle schedule parameters in "depot D"; for fare versions on "tramway lines", etc. When these limiting parameters are actual instances of entities stored in the database, this may be described by the entity VALIDITY CONDITION, related to VERSION FRAME. For instance, a VALIDITY CONDITION may represent a DEPOT, a LINE or a GROUP OF LINEs, a particular OPERATOR, etc.

10.7.5.2 Generic Version Frame – Physical Model

10.7.5.2.1 Version Frame – Physical Model

The following diagram shows the Physical model for the OpRa versioning entities. A VERSION FRAME contains instances of ENTITY IN VERSION IN VERSION FRAME (i.e. the normal NeTEx elements).

In OpRa, VERSION FRAME is specialised to create actual frame instances, for example SITE FRAME, RESOURCE FRAME, etc., each containing a restricted set of ENTITIES for a particular business purpose.

Normally there will only be one VERSION of each of the top level ENTITIES in a given frame, reflecting a single coherent set of data elements. However it is possible to include multiple versions of the same ENTITY if appropriate.

A VERSION FRAME may be given certain VERSION FRAME DEFAULTs which describe properties that apply to all elements in the VERSION FRAME unless explicitly overridden, for example LOCALE and DATA SOURCE.

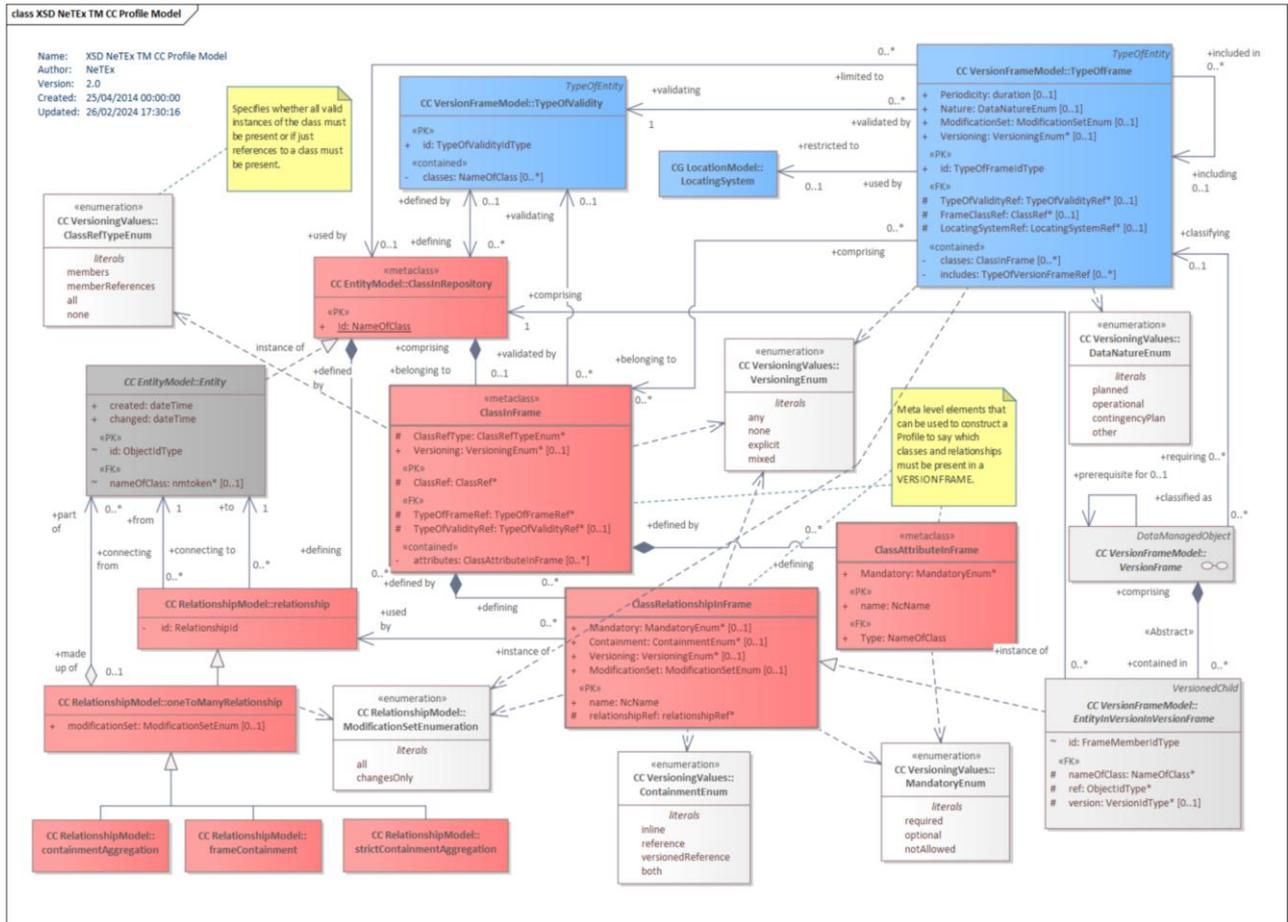


Figure 39 — Profile Model – Physical Model (UML)

10.7.5.3 Version Frame – Attributes and XSD

10.7.5.3.1 VersionFrame – Model Element

10.7.5.3.1.1 General

A VersionFrame is a set of VERSIONS provided by the same DATA SYSTEM and characterised by a given TYPE OF FRAME. The validity of a VERSION FRAME may be restricted by VALIDITY CONDITIONS.

The frame may include default values that apply to all elements so that they do not specify a specific value of their own – see FrameDefaults.

Table 27 — VersionFrame – Element

Classification	Name	Type	Cardinality	Description
::>	::>	DataManagedObject	::>	VERSION FRAME inherits from DATA MANAGED OBJECT.
«PK»	id	VersionFrameIdType	1:1	Identifier of VERSION FRAME.
	Name	MultilingualString	0:1	Name of VERSION FRAME.
	Description	MultilingualString	0:1	Description of VERSION FRAME.
«FK»	TypeOfFrameRef	TypeOfFrameRef	0:1	Reference to TYPE of VERSION FRAME.

«FK»	BaselineVersionFrameRef	VersionRef	0:1	DEPRECATED: Previous baseline version frame that objects in this frame require. -v2.0
«cntd»	codespaces	Codespace	0:*	CODESPACES used in this frame. Normally there will be at least one. A default may be specified in the Frame defaults.
«cntd»	FrameDefaults	FrameDefaults	0:1	Set of default values to assume for values in frame if not explicitly stated on individual elements.
«cntd»	versions	Version	0:*	DEPRECATED: Formal definitions of VERSIONS included in frame. -v2.0
«cntd»	prerequisites	VersionFrameRef	0:*	Prerequisite VERSION FRAMEs containing elements that this frame depends on. +v1.1
«cntd»	traces	Trace	0:*	TRACEs recording changes to ENTITIES in FRAME.
«cntd»	contentValidityConditions	ValidityCondition	0:*	Shared VALIDITY CONDITIONs used by elements in the FRAME. Note that ValidityCondition can be combined and ANDed (all the conditions shall be fulfilled at the same time)
«cntd»	layers	LayerRef	0:*	Layers included in VERSION FRAMEs. +v1.1

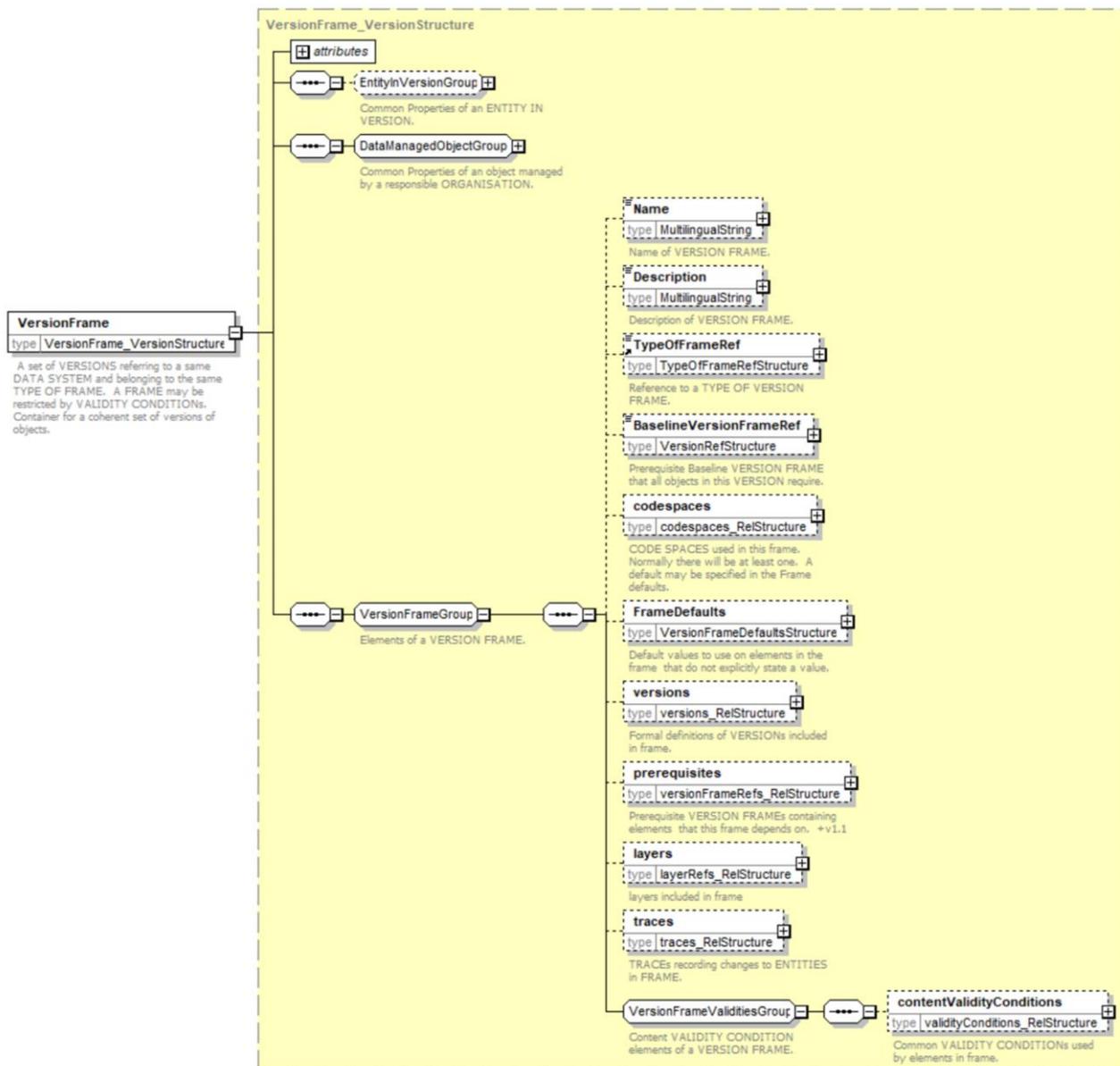


Figure 40 — VersionFrame - XSD

10.7.5.3.1.2 FrameDefaults - Wrapper Element

The FrameDefaults element specifies default values for certain common properties of elements in the frame, such as DATA SOURCE, time-zone etc., to be applied to elements in the frame for which an explicit value has not been specified. The use of defaults can both simplify export and reduce the size of documents.

Table 28 — FrameDefaults - Element

Classification	Name	Type	Cardinality	Description
«FK»	<i>DefaultCodeSpaceRef</i>	<i>CodeSpaceRef</i>	0:1	Default CODESPACE to assume for identifiers that do not have an explicit CODESPACE specified.

«FK»	DefaultDataSourceRef	<i>DataSourceRef</i>	0:1	DATA SOURCE to use for elements in the frame which do not have a DATA SOURCE specified.
«FK»	Default-ResponsibilitySetRef	<i>ResponsibilitySetRef</i>	0:1	RESPONSIBILITY SET to use for elements in the frame which do not have a RESPONSIBILITY SET specified.
«cntd»	DefaultLocale	<i>Locale</i>	0:1	Default LOCALE to use to provide attribute values for elements in the frame which do not have a LOCALE element specified, for example language time zone, etc.
	DefaultLocationSystem	<i>xsd:normalizedString</i>	0:1	Default LOCATION SYSTEM to use for locations in the frame which do not have a LOCATION SYSTEM specified.
«enum»	DefaultSystemOfUnits	<i>SystemOfUnitsEnum</i>	0:1	Default SystemOfUnits to use for measurable attributes in the frame for which no units have been specified. See allowed values.
«enum»	DefaultCurrency	<i>CurrencyType</i>	0:1	Default CURRENCY to use for amount attributes in the frame for which a currency has not been specified.

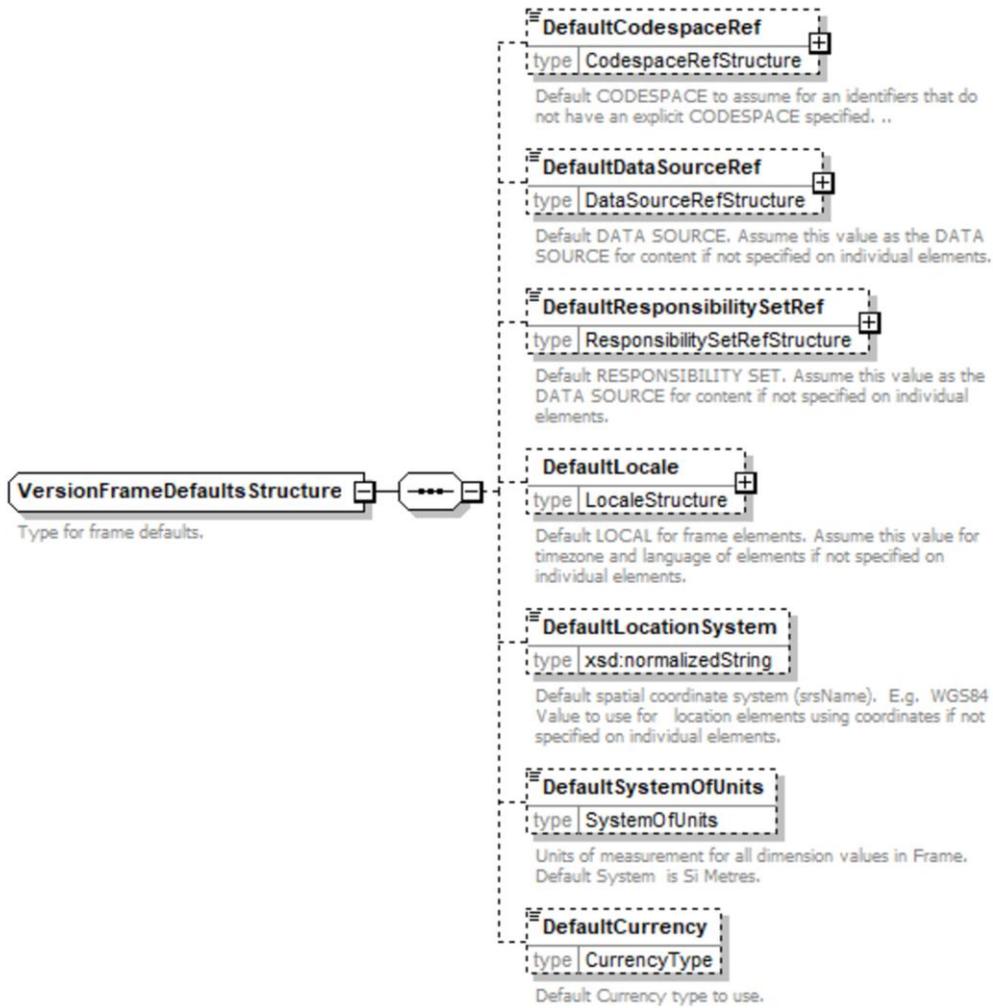


Figure 41 — VersionFrameDefaults - XSD

10.7.5.3.2 EntityInVersionInVersionFrame - Model Element

Many of the data elements exchange by OpRa are instances of ENTITY IN VERSION IN FRAME; they have the attributes of an ENTITY IN FRAME, and exist in the context of a specific VERSION FRAME.

10.7.5.3.3 TypeOfFrame - Model Element

10.7.5.3.3.1 General

A TYPE OF FRAME is a classification of VERSION FRAMES according to a common purpose; e.g. line descriptions for line versions, elements required for a timetable for vehicle schedules, historic data to determine operating costs, etc. A TYPE OF FRAME is ruled by a unique TYPE OF VALIDITY.

A TYPE OF FRAME may be used to classify a profile for using OpRa for a specific purpose; for example, see profile model below.

Table 29 — TypeOfFrame - Element

Classification	Name	Type	Cardinality	Description
::>	::>	TypeOfValue	::>	TYPE OF VALIDITY inherits from TYPE OF VALUE.

«PK»	id	<i>TypeOfFrameIdType</i>	1:1	Identifier of TYPE OF FRAME.
«FK»	TypeOfValidityRef	<i>TypeOfValidityRef</i>	0:1	Reference to a TYPE OF VALIDITY.
«FK»	FrameClassRef	<i>ClassRef</i>	0:1	Reference to an XML class of a type of VERSION FRAME (ResourceFrame , SiteFrame , TimetableFrame , etc.), that is the XML class of frame that VERSION FRAMEs having this specific TYPE OF FRAME should use to hold elements.
«cntd»	classes	<i>ClassInFrameRef</i>	0:*	Classes to be present in frames conforming to the TYPE OF FRAME.
«cntd»	typesOfEntity	<i>TypeOfEntityRef</i>	0:*	TYPEs OF ENTITY to include in FRAME. + v1.1
«cntd»	includes	<i>TypeOfFrame</i>	0:*	TYPEs OF FRAME contained in TYPE OF FRAME. Shall not be cyclic.
«FK»	LocatingSystemRef	<i>SrsNameType</i>	0:1	Required spatial coordinate system (srsName). E.g. WGS84 Value to use for location elements using coordinates if not specified on individual elements.
«enum»	ModificationSet	<i>ModificationSetEnum</i>	0:1	Nature of contents in set: See allowed values below.

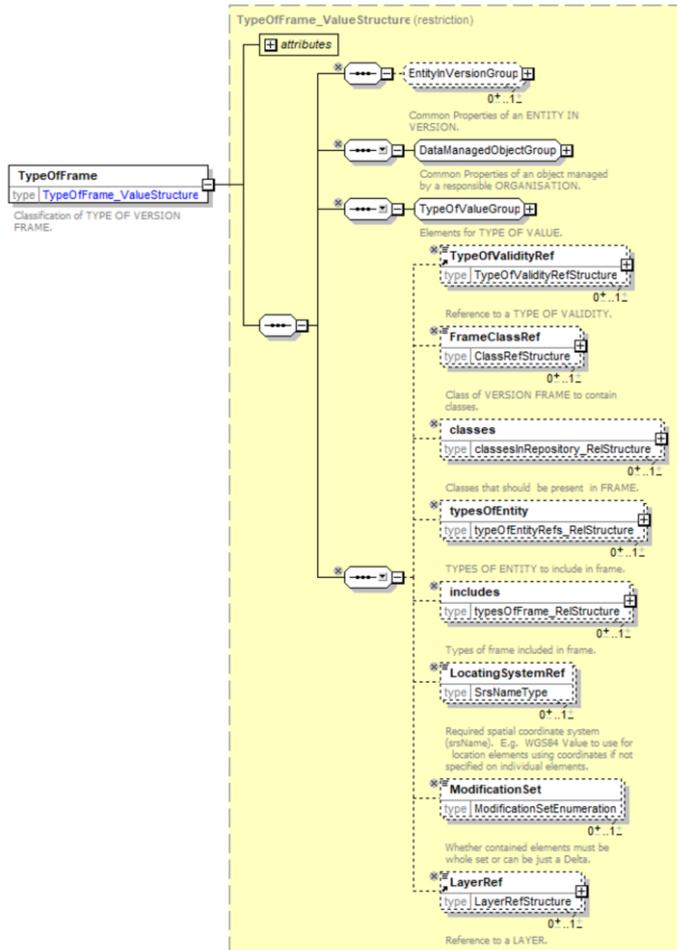


Figure 42 — TypeOfFrame - XSD

10.7.5.3.3.2 ModificationSet - Allowed Values

Allowed values for **ModificationSet** (*ModificationSetEnumeration*).

Table 30 — *ModificationSet* - Allowed values

Value	Description
<i>all</i>	Include all properties (attributes and subcomponents).
<i>changesOnly</i>	Include only those properties which have changed. Required attributes will always be included.

10.7.5.3.4 TypeOfValidity - Model Element

10.7.5.3.4.1 General

A classification of the validity of a TYPE OF FRAME, e.g. VERSION FRAMEs for schedules designed for DAY TYPEs, SITEs, dated schedules, etc.

Table 31 — *TypeOfValidity* - Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>TypeOfValue</i>	::>	TYPE OF VALIDITY inherits from TYPE

				OF VALUE.
«PK»	id	<i>TypeOfValidityIdType</i>	1:1	Identifier of Validity type. Unique within repository.
	Periodicity	<i>xsd:duration</i>	0:1	Periodicity of data in frames of this type.
«enum»	Nature	<i>FrameNatureEnum</i>	0:1	Nature of validity data. See allowed values below.
«cntd»	classes	<i>ClassRef</i>	0:*	Classes that should be present in FRAME.

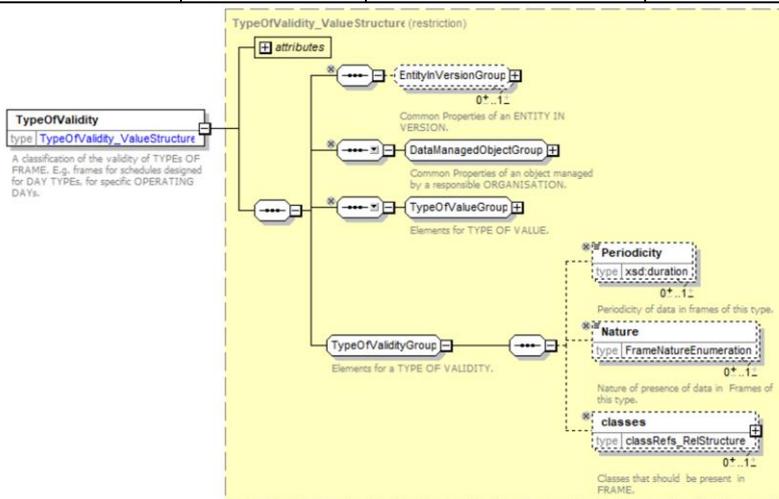


Figure 43 — TypeOfValidity - XSD

10.7.5.3.5 DataNature – Allowed Values

Allowed values for the nature of a TYPE OF VALIDITY (*FrameNatureEnumeration*).

Table 32 — TypeOfValidity – Allowed Values

Value	Description
<i>planned</i>	Data is for planning.
<i>operational</i>	Operational data.
<i>contingencyPlan</i>	Data is for contingency plan to be used in certain circumstances.
<i>other</i>	Other.

10.7.5.3.6 ClassInFrame– Model Element

10.7.5.3.6.1 General

Requirements for use of an XML Class representing a given entity in a VERSION FRAME. This is a meta class that allows services to specify whether an XML class for an ENTITY shall or shall not be present in a given VERSION FRAME (Most OpRa elements are otherwise optional in a VERSION FRAME, as are their attributes). Furthermore, it can indicate whether indirect references should also be included.

For example:

- To state that a **ServiceFrame** shall contain **ServiceLinks**, the **TypeOfFrame** associated with the **ServiceFrame** would have a **ClassInFrame** with 'nameOfClass="ServiceLink", **Mandatory**='true' and 'classRefType="members"
- To state that the **ServiceFrame** shall include any **ScheduledStopPoints** referenced by each **ServiceLink** (but not unreferenced stop points), the **TypeOfFrame** would have a **ClassInFrame** of 'nameOfClass="ScheduledStopPoint"' and 'classRefType="memberReferences"

Table 33 — *ClassInFrame* - Element

Classification	Name	Type	Cardinality	Description
«PK»	<i>nameOfClass</i>	<i>NameOfClass</i>	1:*	Reference to the XML Class of the ENTITY.
«enum»	<i>classRefType</i>	<i>ClassRefTypeEnum</i>	0:1	Type of XML Class reference. The default is 'members'. See allowed values below.
	<i>Mandatory</i>	<i>MandatoryEnum</i>	0:1	Whether element is required, optional or not allowed. Default is optional.
«cntd»	<i>attributes</i>	<i>ClassAttributeInFrame</i>	0:*	Requirements for an attribute of the XML class used to implement an ENTITY.
«cntd»	<i>relationships</i>	<i>ClassRelationshipInFrame</i>	0:*	Requirements for a attribute of the XML class used to implement a relationship of an ENTITY.

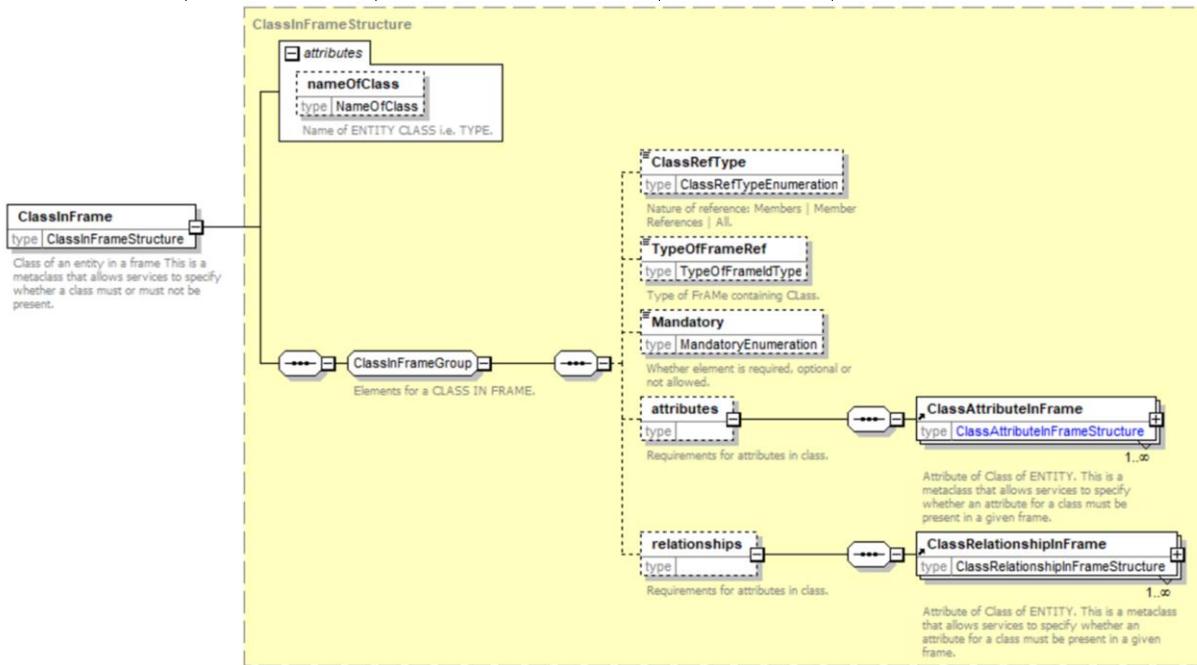


Figure 44 — *ClassInFrame* - XSD

10.7.5.3.6.2 ClassRefType - Allowed Values

Allowed values for class references (*ClassRefTypeEnum*).

Table 34 — *ClassRefType* - Allowed values

Value	Description
<i>all</i>	Include all known instances of class that meet validity conditions.
<i>members</i>	Include all instances of class that are direct components of another class.
<i>memberReferences</i>	Only include ancillary class instance if referenced by another instance.

10.7.5.3.6.3 Mandatory - Allowed Values

Allowed values for *Mandatory* (*MandatoryEnumeration*).

Table 35 — *Mandatory* - Allowed values

Value	Description
<i>required</i>	Include elements that meet selection criteria (e.g. validity condition).
<i>optional</i>	Include elements that are referenced by primary element. E.g. TYPES OF VALUE, OPERATOR etc.
<i>notAllowed</i>	Shall not be included

10.7.5.3.7 ClassAttributeInFrame- Model Element

Requirements for use of an XML Class representing a given entity in a VERSION FRAME. This is a metaclass that allows services to specify whether an XML class for an ENTITY shall or shall not be present in a given VERSION FRAME (Most OpRa elements are otherwise optional in a VERSION FRAME, as are their attributes). Furthermore, it can indicate whether indirect references should also be included.

For example:

- To state that a **ServiceFrame** shall contain **ServiceLinks**, the **TypeOfFrame** associated with the **ServiceFrame** would have a **ClassInFrame** with '**nameOfClass**="ServiceLink", **Mandatory**='required' and '**classRefType**="members"'.
- To state that the **ServiceFrame** shall include any **ScheduledStopPoints** referenced by each **ServiceLink** (but not unreferenced stop points), the **TypeOfFrame** would have a **ClassInFrame** of '**nameOfClass**="ScheduledStopPoint"' and '**classRefType**="memberReferences"'.

Table 36 — *ClassAttributeInFrame* - Element

Classification	Name	Type	Cardinality	Description
«atr», «PK»	nameOfClass	<i>NameOfClass</i>	0:1	Reference to the XML Class of the ENTITY.
«PK»	Type	<i>Xsd:QName</i>	0:1	Name of XML attribute of an ENTITY.
	Mandatory	<i>MandatoryEnum</i>	0:1	Whether any instances of class shall include attribute: required, optional or not allowed. Default is optional.

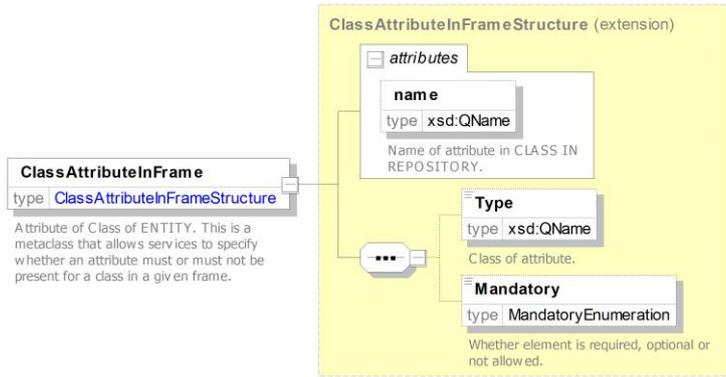


Figure 45 — ClassAttributeInFrame - XSD

10.7.5.3.8 ClassRelationshipInFrame- Model Element

Requirements for use of an XML Class representing a given entity in a VERSION FRAME. This is a metaclass that allows services to specify whether an XML class for an ENTITY shall or shall not be present in a given VERSION FRAME (Most NeTex elements are otherwise optional in a VERSION FRAME, as are their attributes). Furthermore, it can indicate whether indirect references should also be included.

For example:

- To state that a *ServiceFrame* shall contain *ServiceLinks*, the *TypeOfFrame* associated with the *ServiceFrame* would have a *ClassInFrame* with '*nameOfClass*="ServiceLink", *Mandatory*='required' and '*classRefType*="members"'.
- To state that the *ServiceFrame* shall include any *ScheduledStopPoints* referenced by each *ServiceLink* (but not unreferenced stop points), the *TypeOfFrame* would have a *ClassInFrame* of '*nameOfClass*="ScheduledStopPoint"' and '*classRefType*="memberReferences"'.

Table 37 — ClassRelationshipInFrame - Element

Classification	Name	Type	Cardinality	Description
«atr», «PK»	name	<i>xsd:QName</i>	0:1	Reference to the XML Class of the ENTITY.
«FK»	RelationshipRef	<i>RelationshipRef</i>	0:1	Reference to a relationship in metamodel.
	Mandatory	<i>MandatoryEnum</i>	0:1	Whether the relationship mandatory: required, optional or not allowed. Default is optional.
«enum»	Containment	<i>ContainmentEnum</i>	0:1	Nature of containment of attribute. See allowed values below.
«enum»	ModificationSet	<i>ModificationSetEnum</i>	0:1	Whether contained elements shall be whole set or can be just a Delta. See allowed values under DATA MANAGED OBJECT.

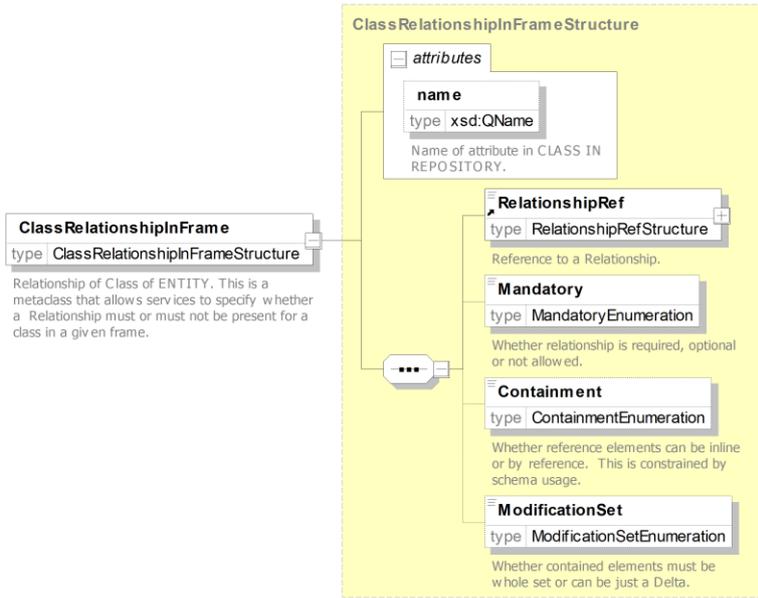


Figure 46 — ClassRelationshipInFrame - XSD

10.7.5.3.9 Class (ClassRef) – Model Element

An XML class implementing a model ENTITY. (A meta reference to an ENTITY).

Table 38 — ClassInFrame – Element

Classification	Name	Type	Cardinality	Description
«PK»	<i>nameOfClass</i>	<i>NameOfClass</i>	1:1	Identifier of XML element used to implement ENTITY that is available in VERSION FRAME.

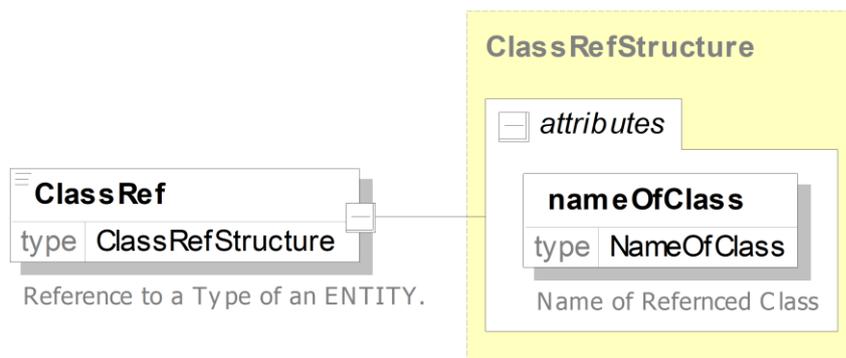


Figure 47 — ClassRef - XSD

10.7.5.3.10 Relationship (RelationshipRef) – Model Element

A specific relationship between two ENTITIES in the model of ENTITIES. (A meta reference).

Table 39 — RelationshipInFrame - Element

Classification	Name	Type	Cardinality	Description
«PK»	nameOfClass	RelationshipId	1:1	Identifier of relationship.

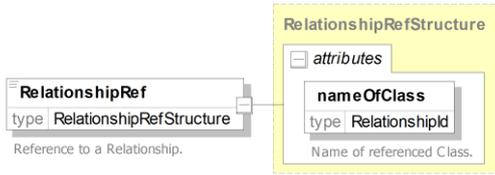


Figure 48 — Relationship (RelationshipRef) - XSD

10.7.5.4 XML Examples of Version Frames

10.7.5.4.1 Simple Version Frame - XML fragment

The following XML fragment shows a simple Service VERSION FRAME with two SCHEDULED STOP POINTs in it and a SERVICE LINK that connects them.

Example:

```
<ServiceFrame version="any" id="mybus:ntwkf001">
  <Name>My Network (V1) </Name>
  <frameValidityConditions>
    <AvailabilityCondition id="mybus:VC002_mf">
      <Description>Current </Description>
      <FromDate>2010-05-17T00:00:00Z</FromDate>
      <ToDate>2011-05-17T00:00:00Z</ToDate>
    </AvailabilityCondition>
  </frameValidityConditions>
  <scheduledStopPoints>
    <ScheduledStopPoint version="any" id="mybus:SSP0001A">
      <Name>Haltstelle A</Name>
      <Description>Stop A</Description>
    </ScheduledStopPoint>
    <ScheduledStopPoint version="any" id="mybus::SSP0002B">
      <Name>Haltstelle B</Name>
      <Description>Version one of stop B</Description>
    </ScheduledStopPoint>
  </scheduledStopPoints>
  <serviceLinks>
    <ServiceLink version="any" id="mybus:SL_AtoB01">
      <Name>Version one of Link from A to B</Name>
      <Distance>1.12</Distance>
      <FromPointRef version="002" ref="mybus::SSP0001A"/>
      <ToPointRef version="002" ref="mybus::SSP0002B"/>
    </ServiceLink>
  </serviceLinks>
</ServiceFrame>
```

Simple Version Frame with Condition - XML fragment

The following XML fragment shows a CompositeFrame VERSION FRAME with a VALIDITY CONDITION.

Example:

```
<CompositeFrame version="1" id="hde:CompositeFrame:CF_1">
  <frameValidityConditions>
    <ValidityRuleParameter version="1"
id="hde:ValidityRuleParameter:VR_1">
      <Name>Applies to area west</Name>
```

```

                <Description>Name </Description>
                <ConditionedObjectRef version="1"
ref="hde:CompositeFrame:CF_1"/>
                <RuleObjectRef version="any"
ref="hde:TopographicPlace:TP_1"/>
                <isValid>true</isValid>
            </ValidityRuleParameter>
        </frameValidityConditions>
        <frames>
        </frames>
    </CompositeFrame>

```

10.7.5.4.2 Use of Type Of Frame - XML fragment

The following example shows a frame definition for the weekly export of Neptune data elements.

Example:

```

<ResourceFrame version="any" id="Neptune:FrameProfile01">
    <Name>RATP--Neptune-Line Profile </Name>
    <typesOfValue>
        <TypeOfValidity version="01" id="Neptune:Weekly_planned">
            <Name>Weekley export of Neptune Planned data</Name>
            <Periodicity>P7D</Periodicity>
            <Nature>planned</Nature>
        </TypeOfValidity>
        <TypeOfFrame version="01" id="Neptune:Composite_frame">
            <Name>Neptune Profile: Composite Frame </Name>
            <Description>RATP--Neptune-Line-orderx frames will be composite frame
containing Service, Timetable (and Service Calendar) and Resource frames with all the
usual NEPTUNE attributes filled in, for a given line order.</Description>
            <TypeOfValidityRef version="01" ref="Neptune:Weekly_planned"/>
            <FrameClassRef nameOfClass="CompositeFrame"/>
            <classes>
                <ClassInContextRef nameOfClass="AvailabilityCondition"
classRefType="memberReferences"/>
            </classes>
            <typesOfFrame>
                <TypeOfFrame version="01" id="Neptune:Time_table_frame">
                    <Name>Neptune Profile Timetable Frame </Name>
                    <FrameClassRef nameOfClass="TimeTableFrame"/>
                    <classes>
                        <ClassInContextRef classRefType="members"
nameOfClass="TimeTableFrame"/>
                        <ClassInContextRef classRefType="members"
nameOfClass="ServiceJourney"/>
                        <ClassInContextRef classRefType="members"
nameOfClass="JourneyPart"/>
                        <ClassInContextRef classRefType="members"
nameOfClass="JourneyPart"/>
                    </classes>
                </TypeOfFrame>
                <TypeOfFrame version="01" id="Neptune:ServiceFrame">
                    <Name>Neptune Profile: Service Frame </Name>
                    <FrameClassRef nameOfClass="ServiceFrame"/>
                    <classes>
                        <ClassInContextRef classRefType="memberReferences"
nameOfClass="ScheduledStopPoint"/>
                        <ClassInContextRef classRefType="memberReferences"
nameOfClass="ServicePattern"/>
                        <ClassInContextRef classRefType="memberReferences"
nameOfClass="Network"/>
                    </classes>
                </TypeOfFrame>
            </typesOfFrame>
        </TypeOfFrame>
    </typesOfValue>

```

prTR OPRA:2025(E)

```

        <ClassInContextRef classRefType="memberReferences"
nameOfClass="RoutePoint"/>
        <ClassInContextRef classRefType="memberReferences"
nameOfClass="RouteLink"/>
        <ClassInContextRef classRefType="memberReferences"
nameOfClass="Line"/>
        <ClassInContextRef classRefType="memberReferences"
nameOfClass="Route"/>
        <ClassInContextRef classRefType="memberReferences"
nameOfClass="Direction"/>
        <ClassInContextRef classRefType="memberReferences"
nameOfClass="TariffZone"/>
        <ClassInContextRef classRefType="memberReferences"
nameOfClass="PassengerStopAssignment"/>
        </classes>
    </TypeOfFrame>
    <TypeOfFrame version="01" id="Neptune:Site_frame">
        <Name>Neptune Profile: Site Frame </Name>
        <FrameClassRef nameOfClass="SiteFrame"/>
        <classes>
            <ClassInContextRef classRefType="memberReferences"
nameOfClass="StopPlace"/>
        </classes>
    </TypeOfFrame>
    <TypeOfFrame version="01" id="Neptune:Service_calendar_frame">
        <Name>Neptune Profile: Frame </Name>
        <FrameClassRef nameOfClass="ServiceCalendarFrame"/>
        <classes>
            <ClassInContextRef classRefType="memberReferences"
nameOfClass="ServiceCalendar"/>
            <ClassInContextRef classRefType="memberReferences"
nameOfClass="DayType"/>
            <ClassInContextRef classRefType="memberReferences"
nameOfClass="DayTypeAssignment"/>
        </classes>
    </TypeOfFrame>
    <TypeOfFrame version="01" id="Neptune:Resource_frame">
        <Name>Neptune Profile: Resource Frame </Name>
        <FrameClassRef nameOfClass="ResourceFrame"/>
        <classes>
            <ClassInContextRef classRefType="memberReferences"
nameOfClass="TypeOfFrame"/>
            <ClassInContextRef classRefType="memberReferences"
nameOfClass="Operator"/>
            <ClassInContextRef classRefType="memberReferences"
nameOfClass="Authority"/>
        </classes>
    </TypeOfFrame>
</typesOfFrame>
</TypeOfFrame>
</typesOfValue>
</ResourceFrame/>
```

The following example shows a Frame definition for a frame to hold data elements to provide a GTFS timetable export.

Example

```
<TypeOfFrame version="1.5" changed="2011-01-20T00:00:00" id="gtfs:GTFS">
    <Name>GTFS Version Frame definition: A GTFS timetable has the following classes
</Name>
    <classes>
```

```

        <ClassInContextRef nameOfClass="AvailabilityCondition"
classRefType="memberReferences"/>
    </classes>
    <typesOfFrame>
        <TypeOfFrame version="1.5" changed="2011-01-20T00:00:00"
id="gtfs:Gtfs_Resource">
            <FrameClassRef nameOfClass="ResourceFrame"/>
            <classes>
                <ClassInContextRef nameOfClass="Authority"
classRefType="memberReferences"/>
            </classes>
        </TypeOfFrame>
        <TypeOfFrame version="1.5" changed="2011-01-20T00:00:00"
id="gtfs:Gtfs_Service_Calendar">
            <FrameClassRef nameOfClass="ServiceCalendarFrame"/>
            <classes>
                <ClassInContextRef nameOfClass="DayType"
classRefType="memberReferences"/>
            </classes>
        </TypeOfFrame>
        <TypeOfFrame version="1.5" changed="2011-01-20T00:00:00"
id="gtfs:Gtfs_Site">
            <FrameClassRef nameOfClass="SiteFrame"/>
            <classes>
                <ClassInContextRef nameOfClass="StopPlace"
classRefType="memberReferences"/>
            </classes>
        </TypeOfFrame>
        <TypeOfFrame version="1.5" changed="2011-01-20T00:00:00"
id="gtfs:Gtfs_Service">
            <FrameClassRef nameOfClass="ServiceFrame"/>
            <classes>
                <ClassInContextRef nameOfClass="Direction"
classRefType="memberReferences"/>
                <ClassInContextRef nameOfClass="Line"
classRefType="memberReferences"/>
                <ClassInContextRef nameOfClass="ScheduledStopPoint"
classRefType="memberReferences"/>
                <ClassInContextRef nameOfClass="TariffZone"
classRefType="memberReferences"/>
            </classes>
        </TypeOfFrame>
        <TypeOfFrame version="1.5" changed="2011-01-20T00:00:00"
id="gtfs:Gtfs_Timetable">
            <FrameClassRef nameOfClass="TimetableFrame"/>
            <classes>
                <ClassInContextRef nameOfClass="AvailabilityCondition"
classRefType="memberReferences"/>
                <ClassInContextRef nameOfClass="TimetableFrame"
classRefType="members"/>
                <ClassInContextRef nameOfClass="ServiceJourney"
classRefType="members"/>
                <ClassInContextRef nameOfClass="TemplateServiceJourney"
classRefType="members"/>
            </classes>
        </TypeOfFrame>
    </typesOfFrame>
</TypeOfFrame>

```

10.8 Responsibility

10.8.1 Introduction

NeTEx data will be used to provide many of the datasets for OpRa and it can have a complex organisational structure. For instance, information is planned, revised, forwarded, enriched, combined with other plans and forwarded again to the final user at some time. This process often involves several organisations or departments that each add, change or remove information in a complex workflow. These participating organisations can be strictly PT concerns, or also be external, such as governmental departments or other management agents. Which organisations are involved, what roles they have and what responsibility they execute cannot be determined beforehand for all possible environments in which NeTEx will be used. Even the structure and implementation of the processes for information planning, collecting and forwarding depend on various factors that cannot be determined beforehand. Hence, NeTEx has a generic organisational and responsibility model that can be applied in a variety of different environments and workflows and be used for a variety of purposes. The model in effect defines metadata as to the ownership of data that can be used to help manage the data.

Whilst the NeTEx responsibility model was designed with passenger information in mind it can be used by OpRa to help define the responsible party for raw and processed data for reporting and statistical purposes.

The use of the responsibility model in a specific situation or environment is optional.

The responsibility model makes it possible:

- To define operational responsibility for the real-life entities that are described by the information. For example, processes for a Stop Place information model it can specify which organisation is responsible for planning and maintenance of the physical stop.
- To define data management related responsibilities for the information itself. E.g. functional or technical IT data management regarding a set of produced, collected or forwarded plan information. This can be used to identify who needs to be contacted to correct or amend data.
- To exchange partial information falling under a certain responsibility set.

If used, the responsibility model can be applied to achieve the following goals:

- Provide as part of the raw data the contact information of agencies or help-desks to turn to in case of questions, queries about the raw data.
- Provide IT and PT related responsibility information for the purpose of management, assessment, etc. activities concerning quality Management and Quality Control.
- Associate Intellectual Property Rights with individual data elements or groups of elements.
- Allow delegation of data management: a receiving system can check the authorizations in relation to responsibility for provided data and see if the provider is authenticated to manage that data. This concept can be used to protect data in VERSION FRAMEs from being changed by the wrong parties.

Full details of how the responsibility model works is contained in the Transmodel and NeTEx documentation, a summary of the key model concepts is included here for ease of understanding the key concept for OpRa only.

10.8.2 Responsibility – Model Dependencies

The Responsibility Model forms part of the core NeTEx framework and is referenced by all other parts. There are three sub models. They extend the basic Entity & Versioning models:

The core RESPONSIBILITY model describes basic tracking attributes.

The RESPONSIBILITY ROLE model describes the roles different organisations may take.

The ORGANISATION model defines the common structures of an organisation. Note that this is further extended in the Reusable Components model (see later) with specific classes for specific types of organisations such as OPERATOR, AUTHORITY, SERVICED ORGANISATION, etc.

The Responsibility Model extends the basic Entity & Versioning models to create the fundamental framework classes from which all the useful NeTeX models are built.

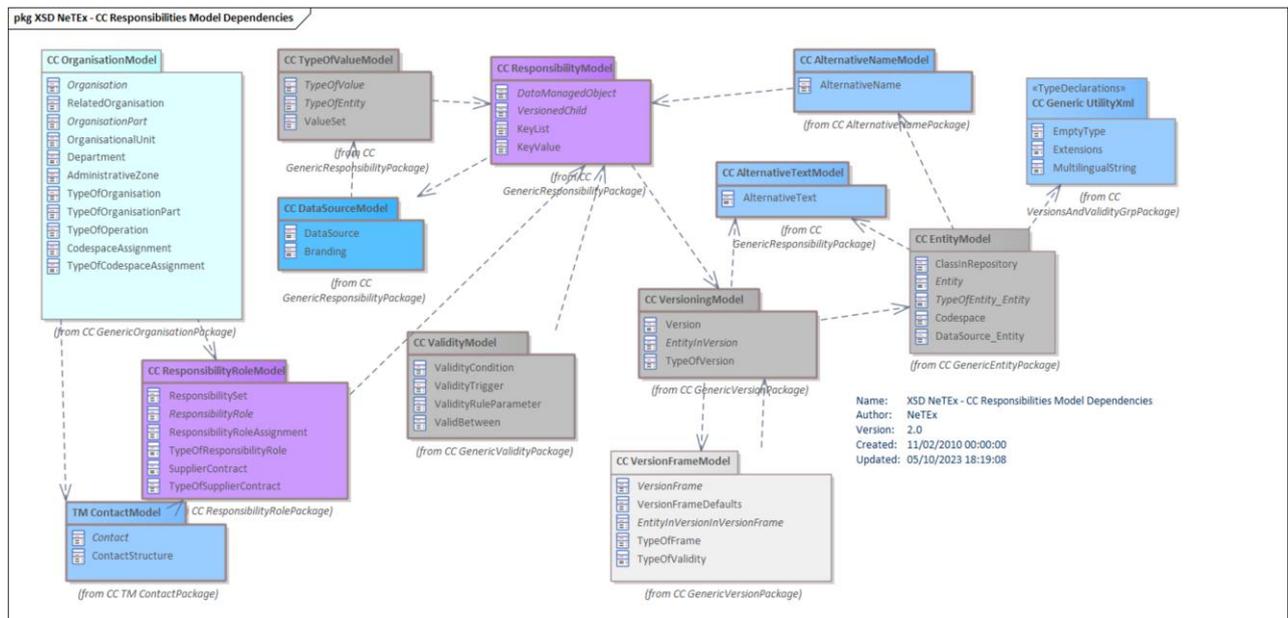


Figure 49 — Responsibility - Model Dependencies (UML)

10.8.3 Generic Responsibility

A certain aspect, or set of aspects, of responsibility in relation to an ENTITY is specified by associating a RESPONSIBILITY SET with the ENTITY. Each RESPONSIBILITY SET can contain one or more RESPONSIBILITY ROLE ASSIGNMENTS that allocate different types of RESPONSIBILITY ROLE to an ORGANISATION or a specific ORGANISATION PART.

RESPONSIBILITY SETs may be used at different levels of aggregation. It is possible to specify a different set for each different ENTITY (or rather ENTITY IN VERSION), or just at the Frame Level. The RESPONSIBILITY SET for an ENTITY may change in successive ENTITY IN VERSIONS.

The RESPONSIBILITY ROLE describes the kind of responsibility that is enacted; the RESPONSIBILITY ROLE ASSIGNMENT assigns the responsibility to the RESPONSIBILITY SET.

The ADMINISTRATIVE ZONE and RESPONSIBILITY ROLE ASSIGNMENT are used to describe the specific situation of the delegation of the regional responsibility of an authority to an organisation. This can be e.g. the delegation using a concession for the operation of a PT service or the delegation of a regional travel information provision service.

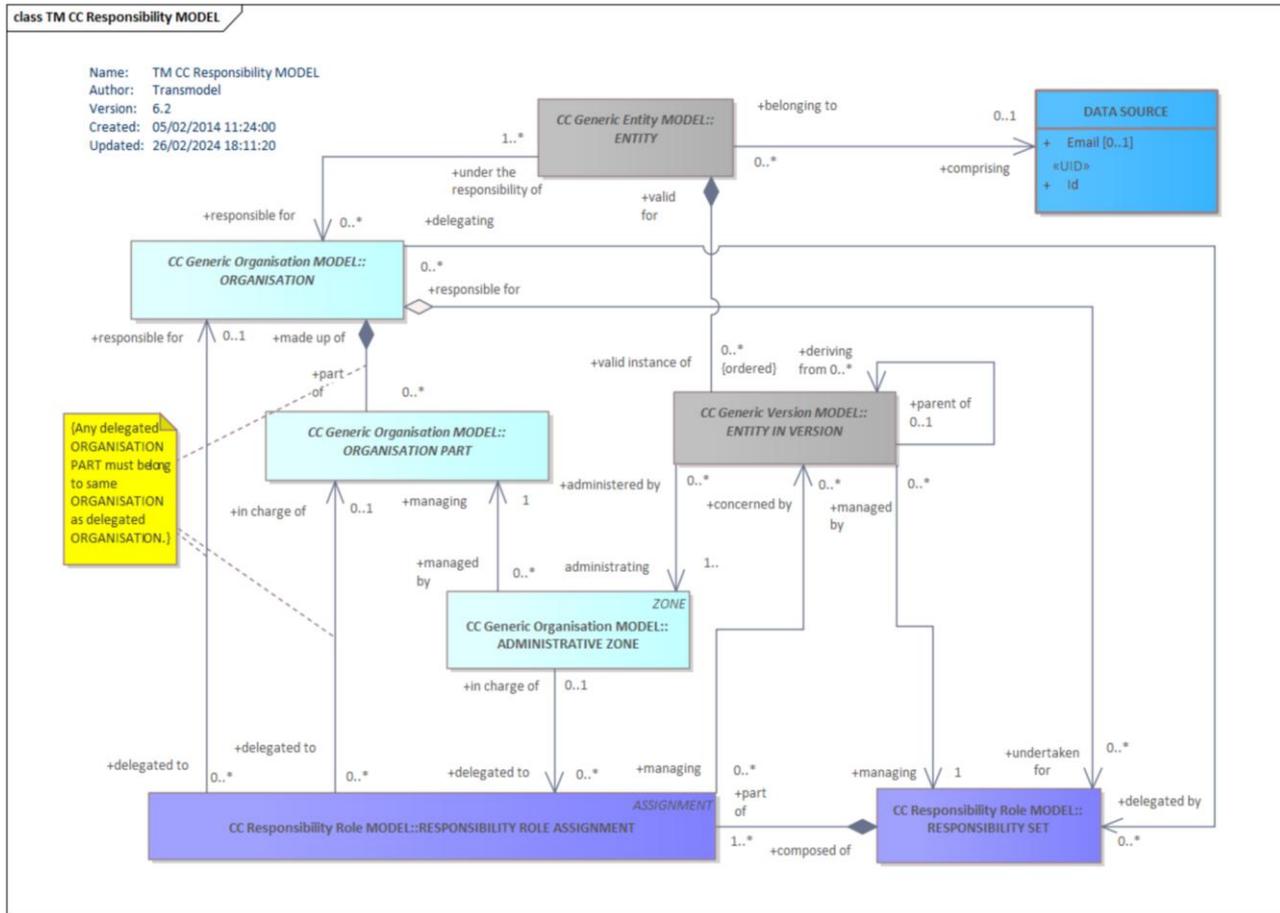


Figure 50 — Responsibility - Conceptual MODEL (UML)

10.8.3.1 Generic Responsibility - Physical Model

10.8.3.1.1 Summary of Responsibility model

The following diagram summarises the Physical model. All NeTeX entities are instances either of DATA MANAGED OBJECT or VERSIONED CHILD.

Only DATA MANAGED OBJECT may have its own RESPONSIBILITY SET. For a composite object, such as a SERVICE PATTERN made up of a number of dependent POINT IN PATTERN instances, all the child elements (which will be a subtype of VERSIONED CHILD) shall take the same RESPONSIBILITY SET as their parent.

The detailed attributes of the Responsibility Physical Model are shown in separate further diagrams.

of the calendar dates covered by this OPERATING DAY, fixed arbitrarily and in most cases on the first calendar date.

A PERIOD is a continuous interval of several days between two particular OPERATING DAYS, which can be used for several purposes (e.g. VALIDITY CONDITION of a VERSION).

10.10 Notation of XML-Elements and XML-Structures

10.10.1 General

The OpRa interfaces presented in this document are defined using XML schemas. Thus, the objects that are exchanged over the interface exist as XML elements. The description of XML elements are displayed in tabular form in the current document, which originated from SIRI (EN 15531-3). It is very compact and clearly arranged and provides a range of structural information reflecting the XML scheme definition. Section 10.10.4 describes the notation of the table format which is used more intensively afterwards.

All element names, data types and attributes are written in English to make exchange with European partners more convenient.

10.10.2 Display of XML Elements in the Text

This document is intended to provide consistent notation of XML elements to make technically important information available to the reader.

- XML elements are written in upper camel case, bold and italics, e.g.: ***VehicleJourneyRef***. The element names – where possible and meaningful – are derived from Transmodel terms. If a corresponding term for a concept or object is missing in Transmodel, an attempt was made to adopt a corresponding term from JourneyWeb or a matching concept from DELFI.
- Data types are displayed italics, e.g. : *xsd:boolean*.
- Code examples are displayed in the Courier font.

10.10.3 Display of Relationships

When a dataset described by an object model (ie: a set of object instances) is exchanged as a data request (for example in an XML based API such as OpRa) it will be serialised into a stream of bytes representing both the data values and the relationships between the instances, both marked by delimiters such as XML tags. In practice there are three ways of representing the relationships between object instances when serialised:

- implicit mechanisms,
- explicit internal references, or
- explicit external references

An implicit mechanism is used when one element is contained in another, i.e., grouped spatially within child tags. In this way a direct child relationship can be expressed by immediate proximity. An explicit mechanism, in contrast, indicates a relationship by the use of the key of the referenced object as an explicit reference; an (explicit) internal reference is an object key where the referenced object is also defined in the same document as the referencing object (and so whose scope of uniqueness need only be local to the exchange, e.g., an identifier of a message); an external reference is an object key whose referent is defined outside the document (and whose scope shall be defined by a namespace common to

both parties of the exchange, e.g., a persistent stop number). References may consist of compound keys (for more detail see 11.2).

It is helpful to be able to distinguish between the use of an identifier to provide the key of an object as part of the object definition and its use as a reference to indicate a relationship to the object from another object. In OpRa, the following convention is used to do this:

- An identifier is an attribute (i.e., a child element) of the defining element that indicates a primary key for the defining element. These identifiers end with a signal word like “code” or “identifier” (sometimes “number” in SIRI). Example: a journey contains the key ***JourneyCode***.
- If an object is referenced in another object, the referencing element ends with “Ref”. For example, the reference for a journey (from a departure board) is: ***JourneyRef***.
- An object instance and its reference use a common underlying data type for the identifier. For example ***JourneyCode*** and ***JourneyRef*** are both of type ***JourneyCodeType***. This allows for a degree of automated type checking so that only objects of the allowed type are made.

10.10.4 Table Notation of XML Structures

10.10.4.1 General

The current document displays XML structures in tabular notation. There is a separate table for each important OpRa-request/response-element. Additional tables are indicated for all essential child elements from which the complex structures are built. In order to save space, the column headings are only displayed in Table 2 and not repeated for the following tables. They use a consistent set of rules to describe XML elements and their related conditions.

Table 40 — Example for the tabular notation of an XML structure

Grouping	Element Name	Min: Max	Data Type	Explanation
ContinuousServiceStructure				[a special form of SERVICE JOURNEY in Transmodel v6] a vehicle movement on a continuous, non-timetabled service.
	a ContinuousMode	-1:1	<i>ContinuousModes</i>	Continuous transport options.
	b IndividualMode	-1:1	<i>IndividualModes</i>	Individual transport options.
DatedJourney	a → <i>OperatingDayRef</i>	1:1	→ <i>OperatingDayRef</i>	Reference to an Operating Day
	a → <i>siri:VehicleRef</i>	0:1	→ <i>siri:VehicleRef</i>	Contains the Vehicle reference of the vehicle. In Transmodel this may be the VEHICLE Id.
ScheduledJourney	a → <i>JourneyRef</i>	1:1	→ <i>JourneyRef</i>	Reference to a Journey
	a PublicCode	0:1	<i>xs:normalizedString</i>	Public code of the SERVICE JOURNEY (same meaning as in NeTEx). Examples: "512", "S8" or "Circle Line" or "ICE 488".
	a → <i>siri:LineDirectionGroup</i>			
Service	a Mode	1:1	<i>+Mode</i>	[a specialisation of MODE in TMv6] an extended range of VEHICLE MODEs, aggregating them with some SUBMODEs
	a ProductCategory	0:1	<i>+ProductCategory</i>	A product category for the service. This is a classification defined in

				NeTEx/SIRI used to identify groups of journeys with some special properties for marketing and fare products, e.g. "TE2" for SNCF or a special panorama train "PE" in Switzerland.	
	a	PublishedServiceName	1:1	+InternationalText	Line name or service description as known to the public, f.e. "512", "S8" or "Circle Line" or "ICE 488".
	a	TrainNumber	0:1	xs:normalizedString	Contains the TrainNumber from NeTEx (TRAIN NUMBER from Transmodel). If several TrainNumber types exist, use the commercial one. In some cases also non-train modes will use TrainNumber.
	a	→siri:VehicleRef	0:1	→siri:VehicleRef	Contains the Vehicle reference of the vehicle. In Transmodel this may be the VEHICLE Id.
	a	OperatorRefs	0:1	+OperatorRefs_Rel	References to the OPERATORS. Multiple OPERATORS in case a ContinuousLeg can be operated by more than one operator, especially in the ALTERNATIVE MODE OF OPERATION where there can be dozens of taxi companies and many free floating sharing companies.
	a	RouteDescription	0:1	+InternationalText	Descriptive text for a route, f.e. "Airport via City Centre"
	a	Via	0:*	+ServiceViaPoint	Via points of the service that may help identify the vehicle to the public.
	a	Attribute	0:*	+GeneralAttribute	Note or service attribute.
	b	SharingService	-0:1	+SharingService	Shared mobility offer.
ServiceOrigin		OriginStopPointRef	0:1	siri:StopPointRefStructure	First stop of the vehicle journey; origin stop point.
		OriginText	1:1	+InternationalText	Label for first stop.
ServiceDestination		DestinationStopPointRef	0:1	siri:StopPointRefStructure	Last stop of vehicle journey; destination stop point.
		DestinationText	1:1	+InternationalText	Label for last stop.
		BookingArrangements	0:1	+BookingArrangementsContainer	Container with information on booking possibilities for this service.
		SituationFullRefs	0:1	+SituationRefList	A list of references to SITUATIONS.
		Extension	0:1	xs:anyType	

10.10.4.2 Grouping

The first column, when filled, contains an identifier that organises the elements into meaningful groupings, e.g. *Service* or *ServiceOrigin*. This is purely for documentation purposes and corresponds in most cases to the names of an XML group that is used in the XML schema. The use of groupings is intended to organise elements to improve clarity and reusability.

In case of nested groups, only the innermost group is displayed next to the elements it contains. The table notation is not able to reflect all the grouping information and this may lead to ambiguities.

10.10.4.3 Element Name

Element names are listed in the second column, e.g. *OperatingDay*. If an element is required, it is written in bold. Optional elements are not printed in bold. The name of the structure itself is displayed in the upper left of the table.

In some cases, a SIRI group name may appear in this column (e.g., →*siri:LineDirectionGroup*), while the elements of the group are not listed. There are proper tables for selected structures and groups from SIRI; they may have three colons “:::” in the name field if the elements are derived by extension or used anonymously.

10.10.4.4 Multiplicity & Choice (min:max)

The third column (Min:Max) indicates whether an element is required or optional or whether it occurs once or multiple times in the superior element. For this purpose, the standard UML conventions “min:max” are used, so that, e.g. “0:1” stands for an optional, single element, “1:1” indicates a required, single element, “0:*” stands for optional multiple elements, etc. Required elements are written in **bold**.

In some cases an element has to be selected from its set (XML-Choice), which is symbolised by a prefixed minus sign, e.g., “-1:1”. In this case a lowercase letter (a, b, ...) precedes the element name. The letters indicate the different choices. A zero in the min-value indicates an optional choice: “-0:1”.

10.10.4.5 Data Type

The data types are displayed in the fourth column in italics, e.g. *InternationalText*. If the namespace differs from an OpRa namespace, it is also indicated, e.g. “*xs:dateTime*” or “*siri:PtSituationElement*”.

- A complex data type, which itself contains structures as child elements, is indicated in the data type column with “+Structure”.
- Where elements are used as references (external or explicit key) for other objects, the name of the element contains the keyword *Ref* or *Reference* and most of the time this also applies to the type (e.g., *LineRef*, *siri:LineRefStructure*). Where elements are used as references (external key) for other objects, the type of referenced object is used as a data type with a preceding arrow. For example “→StopPoint” as the type of a reference (StopPointRefStructure) for an object of type “StopPointType”.
- Enumerated types are provided with a hyperlink to the table containing the corresponding definition and the allowed values (e.g., walk | cycle”).
- In many cases the type is provided with a hyperlink leading to the corresponding section and table describing the structure.
- In order to save space, data types have been truncated. So endings like “Structure” and “Type” are not written out. Instead of “InternationalTextStructure” the data type will be indicated as “InternationalText”.

10.10.4.6 Explanation

The intended use of all elements is explained in the last column. Other passages in the text are referred to, so for example by complex child elements in the spot where their table description can be found. In some places, the explanation is too elaborate and would go beyond the limits of the table format. In this case, notes can be found underneath the tables.

10.10.4.7 Message Exchange

This chapter explains how OpRa messages are exchanged. There are two basic procedures in use

- Request with synchronous response (request response procedure),
- Subscriptions with asynchronous messages (publish subscribe procedure).

These procedures are already established and in use, e.g. in the SIRI interfaces.

10.10.5 Use of SIRI Procedure

In SIRI, the message exchange procedure mentioned above has been defined and described, see EN 15531-2. This procedure has some advantages: previously tested processes and existing SIRI implementations can be used when implementing OpRa services, something that can save time and costs.

The basic procedure is the request with synchronous response. A client sends a request to a server, which answers immediately. In SIRI terminology, the requester is the data consumer, the responding server is called the data producer (see **Error! Reference source not found.**).

Figure 52 — Request with synchronous response (from SIRI, EN 15531-2)

Requests with synchronous response are used for every OpRa service. The role of the requester is taken on by the Home System, which sends requests to the Responding System. If a Distributing System is "in the middle" then a concatenation of the data request / response pattern applies.

Somewhat more complicated is the subscription mechanism. A data consumer is interested in new messages but does not know when they will occur. Instead of making regular requests and thus creating a base load (and to risk learning about the new message only as a result of the next regular request), they can set up a subscription.

Figure 54 shows the basic connections. The data consumer has to fill two roles, that of the subscriber and that of the notification consumer. The data consumer makes a subscription request from the server. In doing so, they indicate to the server which type of incidents should be sent. The server sets up the subscription by registering it with the subscription manager. Afterwards, something occurs when an incident arises that needs to be communicated to consumers. In this case the server, as the notification producer, sends the message to the data consumers with the new incident (delivery). This process is repeated until the subscription runs out or is ended by the data consumers.

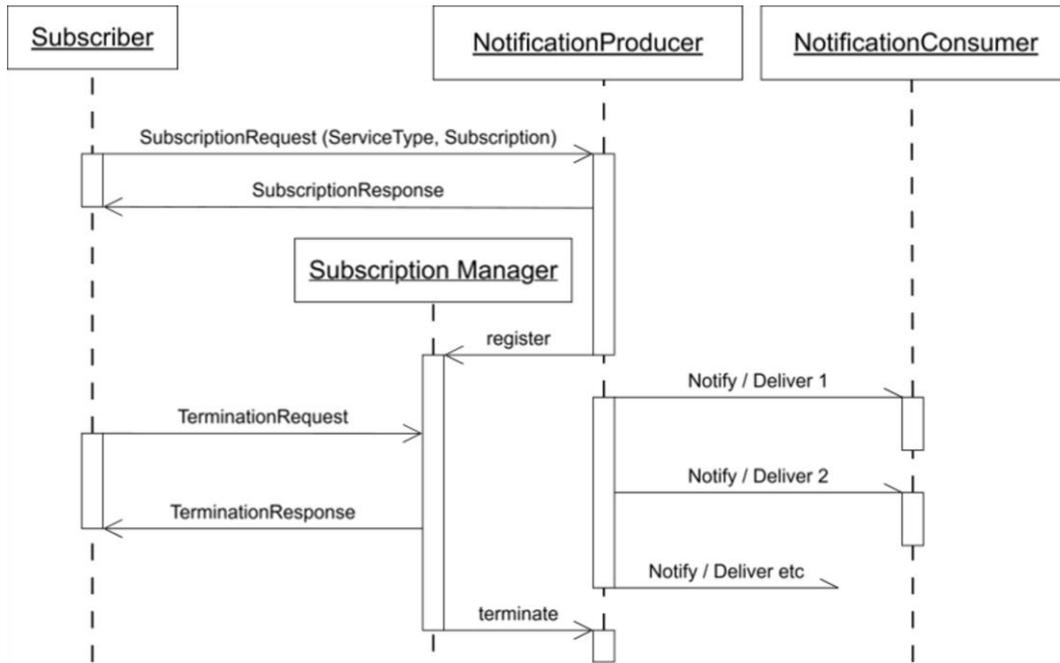


Figure 53 — Publish / Subscribe with asynchronous notifications (from SIRI, EN 15531-2)

Both client and server must fulfil two roles, the client has the roles of the subscriber and notification consumer, and the server has the roles of the notification producers and subscription manager. In most implementations there is no distinction, and each individual software component fulfils both roles.

The subscription process is completed by additional requests. The status requests allow the status of the interface partner to be requested and its availability to be tested. The heartbeat request, one which is sent regularly by the server, enables a data consumer to recognise when a server is available and sends signals (ping or heartbeat). Details can be found in SIRI (EN 15531-2), Chapter 5.

In OpRa services the subscription mechanism exists as a notification service when a data consumer (through their Home System) wants to inform themselves about incidents or other occurrences.

10.10.6 HTTP and REST

The conversion of SIRI messaging procedure occurs in OpRa using HTTP/1.1 (Hypertext Transfer Protocol) as a transport protocol and XML (Extensible Markup Language) for the message content. This is done following the rules of the the SIRI Lite protocol described in "**Chapter 12- SIRI for Simple Web Services – SIRI Lite**" of EN 15531-2.

An HTTP-request is immediately responded to by a server using the previously opened IP port.

If multiple requests are sent in rapid succession, the HTTP mechanism “keep alive” can be used to keep a port open for a time for reuse to prevent frequent opening and closing of the port.

For larger messages, which are likely in some of the OpRa use cases, the use of a compression procedure is recommended. This type of method is also specified for HTTP.

System designers will need to consider the need for security and assurance of data, such as using VPN or HTTPS technologies – but this is beyond the scope of this Technical Specification.

10.10.7 Roles of Server and Client

Using the synchronous request-response-procedure, a data consumer (the requester) is an HTTP client, the data provider (the responding server) an HTTP server.

Only for the notification service, if the subscription procedure is used, is the situation more complicated. In this case, the data consumer and the data provider both have to assume the roles of client and server in the sense of HTTP. Because when the data provider (notification producer) wants to send a message to the data consumer, it becomes a client (in the sense of HTTP) and the data consumer then becomes a server. In other words, the active part that initiates a communication is always an HTTP-client and the passive part that responds to a message always plays the role of an HTTP-server.

10.11 Service Discovery

This section will be developed based on the SIRI Discovery service following CIB.

11 Services and XML Schemas

11.1 General

This document describes interface definitions for services between software components.

The OpRa interface services are defined as XML schemas. An overview of the services and their implementation as XML schemas makes up the first part of the section below. Several structure definitions are useful in multiple services and therefore are hierarchically defined in several schema files as a common basis to enable repeated use. The approach followed is very strongly oriented towards the principles of object orientation. The commonly used structure definitions are described in the second part of this section. The third part introduces the XML schemas that are imported from SIRI. A classification of error states can be found in the fourth part of this section.

11.2 Services Provided

Table 41 — List of the OpRa services and their request elements

Service	Name of request element	Schema file	Section
Publication Request	PublicationRequest	OpRa_publication.xsd	
Request / Response Delivery	OpRaRequests	opra_request.xsd	

11.3 XML Schemas Used Across Services

In order to avoid the need to redundantly define structures that are used in more than one service, a commonly used basic XML schema was introduced to include them hierarchically. The inclusion sequence and the customization of schema files have been selected to place content-related elements together in a file and to only include in each schema, that which is the required for the task.

The hierarchic dependency between the individual schemas is shown in the following figures. The blue-coloured schema files are imported from SIRI. The documentation for the SIRI parts can be found in the SIRI specification. The schemas can be found here: <https://github.com/SIRI-CEN/SIRI> . The ??-coloured schema files are imported from NeTEx. The documentation for the NeTEx parts can be found in the NeTEx specification. The schemas can be found here: <https://github.com/NeTEx-CEN/NeTEx>.

The XSD file describing the XML schema can be freely downloaded from <https://github.com/OpRa-CEN/OpRa>. This includes examples and limited documentation.

11.4 Imported Schemas

Several schema files are imported from the SIRI and NeTeX interfaces to OpRa. By importing from SIRI definitions, the SIRI process for message exchange can also be used for the exchange of OpRa messages. Additionally, specific structure definitions can be reused from SIRI and NeTeX, a fact that ensures the consistence between these interface standards. This affects the definition of modes, incidents and stop assets or facilities.

The imports have been taken from SIRI v2.1 and NeTeX 2.0

11.5 Problems and Error States When Operating OpRa Services

Problems and error situations may arise at different levels when requesting or receiving a response. To support multi-layered error situations, problems can be signalled at the following levels:

- Errors regarding message transfer inherited from the SIRI and NeTeX services
- Problems describing general OpRa error situations across services (codes OPRAGENERIC_*)
- Service-specific problems (codes, e.g., STOPEVENT_*)
- SIRI and NeTeX errors allowing for providing additional details (e.g., indicating unsupported parameters)

11.5.1 Error Codes from SIRI

In SIRI (EN 15531-2), section 5.7, a number of error codes are defined that play an important role for the message transfer procedure. These codes have been divided into the following groups: success, systemic error and application error. The table lists the error codes taken from SIRI v2.1; the OpRa schema, however, does not specify which of these codes have to be used. Each system should use those which are relevant and any additional ones that may be necessary.

Table 42 — List of error codes as they are defined in SIRI

Group	Condition	Description
Success	OK (true)	Request successful
Systemic Error	<i>RequestTimeout</i>	Server not responding.
	<i>InvalidRequest</i>	The server does not "understand" the request. The client should not repeat the request.
	<i>Unauthorized</i>	User name and password are required for the request, or credentials not satisfied.
	<i>Forbidden</i>	The server "understands" the request, but cannot carry it out.
	<i>NotFound</i>	The requested URL was not found.
Distribution	<i>UnapprovedKeyAccessError</i>	Error: Recipient of a message to be distributed is not available.
	<i>UnknownParticipant</i>	Recipient for a message to be distributed is unknown.
	<i>UnknownEndpoint</i>	Endpoint to which a message is to be distributed is unknown.

Group	Condition	Description
	<i>EndpointDeniedAccess</i>	Distribution message could not be delivered because not authorised.
	<i>EndpointNotAvailable</i>	Recipient of a message to be distributed is not available.
Access	<i>UnapprovedKey</i>	User authentication key is not approved. SIRI v2.0
Application Error	<i>VersionNotSupported</i>	Service is not available.
	<i>CapabilityNotSupported</i>	Service does not support the requested capability.
	<i>ServiceNotAvailable</i>	Functional service is not available to use (but it is still capable of giving this response).
	<i>AccessNotAllowed</i>	Requestor is not authorised to the service or data requested.
	<i>InvalidDataReferences</i>	Request contains references to identifiers that are not known.
	<i>BeyondDataHorizon</i>	Request is for data outside of real-time data horizon.
	<i>NoInfoForTopic</i>	Valid request was made but service does not hold any data for the requested topic expression.
	<i>ParametersIgnored</i>	Request contained parameters that were not supported by the producer. A response has been provided but some parameters have been ignored.
	<i>UnknownExtensions</i>	Request contained extensions that were not supported by the producer. A response has been provided but some or all extensions have been ignored.
	<i>UnknownSubscriber</i>	Subscriber not found.
	<i>UnknownSubscription</i>	Subscription not found.
	<i>AllowedResourceUsageExceeded</i>	Valid request was made, but request would exceed the permitted resource usage of the client.
<i>OtherError</i>	Other Error Type.	

11.5.2 OpRa ErrorCondition

The xxxDelivery and xxxResult structures of each service contain elements ErrorCondition to transmit error information. ErrorCondition can either hold a siri:Error or an OPRAErrorStructure. OJPErrorStructure should always be used if one of the pre-defined error codes applies to the situation

Table 43 — OPRAErrorStructure

OPRAErrorStructure		siri:OtherErrorStructure	Type for Error: OPRAError.
	→Error	1:1	→ <i>Error</i>
ProblemDetail	Details	0:1	+ <i>InternationalText</i>
	LogData	0:1	<i>xs:string</i>

The codes (element *Error*) available for each service are listed in the sections describing the respective service deliveries

11.5.3 General OpRa Problems

To signal general OpRa problems that may appear with any service delivery, the *OJPGenericProblemType* is used. It supports the following codes:

Table 44 — General OJP problems that can appear in any service delivery

OJPGenericProblemType	Description
<i>OPRAGENERIC_REQUESTNOTSUPPORTED</i>	The server does not support the specific request.
<i>OPRAGENERIC_FEATURENOTSUPPORTED</i>	The server does not support the requested feature
<i>OPRAGENERIC_LANGUAGENOTSUPPORTED</i>	For the display of texts within the result, the server does not support (at least within the context of this request) the language required by the requestor.
<i>OPRAGENERIC_EXCEPTIONFROMREQUESTEDLANGUAGE</i>	For the display of texts within the result, the server does not support the language required by the requestor for all of the occurring text elements.
<i>OPRAGENERIC_DATAFRAMEREFNOTAVAILABLE</i>	The server cannot provide the data frame (data version) required by the requestor.
<i>OPRAGENERIC_SYSTEMID_NOT_FOUND</i>	The server was not able to work with the SystemID that was provided.
<i>OPRAGENERIC_OTHER</i>	A problem has occurred that does not have a specific problem type.

12 Common XML Structures

12.1.1 General

This chapter is intended to explain basic XML structures that are commonly used by several OpRa services. The organization of the sections in this chapter follows the structure of the XML schema files.

12.2 Root Element OpRa

12.2.1 General

The XML schema file OpRa.xsd defines the general root element OpRa which is shared by all messages of all OpRa services.

The root element is named *OpRa* and is a choice between *OPRAPublicationRequest* (*OPRAPublicationRequestStructure* defined as a NeTeX *NetworkFrame*) and *OPRAPublicationDelivery* (*OPRAPublicationDeliveryStructure* defined as a NeTeX *PublicationDeliveryStructure*).

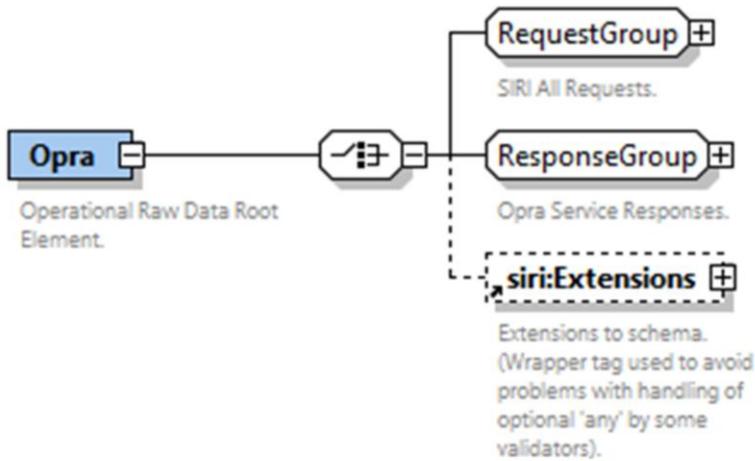


Figure 54 — Content of OpRa.xsd

The subsequent sections describe the complex structures defined in OpRa.xsd.

12.2.2 Service Request Structure

Table 45 — Description of ServiceRequestStructure

	<i>ServiceRequestStructure</i>		<i>+Structure</i>	Request from a Consumer to a Producer for immediate delivery of data. Answered with a <i>ServiceDelivery</i> . (For <i>Fetches Delivery</i> this will be after a further <i>DataReadyRequest</i>).
	<i>ServiceRequestContext</i>	0:1	<i>+Structure</i>	General request properties – typically configured rather than repeated on request.

12.2.3 Frames

The OpRa payload uses Frames to manage the data content being transferred

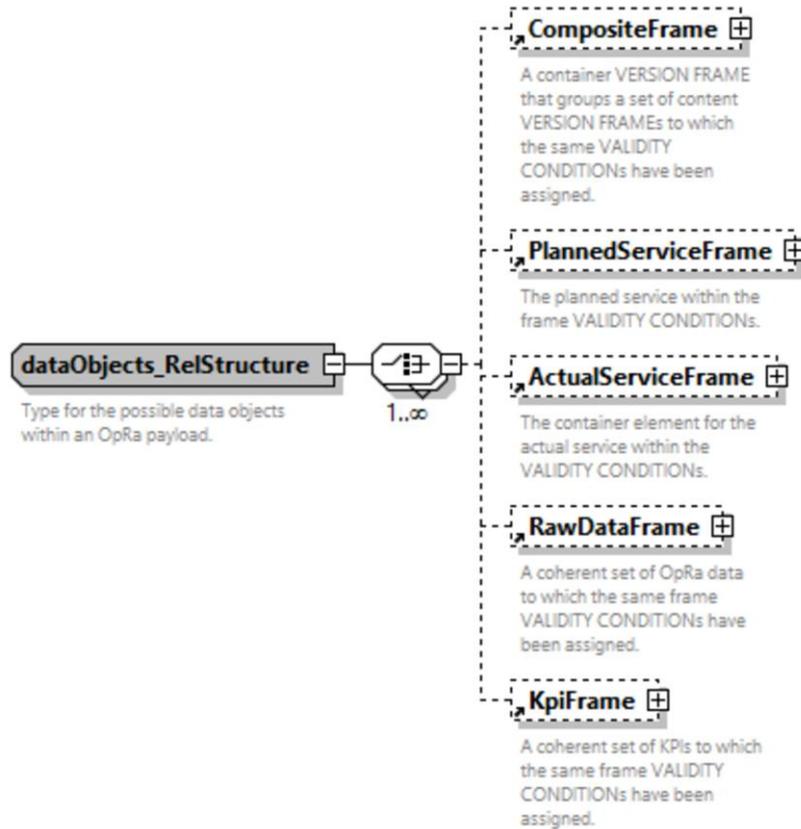


Figure 55 — OpRa data frames

Table 46 — OpRa data frames

Frame Name	Description	Described in paragraph
CompositeFrame	A container VERSION FRAME that groups a set of content VERSION FRAMEs to which the same VALIDITY CONDITIONs have been assigned.	10.7.6
PlannedServiceFrame	The planned service within the frame VALIDITY CONDITIONs.	12.6.1
ActualServiceFrame	The container element for the actual service within the VALIDITY CONDITIONs.	12.6.2
RawDataFrame	A coherent set of OpRa data to which the same frame VALIDITY CONDITIONs have been assigned.	12.6.3
KpiFrame	A coherent set of KPIs to which the same frame VALIDITY CONDITIONs have been assigned.	12.6.4

12.3 Generic XML Issues

12.3.1 Indicator-related base definitions

Implementations of each use case consist of a specific INDICATOR / INDICATOR LOG ENTRY design pattern, which follows the LOGGABLE OBJECT / LOG ENTRY design pattern. INDICATOR LOG ENTRY instances can provide states (“history”) or transactions (“deltas”) about the changes to the INDICATOR through its lifecycle.

The opra_GeneralLogEntry.xsd contains base definitions for all INDICATORS following the above-mentioned design pattern:

Table 47 — Design patterns for INDICATOR definitions

Name	XSD node type	Additional information
LoggableObjectIdType	simpleType	A specialization of <i>netex:ObjectIdType</i>
LoggableObjectRefStructure	complexType	Type of reference to a LOGGABLE OBJECT. Specialization of <i>netex:VersionOfObjectRefStructure</i> . Restricts <i>id</i> to type <i>LoggableObjectIdType</i> .
LogEntry_VersionStructure	complexType	Type of the INDICATOR in a version, that extends <i>netex:DataManagedObjectStructure</i> .

For each use case, a separate .xsd has been created that consists of:

- The versioned INDICATOR element, type of which is an extension of *LoggableObject_VersionStructure*. The element substitutes for the *LoggableObject_* abstract element. The extension consists of exactly one additional group, which is named after the INDICATOR. The group contains a collection of INDICATOR LOG ENTRIES, which has to have at least one INDICATOR LOG ENTRY.
- The INDICATOR LOG ENTRY element of type *netex:LogEntry_VersionStructure*
- The definition of the collection of the INDICATORS elements.
- The definition of the collection of the INDICATOR LOG ENTRY elements.

The XSD naming conventions used are detailed in the table below.

Table 48 — XSD naming conventions INDICATOR

Naming format	XSD node type	Additional information
[Indicator]IdType	simpleType	A specialisation of <i>LoggableObjectIdType</i> .
[Indicator]Ref	element	Reference element definition for the INDICATOR
[Indicator]RefStructure	complexType	Type to refer to an INDICATOR instance which is identified by <i>[Indicator]IdType</i> .
[Indicator]ScopeGroup	group	Additional element references that extend the common PT SCOPE and

		the corresponding PtScopeGroup.
[Indicator]	element	Element of type <i>[Indicator]_VersionStructure</i> . Substitutes for <i>LoggableObject_</i> abstract element. The “id” attribute is restricted to <i>[Indicator]IdType</i>
[Indicator]_VersionStructure	complexType	
[Indicator]Group	group	INDICATOR-specific data elements’ sequence. Consists of <i>Generic CalendarAwarePtScopeGroup</i> <i>[Indicator]ScopeGroup</i> <i>[Indicator Log Entry]</i> collection (if applicable) Statistics-related groups
[Indicators]_RelStructure	complexType	The collection of one or more INDICATORS of the same type.
[Indicator Log Entry]	element	Element of type <i>[Indicator Log Entry]_VersionStructure</i>
[Indicator Log Entry]_VersionStructure	complexType	Type of the INDICATOR LOG ENTRY in a version
[Indicator Log Entry]Group	group	INDICATOR LOG ENTRY-specific data elements
[Indicator Log Entries]_RelStructure	complexType	The collection of one or more INDICATOR LOG ENTRIES of the same type.

12.3.2 Generic scope for indicators

To provide segmentation capability to managing indicators, PT SCOPE is applied to an indicator as the relevant subset of the organizational, network, and journey model. To add calendar aspects (day types, operational days, time bands), PT SCOPE has been extended into CALENDAR AWARE PT SCOPE that introduces references to the calendar-related entities.

- status request (check status / heartbeat notification)

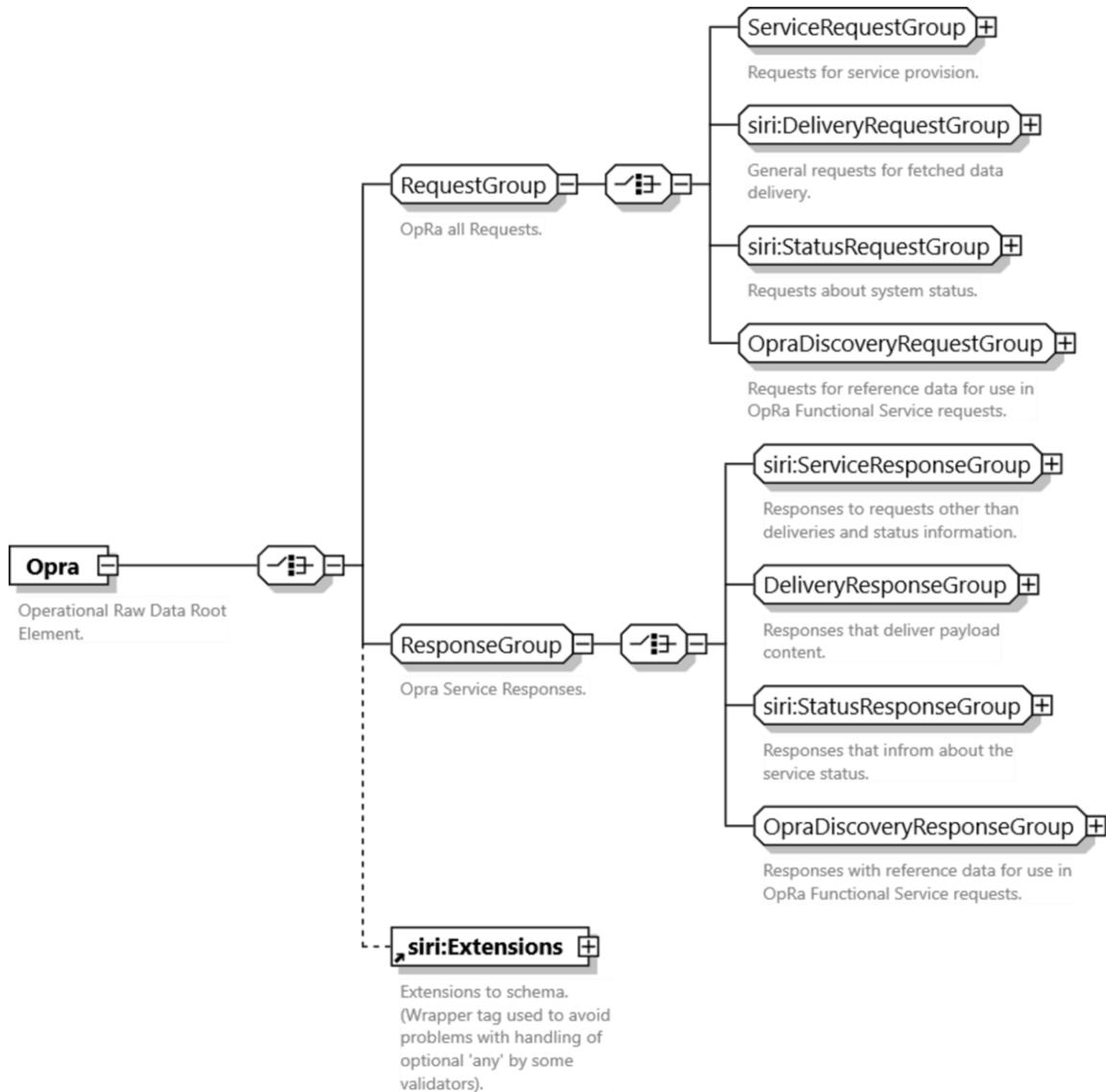


Figure 57 — OpRa Service Capabilities

12.4.1 Service Capabilities Modelling Approach

OpRa service framework defines abstract OpRa base elements, which are to be instantiated via the substitutionGroup XSD language capability. These instances are collected into collections within the Opra element, named as the plural form (eg. *OpraFunctionalRequest* instances are collected into an *OpraFunctionalRequests* element)

- *OpraFunctionalRequest* is the base element for functional requests.
- *OpraSubscriptionRequest* is the base element for subscription requests.
- *OpraServiceCapabilitiesRequest* is the base element for polling producers of functional service capabilities

- *OpraDiscoveryRequest* is the base element for polling producers of available data of functional services.
- *OpraFunctionalDelivery* is the base element for data delivery responding to functional requests.
- *OpraServiceCapabilitiesResponse* is the base element for functional service capabilities' responses.
- *OpraDiscoveryDelivery* is the base element for returning available data information of functional services.

12.4.1.1 Generic OpRa Payload Delivery

OpRa functional deliveries are always contained within an *OpraPayloadDelivery* element (instantiating *OpraFunctionalDelivery*), which consists of frame instances containing different subsets of data:

- planned service data (in PLANNED SERVICE FRAMES)
- actual service data (in ACTUAL SERVICE FRAMES)
- raw data and simple statistics where applicable (in RAW DATA FRAMES)
- generic key performance indicators (in KPI FRAMES)

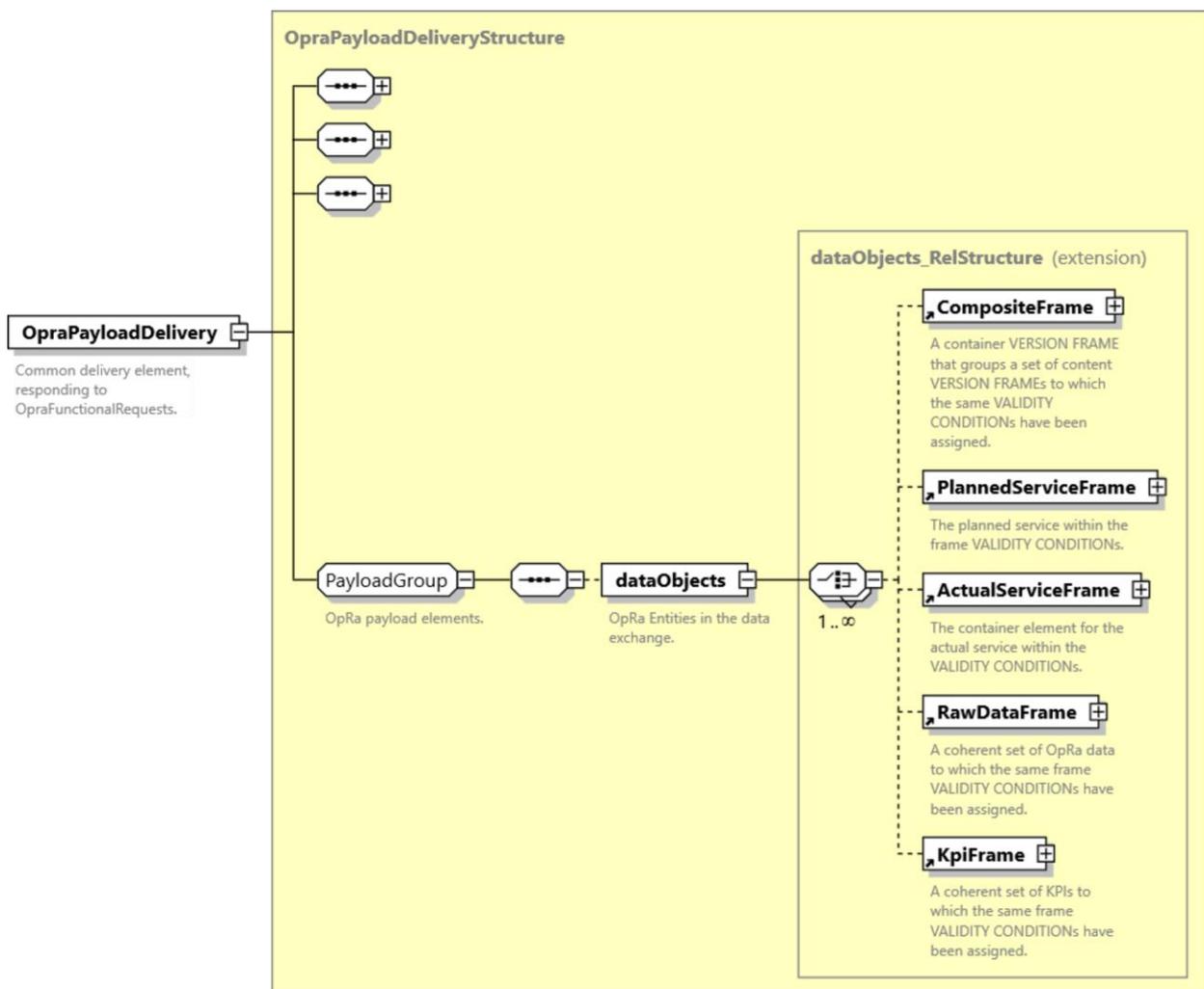


Figure 58 — OpRa payload delivery

12.5 Sample network used in use cases

To enable understanding of the nuances in each use case implemented in OpRa the XML examples are all based off the same example public transport network (see Annex B)

12.6 FRAMEs within the exchange

OpRa utilizes the common FRAME concept to select a finite subset of the PT service, limited by date, space, network, etc. The data exchange can be split to four different types of FRAME. PLANNED SERVICE FRAME and ACTUAL SERVICE FRAME, while setting up validity limitations, also serve as reference targets – network, tactical planning, etc. –for various elements ([Entity]Ref XML elements) within other frames.

12.6.1 PLANNED SERVICE FRAME

The PLANNED SERVICE FRAME is used to exchange the planned service. It contains an embedded or a URL-referred NeTEx data structure.

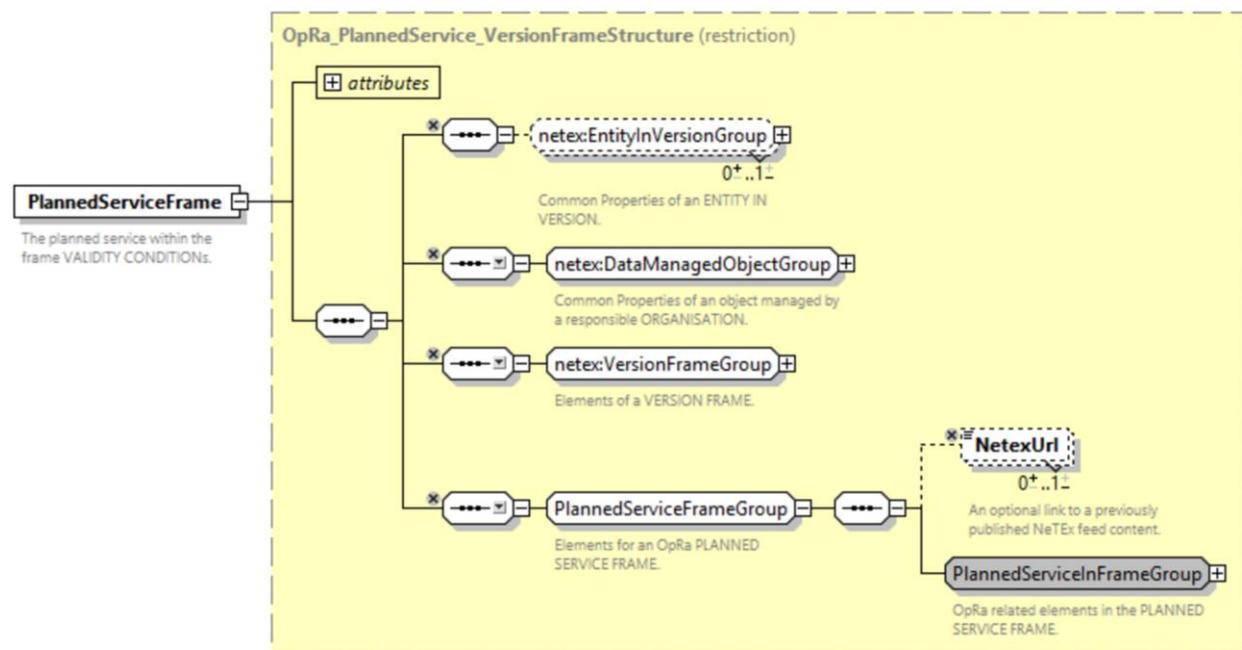


Figure 59 — Planned Service Frame

12.6.1.1 Requesting

To receive a PLANNED SERVICE FRAME, a **PlannedServiceFrameRequest** element must be included in the **OpraFunctionalRequests** collection.

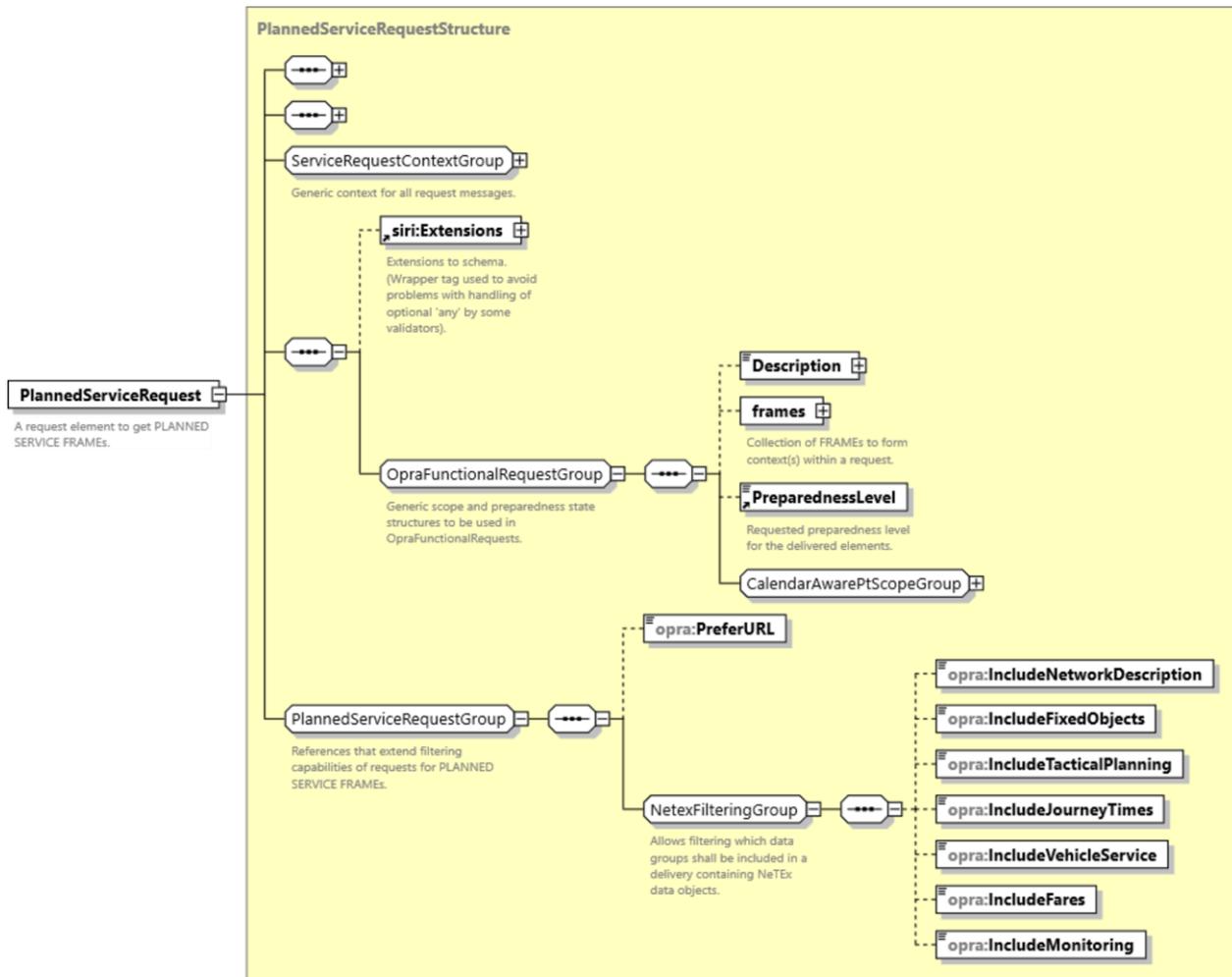


Figure 60 — Planned Service Request

Elements within *PlannedServiceFrameRequest* are used to allow filtering which data groups shall be included in a delivery:

- *PreferURL*: specifies whether the URL reference is preferred or the consumer requests for an embedded NeTEx structure.
- *IncludeNetworkDescription*
- *IncludeFixedObjects*
- *IncludeTacticalPlanning*
- *IncludeJourneyTimes*
- *IncludeVehicleService*
- *IncludeFares*
- *IncludeMonitoring*

12.6.2 ACTUAL SERVICE FRAME

The ACTUAL SERVICE FRAME is used to exchange the actual service. By comparing the PLANNED and ACTUAL SERVICE FRAMEs, the adjustments and deviations from the planned service can be determined by the consumer system.

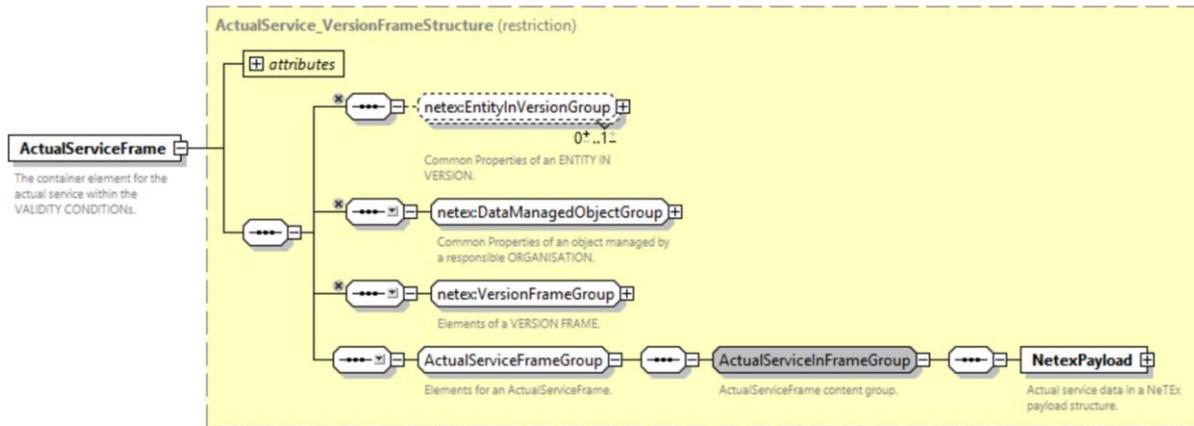


Figure 61 — Actual Service Frame

12.6.2.1 Requesting

To receive an ACTUAL SERVICE FRAME, an ActualServiceFrameRequest element must be included in the OpraFunctionalRequests collection.

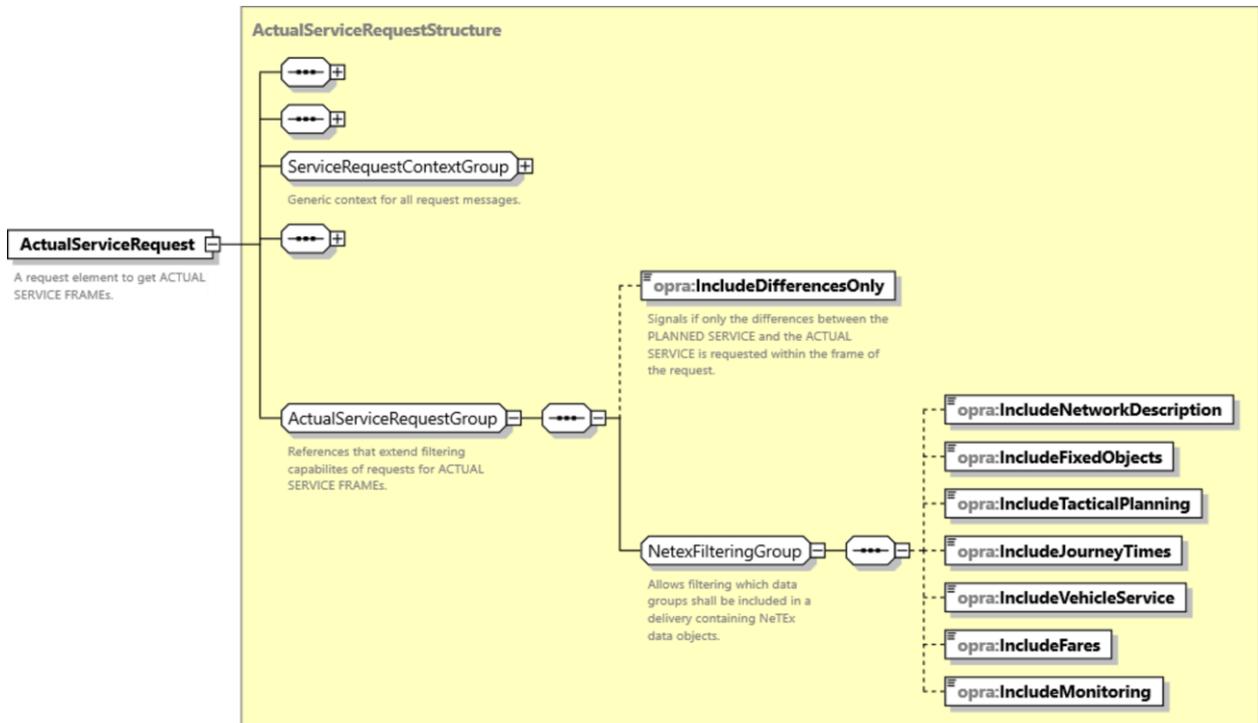


Figure 62 — Actual Service Request

The data depth within the ACTUAL SERVICE FRAME can be controlled with the same specification methods as in *PlannedServiceFrameRequest*.

12.6.3 RAW DATA FRAME

The RAW DATA FRAME is the container to exchange INDICATORS and the respective INDICATOR LOG ENTRIES if required. Each implemented INDICATOR has its own containing collection which shall be included in the RAW DATA FRAME.

If the INDICATOR LOG ENTRY is present, there can be two types of relationship between the INDICATOR and the INDICATOR LOG ENTRY:

- The INDICATOR LOG ENTRY affects multiple INDICATORS. In this case, the container of the INDICATOR LOG ENTRIES is listed separately, preceding the INDICATOR. The INDICATOR will refer to each INDICATOR LOG ENTRY through its “id” field.
- The INDICATOR LOG ENTRY is related to exactly one INDICATOR. In this case, the INDICATOR LOG ENTRY is contained within the INDICATOR (as a state or transaction recording).

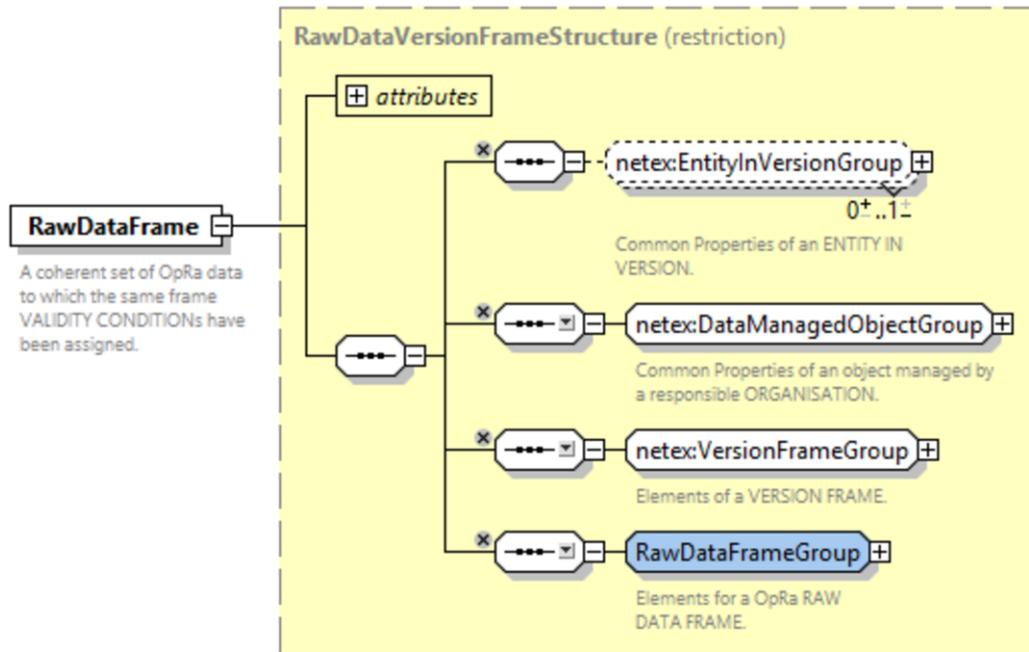


Figure 63 — Raw Data Frame

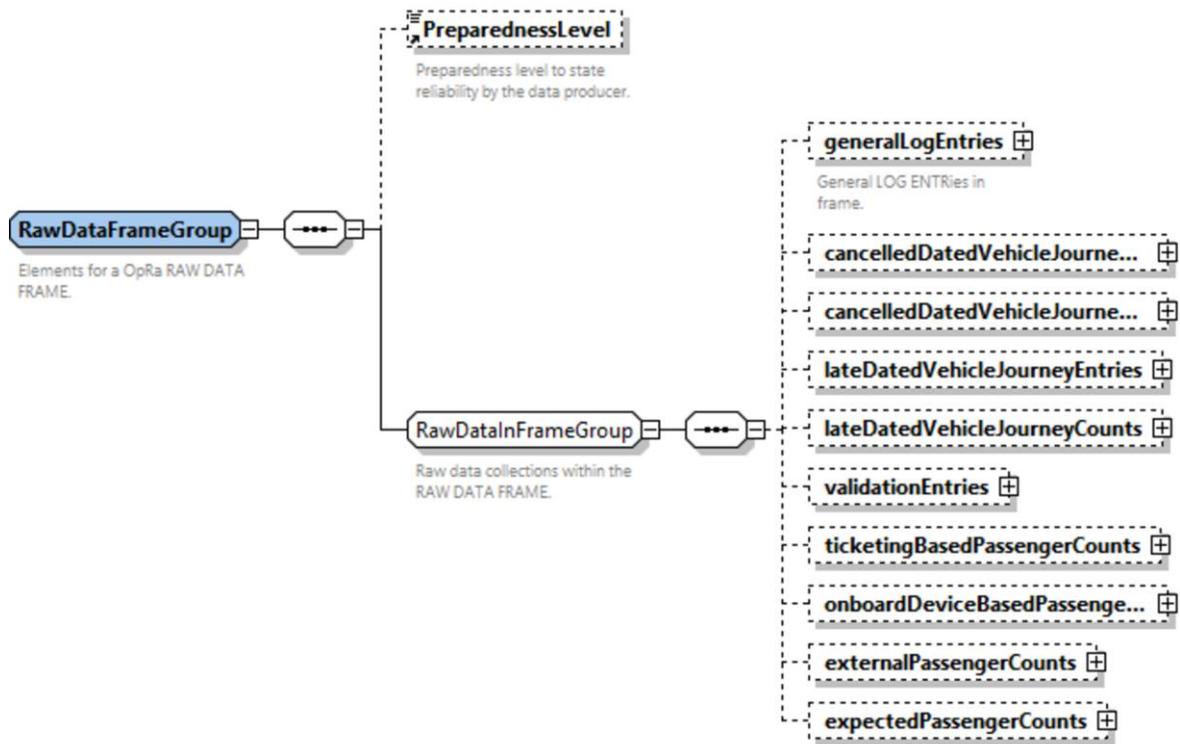


Figure 64 — Elements in a Raw Data Frame

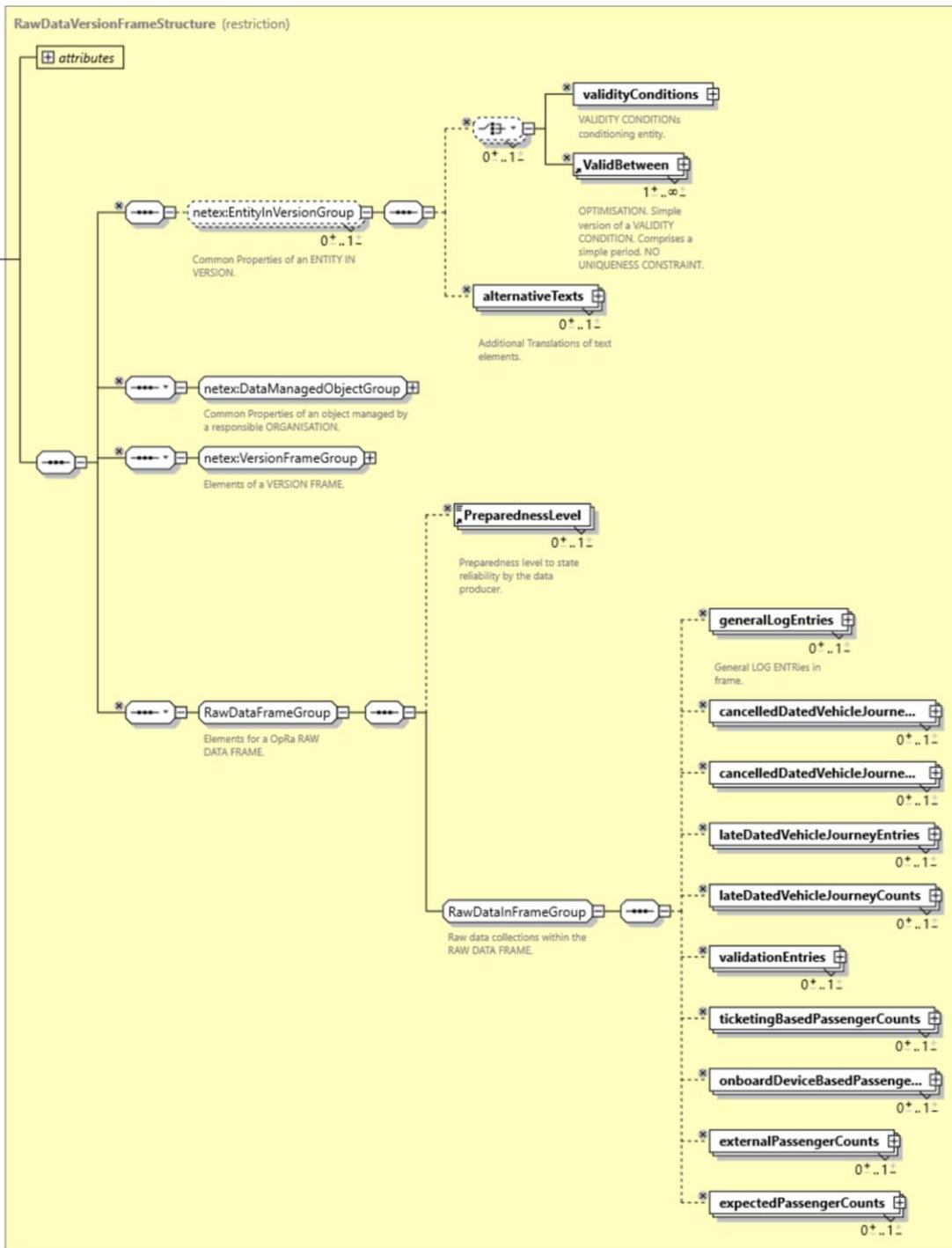


Figure 65 — Actual Service Frame

This version of the OpRa standard contains the following requests and deliveries:

Table 49 — Data elements in requests and deliveries

Use Case	Data element
INDICATOR data group within the RAW DATA FRAME	
EFFICIENCY-011: Number of late journeys	R: <i>LateDatedVehicleJourneysRequest</i>

<i>lateDatedVehicleJourneys</i>	DR: <i>LateDatedVehicleJourneysDiscoveryRequest</i> DD: <i>LateDatedVehicleJourneysDiscoveryDelivery</i>
EFFICIENCY-014: Number of cancelled journeys <i>cancelledDatedVehicleJourneys</i>	R: <i>CancelledDatedVehicleJourneysRequest</i> DR: <i>CancelledDatedVehicleJourneysDiscoveryRequest</i> DD: <i>CancelledDatedVehicleJourneysDiscoveryDelivery</i>
DEMAND-001-001: Expected number of passengers <i>expectedPassengerCounts</i>	R: <i>ExpectedPassengerCountRequest</i> DR: <i>ExpectedPassengerCountDiscoveryRequest</i> DD: <i>ExpectedPassengerCountDiscoveryDelivery</i>
DEMAND-002-001: Measured number of passengers <i>ticketingBasedPassengerCounts / onboardDeviceBasedPassengerCounts / externalPassengerCounts</i>	R: <i>TicketingBasedPassengerCountRequest / OnboardDeviceBasedPassengerCountRequest / ExternalPassengerCountRequest</i> DR: <i>TicketingBasedPassengerCountDiscoveryRequest / OnboardDeviceBasedPassengerCountDiscoveryRequest / ExternalPassengerCountDiscoveryRequest</i> DD: <i>TicketingBasedPassengerCountDiscoveryDelivery / OnboardDeviceBasedPassengerCountDiscoveryDelivery / ExternalPassengerCountDiscoveryDelivery</i>

R: Request element

DR: Discovery request element

DD: Discovery delivery element

To see details about the request and delivery elements, see the corresponding chapter about the use case.

12.6.3.1 Raw Data Frame PreparednessLevel

The PreparednessLevel element is an enumeration of

- Raw
- Cleansed
- Processed

The different preparedness levels are described in 9.1.1 Data preparedness.

12.6.4 KPI FRAME

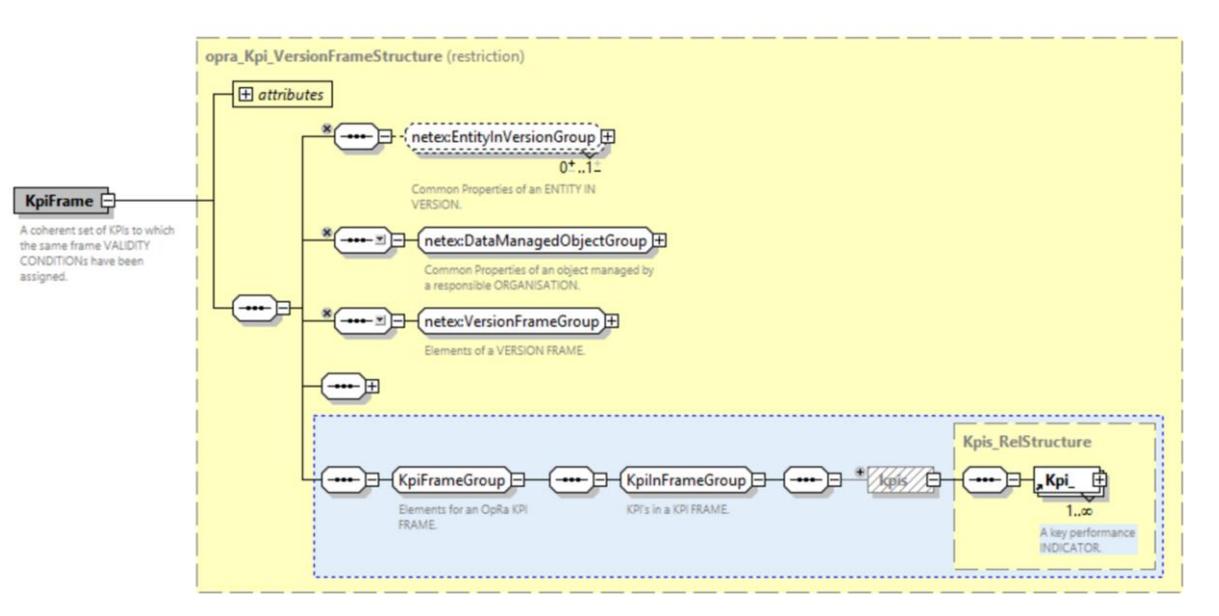


Figure 66 — KPI frame

This frame will be developed for the final version of the TS.

13 Worked Used Cases

13.1 EFFICIENCY-011: Number of late journeys

Information about delayed services can be exchanged in dedicated containers in a RAW DATA FRAME. Multiple segmentations can occur in a data exchange, so LATE DATED VEHICLE JOURNEY ENTRIES are presented separately, and LATE DATED VEHICLE JOURNEY COUNT elements can refer to these entries.

The INDICATOR role is fulfilled by the LATE DATED VEHICLE COUNT entity, which shall contain the number of late journeys within the INDICATOR’s scope. Beyond the generic CALENDAR AWARE PT SCOPE, an additional SITUATION reference (delays that happened due to that SITUATION), and a TYPE OF DELAY reference (classifying the delay, where each class can differ in consequences like refunds) can be specified.

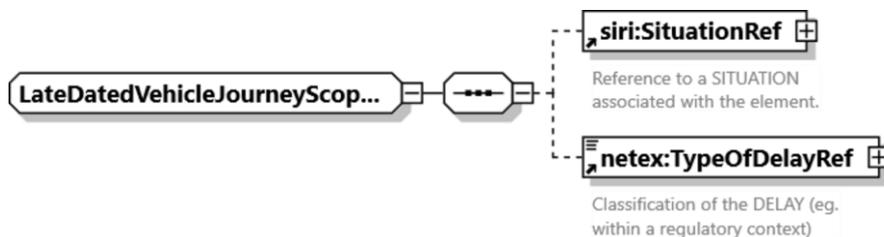


Figure 67 — LateDatedVehicleJourney

To exchange data about individual journeys, LATE DATED VEHICLE ENTRIES can be utilized. It can be used to monitor the escalation of a DELAY (by recording it for each SCHEDULED STOP POINT for a DATED VEHICLE JOURNEY), but it can also be used to register the DELAY only at the terminus. This depends on the regulatory context in which the data exchange is used.

The LATE DATED VEHICLE JOURNEY COUNT may refer to the individual entries that were used to compute the statistics within itself.

13.1.1 Discovery

To discover available data, a `LateDatedVehicleJourneyDiscoveryRequest` element must be included in the `OpraDiscoveryRequests` collection. The producer shall return a `LateDatedVehicleJourneyDiscoveryDelivery` element for this request, containing the collection of the operating days that have data for the requested discovery scope.

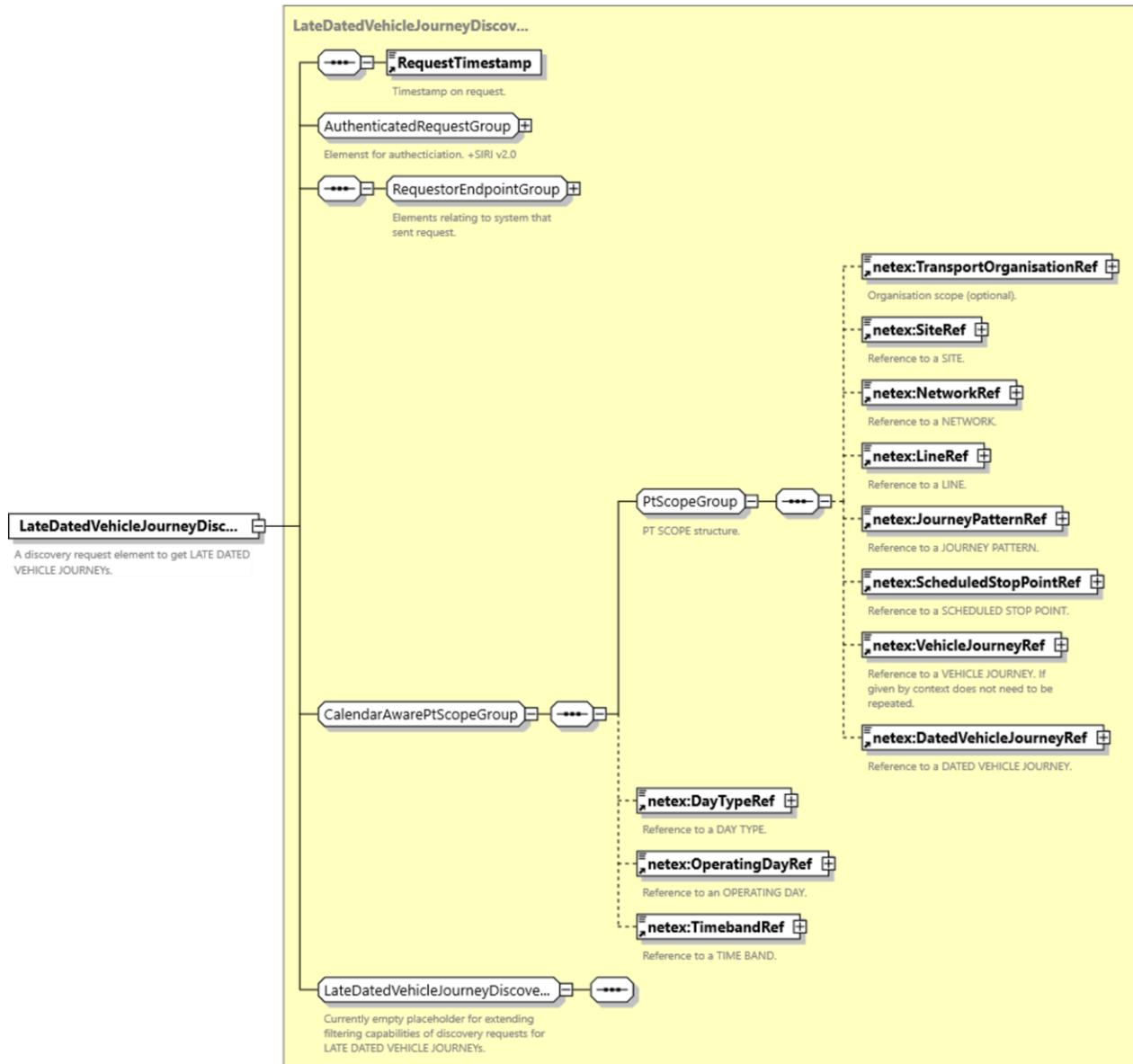


Figure 68 — LateDatedVehicleJourneyDiscoveryRequest structure

13.1.2 Requesting

To receive data about LATE DATED VEHICLE JOURNEYS, a `LateDatedVehicleJourneyRequest` element must be included in the `OpraFunctionalRequests` collection.

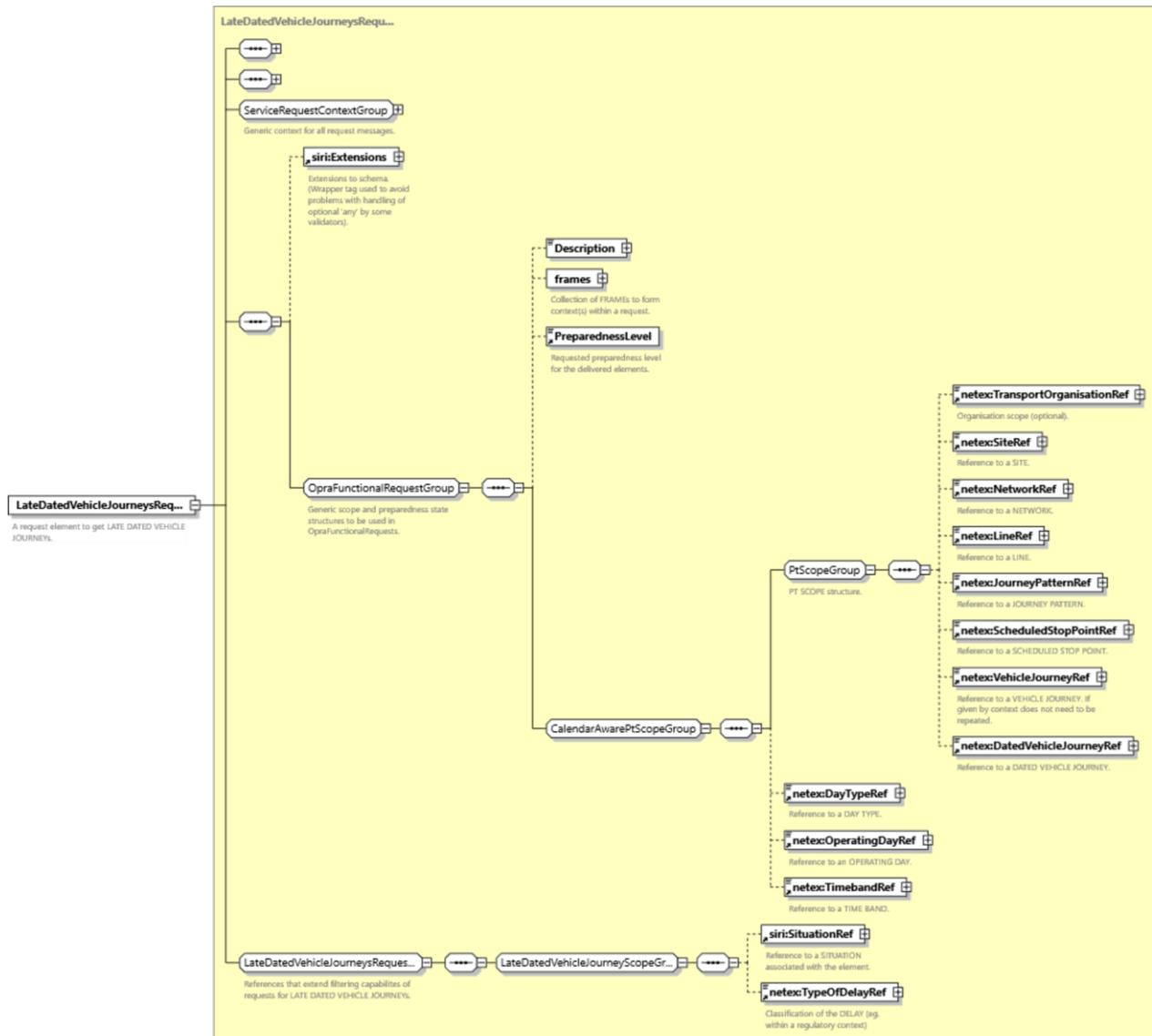


Figure 69 — LateDatedVehicleJourneyRequest structure

13.1.3 Payload

LATE DATED VEHICLE JOURNEY data group is organized under *lateDatedVehicleJourneyCounts* collection element within a RAW DATA FRAME. A *LateDatedVehicleJourneyCount* element aggregates multiple *LateDatedVehicleJourneyEntry* elements' data and contains the distribution of the delays' duration in the *intervals* collection. The entry elements are optionally presented in a separate *lateDatedVehicleJourneyEntries* collection, preceding *lateDatedVehicleJourneyCounts*.

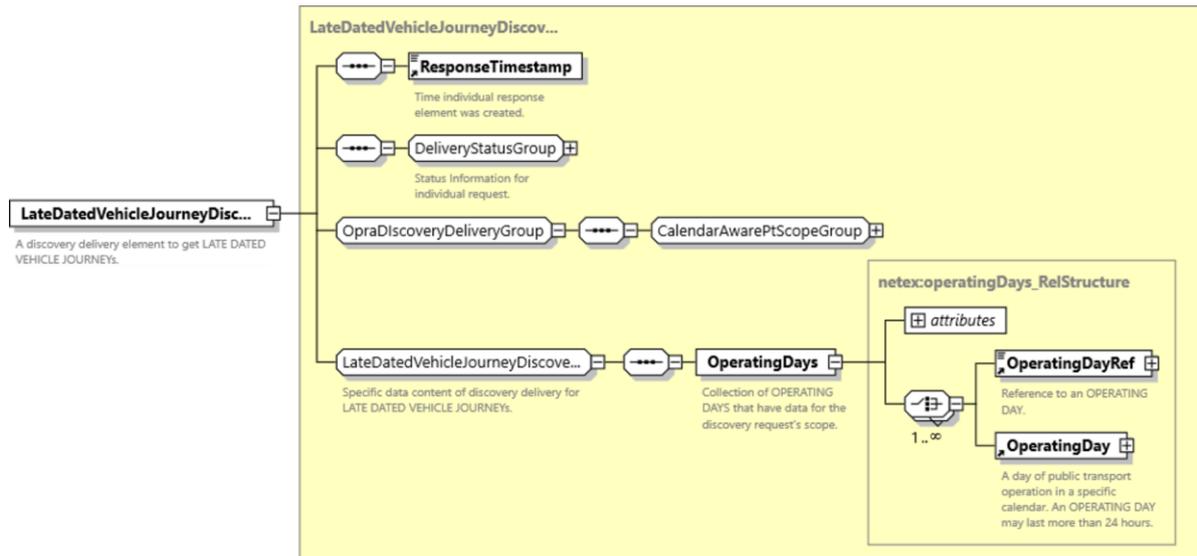


Figure 70 — LateDatedVehicleJourney delivery structure

13.1.4 Example

In this example, the sample delay statistics for the sample network (see Annex B) is represented in a RAW DATA FRAME.

```
<RawDataFrame>
  <netex:ValidBetween>
    <netex:FromDate>2025-03-16T00:00:00</netex:FromDate>
    <netex:ToDate>2025-03-18T23:59:59</netex:ToDate>
  </netex:ValidBetween>
  <PreparednessLevel>Cleansed</PreparednessLevel>
  <lateDatedVehicleJourneyEntries>
    <LateDatedVehicleJourneyEntry>
      <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
      <netex:DatedVehicleJourneyRef ref="Test:DVJ:20250316_0700"/>
      <DelayDuration>PT5M</DelayDuration>
    </LateDatedVehicleJourneyEntry>
    <LateDatedVehicleJourneyEntry>
      <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
      <netex:DatedVehicleJourneyRef ref="Test:DVJ:20250316_1330"/>
      <DelayDuration>PT5M</DelayDuration>
    </LateDatedVehicleJourneyEntry>
    <LateDatedVehicleJourneyEntry>
      <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
      <netex:DatedVehicleJourneyRef ref="Test:DVJ:20250317_0730"/>
      <DelayDuration>PT5M</DelayDuration>
    </LateDatedVehicleJourneyEntry>
    <LateDatedVehicleJourneyEntry>
      <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
      <netex:DatedVehicleJourneyRef ref="Test:DVJ:20250317_1300"/>
      <DelayDuration>PT10M</DelayDuration>
    </LateDatedVehicleJourneyEntry>
    <LateDatedVehicleJourneyEntry>
      <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
      <netex:DatedVehicleJourneyRef ref="Test:DVJ:20250318_1800"/>
      <DelayDuration>PT10M</DelayDuration>
    </LateDatedVehicleJourneyEntry>
  </lateDatedVehicleJourneyEntries>
</RawDataFrame>
```

```

</lateDatedVehicleJourneyEntries>
<lateDatedVehicleJourneyCounts>
  <LateDatedVehicleJourneyCount>
    <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
    <intervals>
      <DurationInterval>
        <!-- statistics for delays less than 5 (excl.) minutes -
->
          <EndOfInterval>PT5M</EndOfInterval>
          <EndIncluded>>false</EndIncluded>
          <NumberOfOccurrences>69</NumberOfOccurrences>

        <PercentageOfOccurrences>0.9324</PercentageOfOccurrences>
        </DurationInterval>
        <DurationInterval>
          <!-- statistics for delays between 5 (incl.) and 10
(excl.) minutes -->
          <StartOfInterval>PT5M</StartOfInterval>
          <StartIncluded>>true</StartIncluded>
          <EndOfInterval>PT10M</EndOfInterval>
          <EndIncluded>>false</EndIncluded>
          <NumberOfOccurrences>3</NumberOfOccurrences>

        <PercentageOfOccurrences>0.0405</PercentageOfOccurrences>
        </DurationInterval>
        <DurationInterval>
          <!-- statistics for delays of length at least 10 (incl.)
minutes -->
          <StartOfInterval>PT10M</StartOfInterval>
          <StartIncluded>>true</StartIncluded>
          <NumberOfOccurrences>2</NumberOfOccurrences>

        <PercentageOfOccurrences>0.0270</PercentageOfOccurrences>
        </DurationInterval>
      </intervals>
    </LateDatedVehicleJourneyCount>
  </lateDatedVehicleJourneyCounts>
</RawDataFrame>

```

13.2 EFFICIENCY-014: Number of cancelled journeys

Information about cancelled services can be exchanged in dedicated containers.

The INDICATOR role is fulfilled by the CANCELLED DATED VEHICLE COUNT entity, which shall contain the number of cancelled journeys within the INDICATOR's scope. Beyond the generic CALENDAR AWARE PT SCOPE, an additional SITUATION reference (delays that happened due to that SITUATION), and a CONTROL ACTION reference (the corresponding operational action) can be specified.

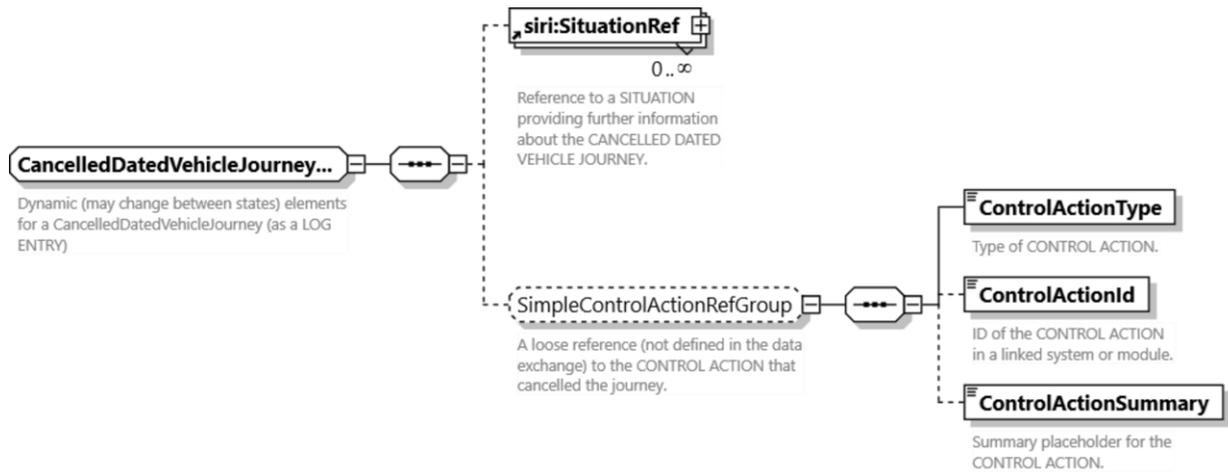


Figure 71 — CancelledDatedVehicleJourney

13.2.1 Discovery

To discover available data, a *CancelledDatedVehicleJourneyDiscoveryRequest* element must be included in the *OpraDiscoveryRequests* collection.

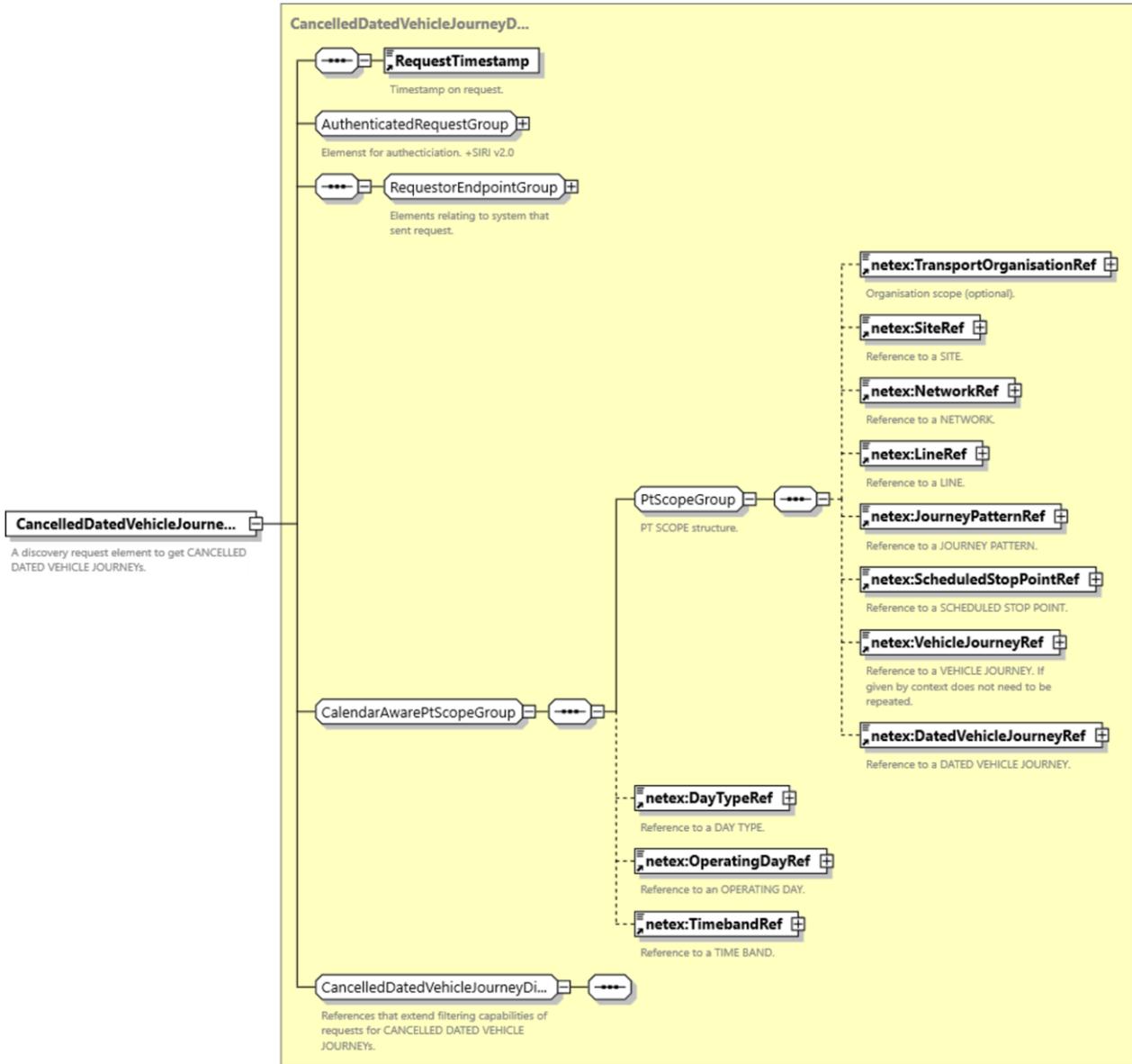


Figure 72 — CancelledDatedVehicleJourneyDiscoveryRequest structure

The producer shall return a *CancelledDatedVehicleJourneyDiscoveryDelivery* element for this request, containing the collection of the operating days that have cancellation data for the requested discovery scope.

13.2.2 Requesting

To receive data about CANCELLED DATED VEHICLE JOURNEYS, a *CancelledDatedVehicleJourneyRequest* element must be included in the *OpraFunctionalRequests* collection.

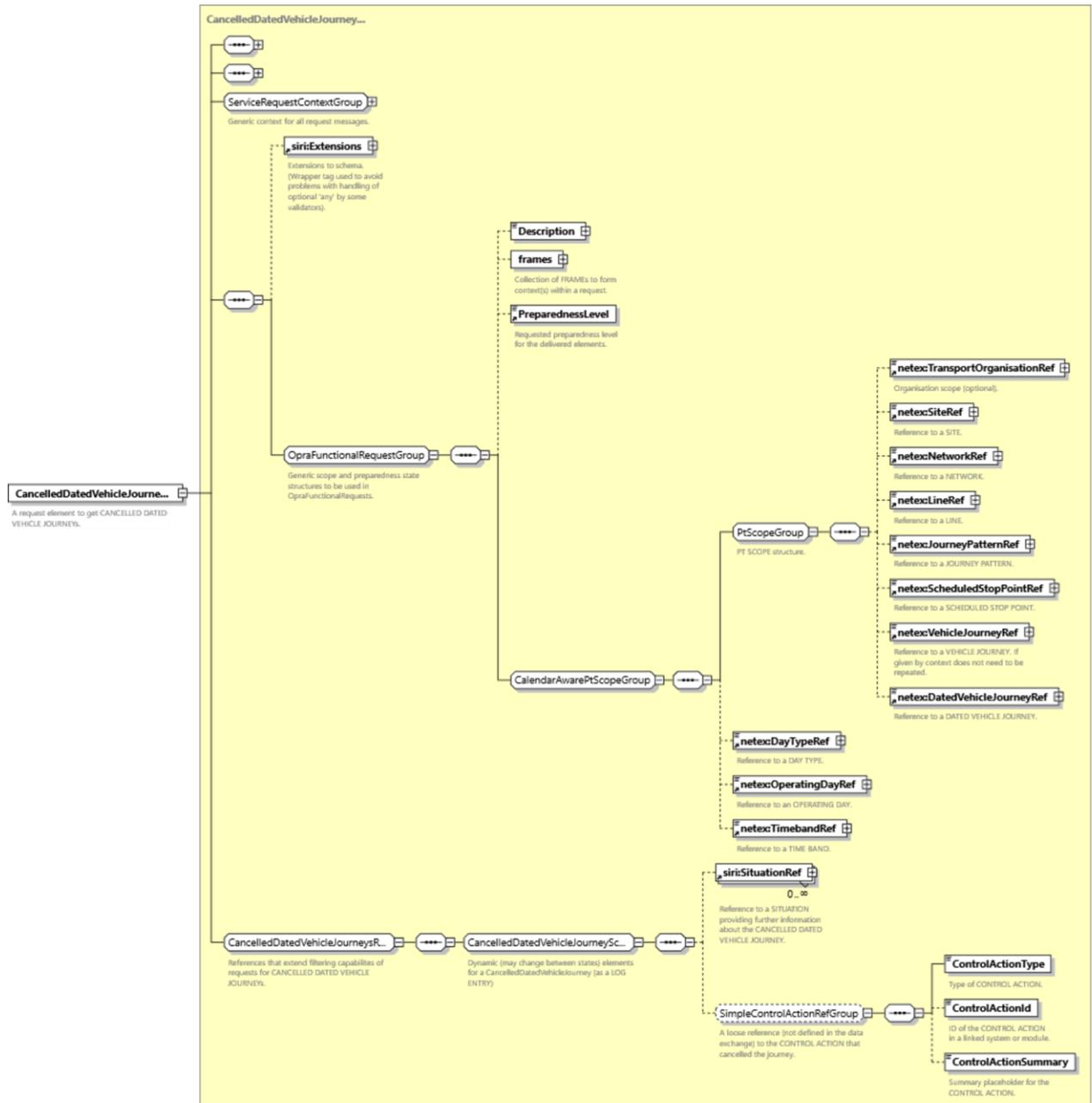


Figure 73 — CancelledDatedVehicleJourneyRequest structure

13.2.3 Payload

CANCELLED DATED VEHICLE JOURNEY data group is organized under *cancelledDatedVehicleJourneyCounts* collection element within a RAW DATA FRAME. A *CancelledDatedVehicleJourneyCount* element aggregates multiple *CancelledDatedVehicleJourneyEntry* elements' data, containing how many cancellations happened within its scope. The individual entry elements are optionally presented in a separate *cancelledDatedVehicleJourneyEntries* collection, preceding *cancelledDatedVehicleJourneyCounts*.

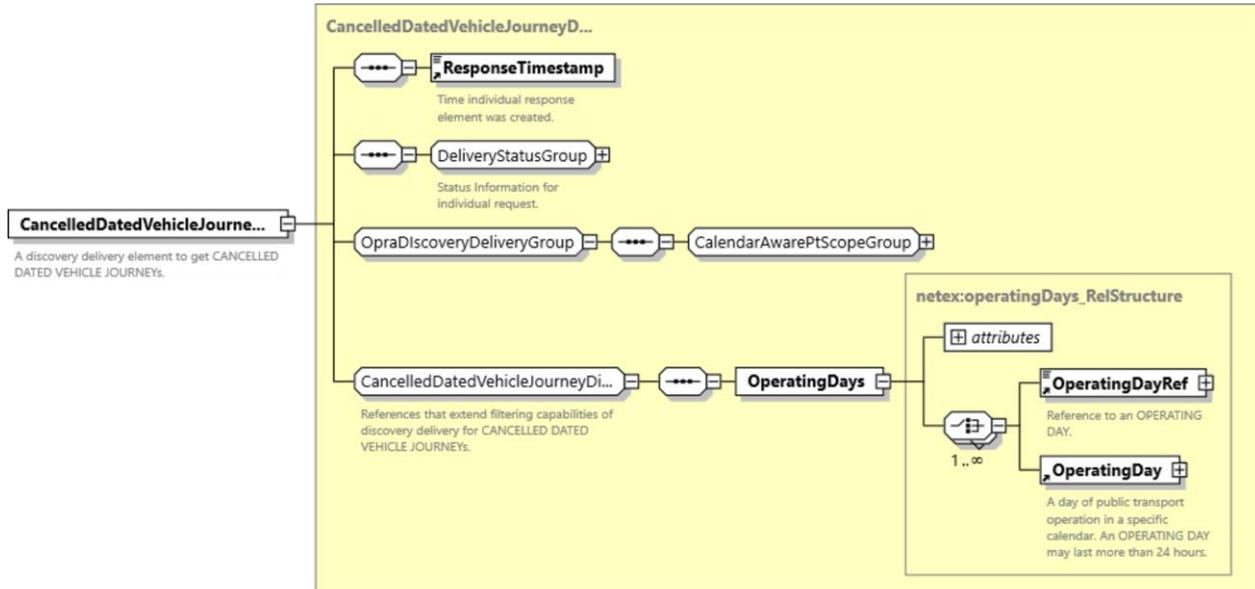


Figure 74 — CancelledDatedVehicleJourneyDiscoveryDelivery structure

13.2.4 Example

```

<RawDataFrame>
  <netex:ValidBetween>
    <netex:FromDate>2025-03-16T00:00:00</netex:FromDate>
    <netex:ToDate>2025-03-18T23:59:59</netex:ToDate>
  </netex:ValidBetween>
  <PreparednessLevel>Cleansed</PreparednessLevel>
  <cancelledDatedVehicleJourneyEntries>
    <CancelledDatedVehicleJourneyEntry>
      <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
      <netex:DatedVehicleJourneyRef ref="Test:DVJ:20250317_1700"/>
      <ReplacementJourneyCreated>>true</ReplacementJourneyCreated>
    </CancelledDatedVehicleJourneyEntry>
    <CancelledDatedVehicleJourneyEntry>
      <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
      <netex:DatedVehicleJourneyRef ref="Test:DVJ:20250318_0630"/>
      <ReplacementJourneyCreated>>false</ReplacementJourneyCreated>
    </CancelledDatedVehicleJourneyEntry>
  </cancelledDatedVehicleJourneyEntries>
  <cancelledDatedVehicleJourneyCounts>
    <CancelledDatedVehicleJourneyCount>
      <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
      <NumberOfOccurrences>2</NumberOfOccurrences>
      <PercentageOfOccurrences>0.0270</PercentageOfOccurrences>
    </CancelledDatedVehicleJourneyCount>
  </cancelledDatedVehicleJourneyCounts>
</RawDataFrame>

```

13.3 Vehicle load related use cases

In OpRa, use cases that cover vehicle loads (not only passengers but also wheelchairs, bikes) are using the implementation of VEHICLE LOAD ENTRY by including *OccupancyScopeFilter* and *OccupancyValues* structures from the SIRI standard.

OccupancyScopeFilter allows further segmenting the vehicle load data, and is added after the indicator-level scopes for each specialised indicator. It provides capabilities for TRAINS, fare structure filtering, and categorising passengers.

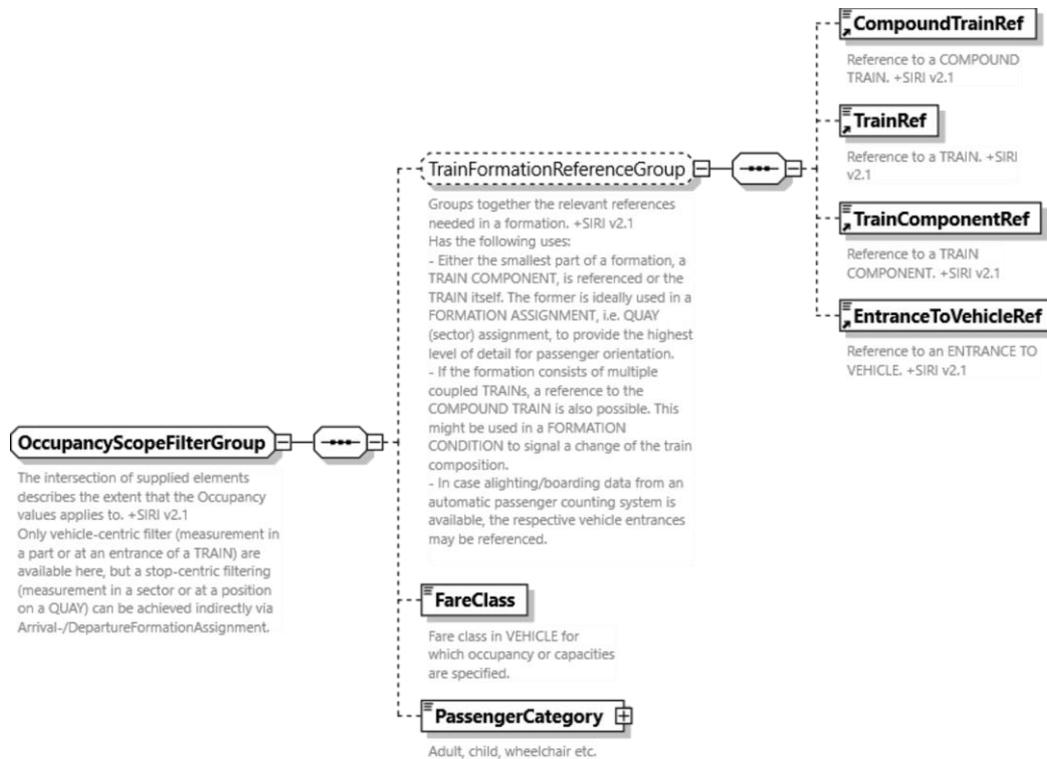


Figure 75 — OccupancyScopeFilter

OccupancyValues allows adding crowdedness classifications (can be used in “spot effect” passenger counting, or “levels of occupancy” displays in train compartments), occupancy percentage, alighting and boarding, onboard counts, and occupancy elements for special capacities.

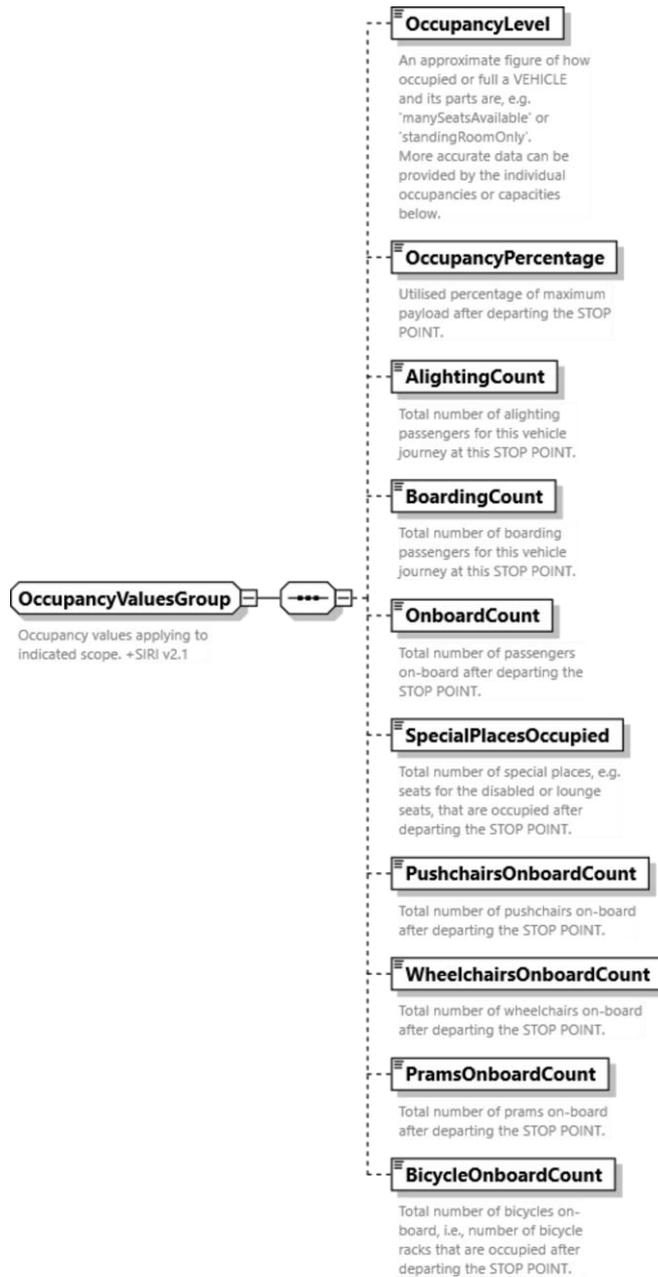


Figure 76 — OccupancyValues

When dealing with statistical probability and confidence, a maximum and minimum VEHICLE LOAD ENTRY is also presentable to the consumer to cover fluctuation within the use of public transport services and provide available indicators for further calculations.

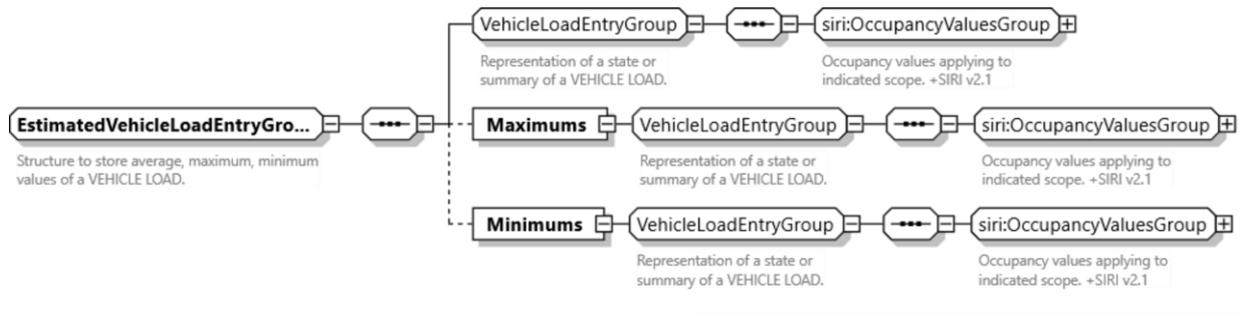


Figure 77 — EstimatedVehicleLoadEntry

13.3.1 DEMAND-001-001: Expected number of passengers

EXPECTED PASSENGER COUNT is a computed passenger count forecast, depending on earlier measurements within the public transport system (ticketing based or onboard device based), demand analysis, or various forecasting methods. Depending on the depth of the source data, EXPECTED PASSENGER COUNT can be filtered to refer to elements of the network, a specified time band, a certain day type, and so on.

EXPECTED PASSENGER COUNT has no additional scope defined, though placeholder elements have been defined in the schema.

13.3.1.1 Discovery

To discover available data, an *ExpectedPassengerCountDiscoveryRequest* element must be included in the *OpraDiscoveryRequests* collection. The producer shall return a *ExpectedPassengerCountDiscoveryDelivery* element for this request, containing the collection of the operating days that have EXPECTED PASSENGER COUNT data for the requested discovery scope.

13.3.1.2 Requesting

To receive data about EXPECTED PASSENGER COUNT, an *ExpectedPassengerCountRequest* element must be included in the *OpraFunctionalRequests* collection.

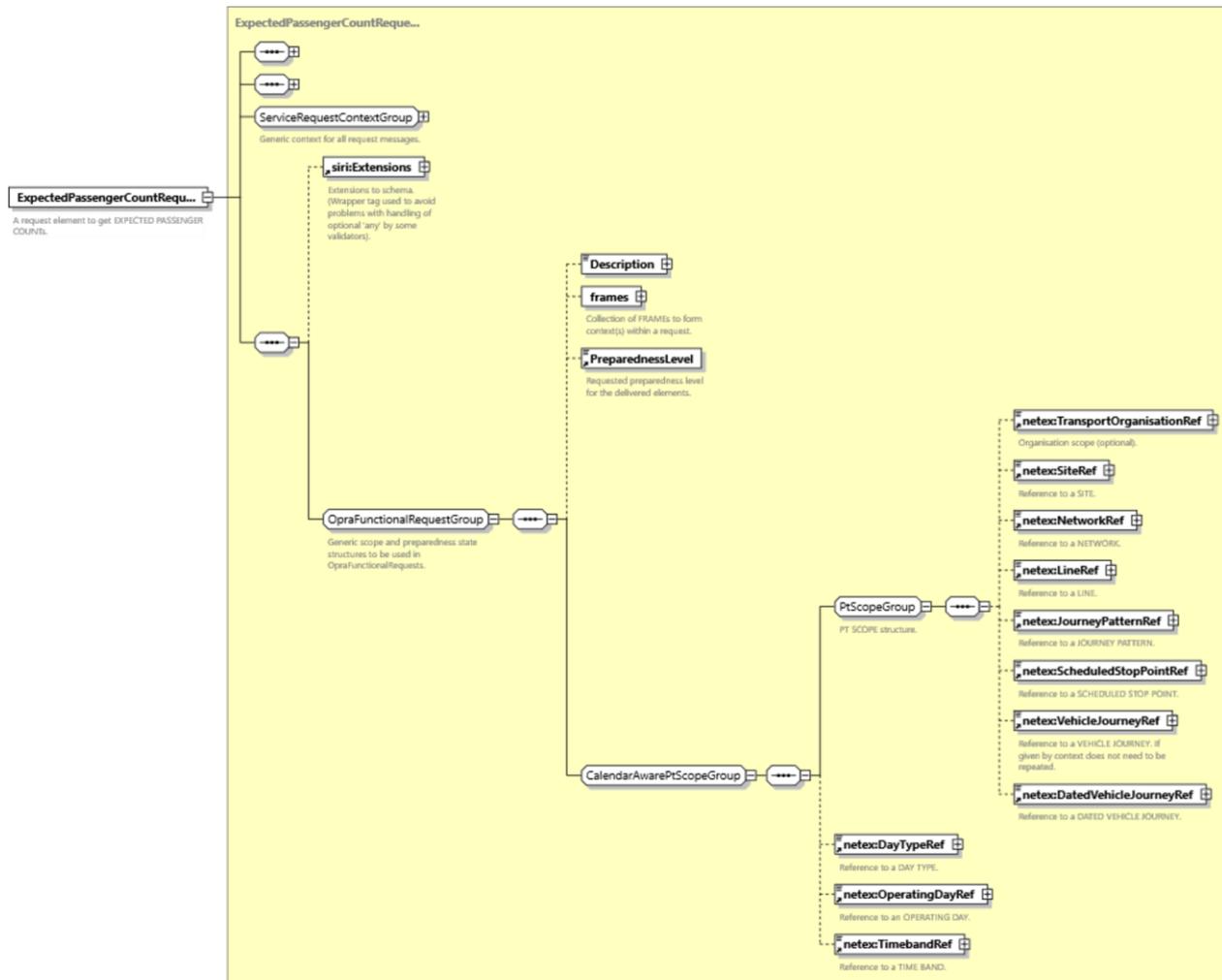


Figure 78 — ExpectedPassengerCountRequest structure

13.3.1.3 Payload

EXPECTED PASSENGER COUNT data group is organized under *expectedPassengerCounts* collection element within a RAW DATA FRAME. An *ExpectedPassengerCount* element contains passenger count information within its scope.

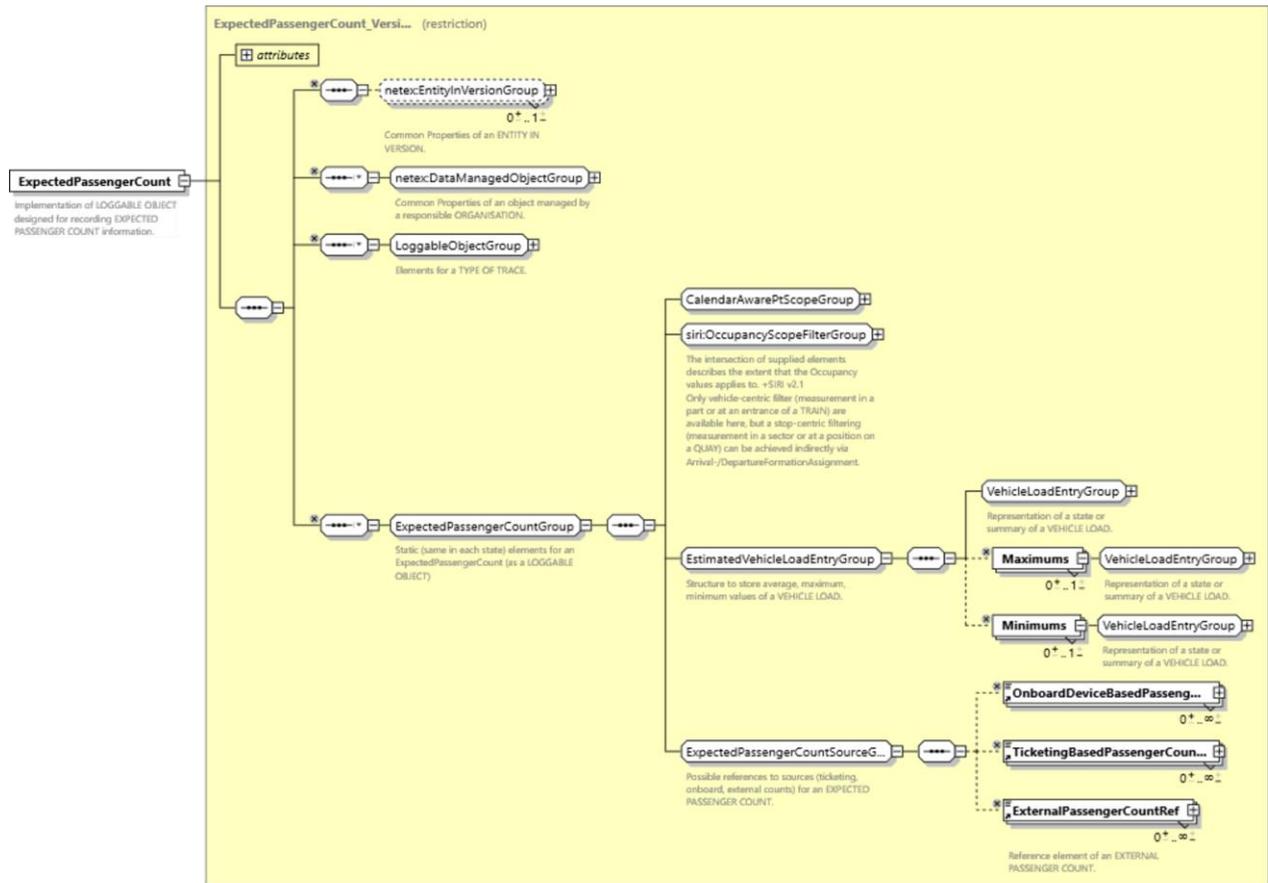


Figure 79 — ExpectedPassengerCount delivery structure

ExpectedPassengerCount element allows minimums and maximums to be optionally used for probabilities and also provides optional references to sources that contributed to the calculations covered.

13.3.1.4 Example

```
<RawDataFrame>
  <PreparednessLevel>Raw</PreparednessLevel>
  <expectedPassengerCounts>
<!-- Listing expected passenger counts at SCHEDULED STOP POINTs within a JOURNEY PATTERN:
TIME BAND 0-6 -->
  <ExpectedPassengerCount>
    <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
    <netex:ScheduledStopPointRef ref="Test:SSP:T1"/>
    <netex:TimebandRef ref="Test:TimeBand:0-6"/>
    <siri:BoardingCount>25</siri:BoardingCount>
    <siri:OnboardCount>25</siri:OnboardCount>
  </ExpectedPassengerCount>
  <ExpectedPassengerCount>
    <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
    <netex:ScheduledStopPointRef ref="Test:SSP:1a"/>
    <netex:TimebandRef ref="Test:TimeBand:0-6"/>
    <siri:AlightingCount>5</siri:AlightingCount>
    <siri:OnboardCount>20</siri:OnboardCount>
  </ExpectedPassengerCount>
  <ExpectedPassengerCount>
    <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
```

```

        <netex:ScheduledStopPointRef ref="Test:SSP:T3a"/>
        <netex:TimebandRef ref="Test:TimeBand:0-6"/>
        <siri:AlightingCount>20</siri:AlightingCount>
        <siri:OnboardCount>0</siri:OnboardCount>
    </ExpectedPassengerCount>
    <!-- A total row for TIME BAND 0-6 in the JOURNEY PATTERN - no SSP tag is
present -->
    <ExpectedPassengerCount>
        <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
        <netex:TimebandRef ref="Test:TimeBand:0-6"/>
        <siri:AlightingCount>25</siri:AlightingCount>
        <siri:BoardingCount>25</siri:BoardingCount>
    </ExpectedPassengerCount>
    <!-- TIME BAND 6-9 -->
    <ExpectedPassengerCount>
        <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
        <netex:ScheduledStopPointRef ref="Test:SSP:T1"/>
        <netex:TimebandRef ref="Test:TimeBand:6-9"/>
        <siri:BoardingCount>200</siri:BoardingCount>
        <siri:OnboardCount>200</siri:OnboardCount>
    </ExpectedPassengerCount>
    <ExpectedPassengerCount>
        <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
        <netex:ScheduledStopPointRef ref="Test:SSP:1a"/>
        <netex:TimebandRef ref="Test:TimeBand:6-9"/>
        <siri:AlightingCount>10</siri:AlightingCount>
        <siri:BoardingCount>10</siri:BoardingCount>
        <siri:OnboardCount>200</siri:OnboardCount>
    </ExpectedPassengerCount>
    <ExpectedPassengerCount>
        <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
        <netex:ScheduledStopPointRef ref="Test:SSP:2"/>
        <netex:TimebandRef ref="Test:TimeBand:6-9"/>
        <siri:AlightingCount>30</siri:AlightingCount>
        <siri:BoardingCount>30</siri:BoardingCount>
        <siri:OnboardCount>200</siri:OnboardCount>
    </ExpectedPassengerCount>
    <ExpectedPassengerCount>
        <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
        <netex:ScheduledStopPointRef ref="Test:SSP:3"/>
        <netex:TimebandRef ref="Test:TimeBand:6-9"/>
        <siri:AlightingCount>10</siri:AlightingCount>
        <siri:BoardingCount>10</siri:BoardingCount>
        <siri:OnboardCount>200</siri:OnboardCount>
    </ExpectedPassengerCount>
    <ExpectedPassengerCount>
        <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
        <netex:ScheduledStopPointRef ref="Test:SSP:4"/>
        <netex:TimebandRef ref="Test:TimeBand:6-9"/>
        <siri:AlightingCount>10</siri:AlightingCount>
        <siri:BoardingCount>10</siri:BoardingCount>
        <siri:OnboardCount>200</siri:OnboardCount>
    </ExpectedPassengerCount>
    <ExpectedPassengerCount>
        <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
        <netex:ScheduledStopPointRef ref="Test:SSP:T3a"/>
        <netex:TimebandRef ref="Test:TimeBand:6-9"/>
        <siri:AlightingCount>200</siri:AlightingCount>
        <siri:OnboardCount>0</siri:OnboardCount>
    </ExpectedPassengerCount>
    <!-- Total for TIME BAND 6-9 -->
    <ExpectedPassengerCount>

```

```

        <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
        <netex:TimebandRef ref="Test:TimeBand:6-9"/>
        <siri:AlightingCount>260</siri:AlightingCount>
        <siri:BoardingCount>260</siri:BoardingCount>
    </ExpectedPassengerCount>
    <!-- etc.. -->
    <!-- Total for JOURNEY PATTERN - no TimeBandRef tag is present -->
    <ExpectedPassengerCount>
        <netex:JourneyPatternRef ref="Test:JP:10:T1-T3a"/>
        <siri:AlightingCount>820</siri:AlightingCount>
        <siri:BoardingCount>820</siri:BoardingCount>
    </ExpectedPassengerCount>
</expectedPassengerCounts>
</RawDataFrame>

```

13.3.2 DEMAND-002-001: Measured number of passengers

The concept of “measured number of passengers” is classified into three methods:

- TICKETING BASED PASSENGER COUNT refers to data collection via the ticketing system, aggregating individual VALIDATION ENTRIES which are registered at TICKET VALIDATOR EQUIPMENTS. Ticketing-related additional scope introduces optional filtering and classification capabilities for PASSENGER TRAVEL ENTRY type, the corresponding TICKET VALIDATOR EQUIPMENT, and the FARE PRODUCT.

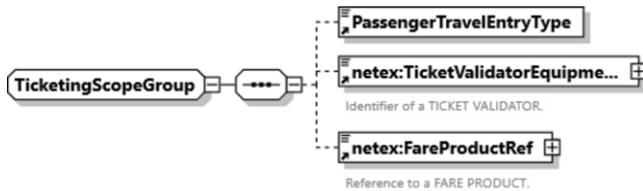


Figure 80 — TicketingScope for ticketing based passenger counts

- ONBOARD DEVICE BASED PASSENGER COUNT refers to data collection via onboard passenger counting ENTRANCE SENSORS at PASSENGER ENTRANCES, but may also include manual recording of passenger movement. Onboard device related additional scope introduces optional filtering classification capabilities for a (physical) VEHICLE, a PASSENGER SPACE (decks, compartments, etc.), a PASSENGER ENTRANCE (doors), and an ENTRANCE SENSOR (the physical sensor equipment that records the passenger movement)

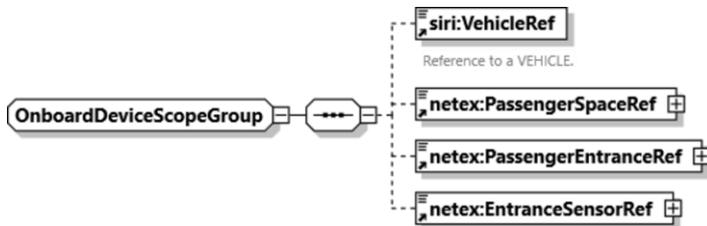


Figure 81 — OnboardDevicesScope for sensor based passenger counts

- EXTERNAL PASSENGER COUNT refers to passenger volume data from a data source outside the PT system’s boundaries, such as demand research, plant operation planning, etc. Receiving information

about this category can provide further demand context for planning PT services and operations. No additional scope was defined for this data group.

A consumer might specify any number of the above categories for requested data by adding multiple requests to the *OpraFunctionalRequests* collection. Multiple requests of the same category can also be included, practically with different refined scopes (such as a line or site of interest) to optimise the workload on the producer side.

13.3.2.1 Discovery

To discover available data, a *TicketingBasedPassengerCountDiscoveryRequest* / *OnboardDeviceBasedPassengerCountDiscoveryRequest* / *ExternalPassengerCountDiscoveryRequest* element must be included in the *OpraDiscoveryRequests* collection. The producer shall return a *TicketingBasedPassengerCountDiscoveryDelivery* / *OnboardDeviceBasedPassengerCountDiscoveryDelivery* / *ExternalPassengerCountDiscoveryDelivery* element for the request, respectively, containing the collection of the operating days that have corresponding data for the requested discovery scope.

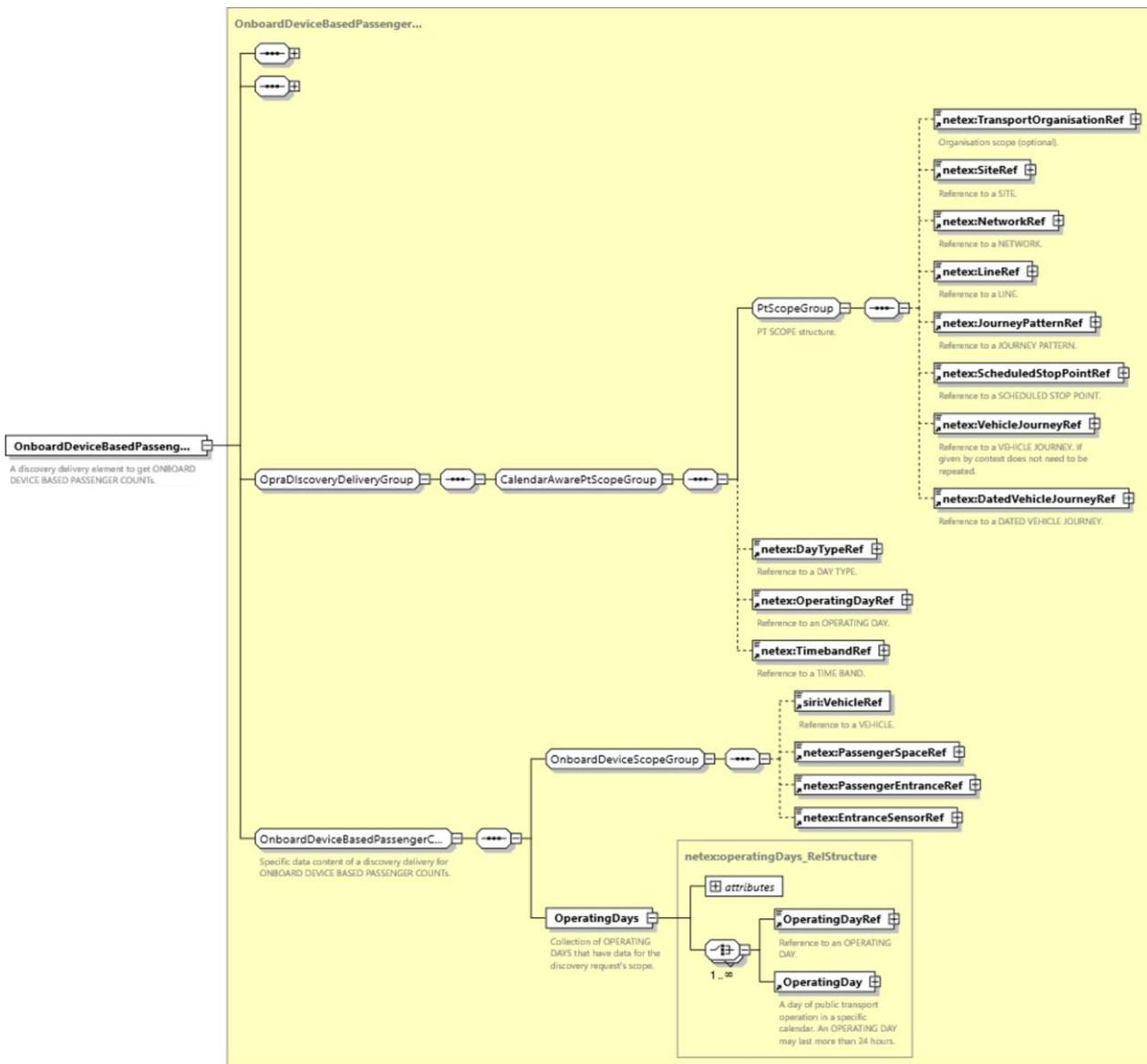


Figure 82 — Onboard Device Based Passenger Count Discovery Delivery structure

The figure above illustrates the returned collection of OPERATING DAYS having existing data for the requested scope

13.3.2.2 Requesting

To receive data about TICKETING BASED PASSENGER COUNT, an *TicketingBasedPassengerCountRequest* element must be included in the *OpraFunctionalRequests* collection.

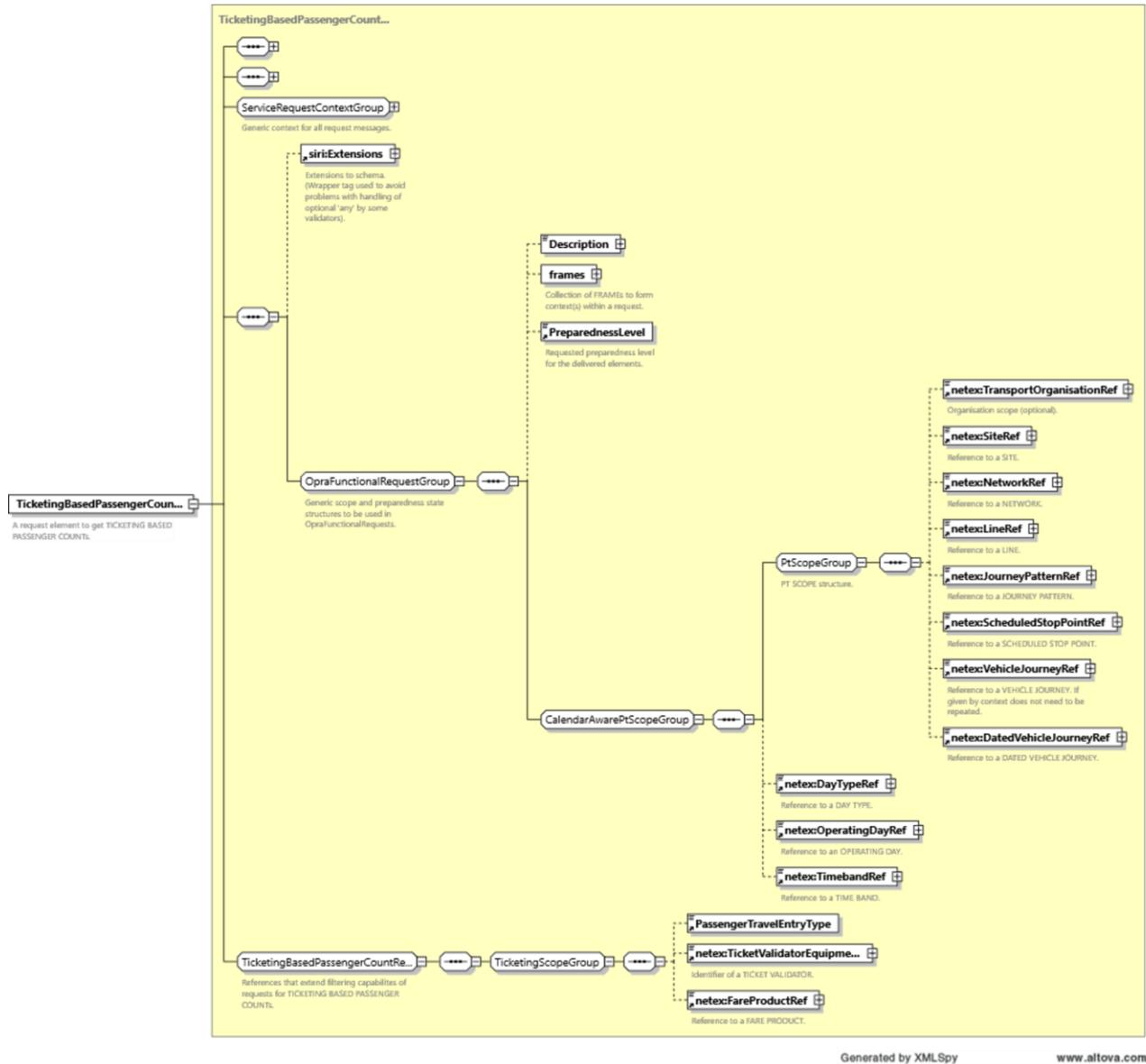


Figure 83 — Ticketing based passenger count request structure

To receive data about ONBOARD DEVICE BASED PASSENGER COUNT, an *OnboardDeviceBasedPassengerCountRequest* element must be included in the *OpraFunctionalRequests* collection.

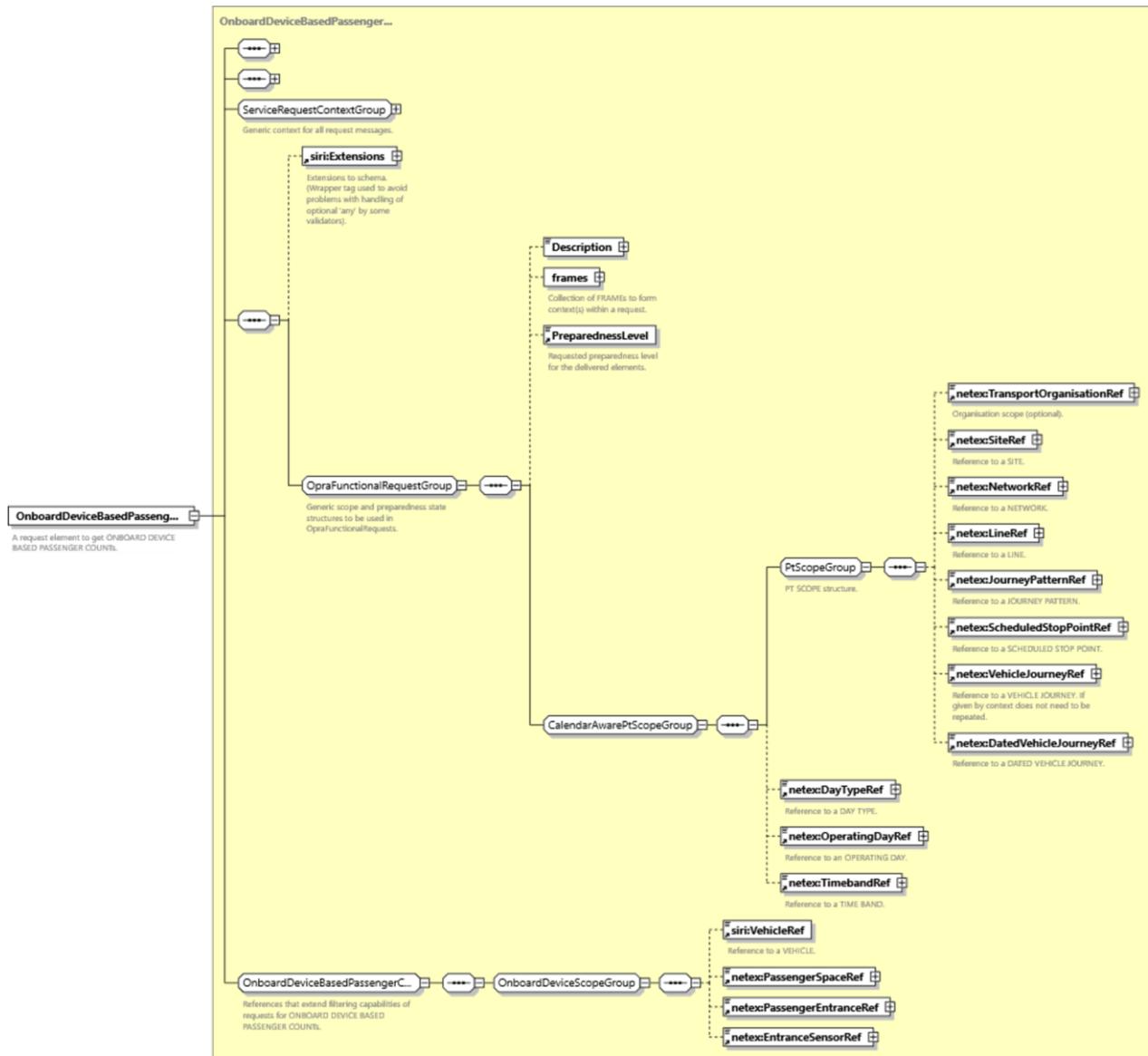


Figure 84 — Onboard device based passenger count request structure

To receive data about EXTERNAL BASED PASSENGER COUNT, an *ExternalPassengerCountRequest* element must be included in the *OpraFunctionalRequests* collection.

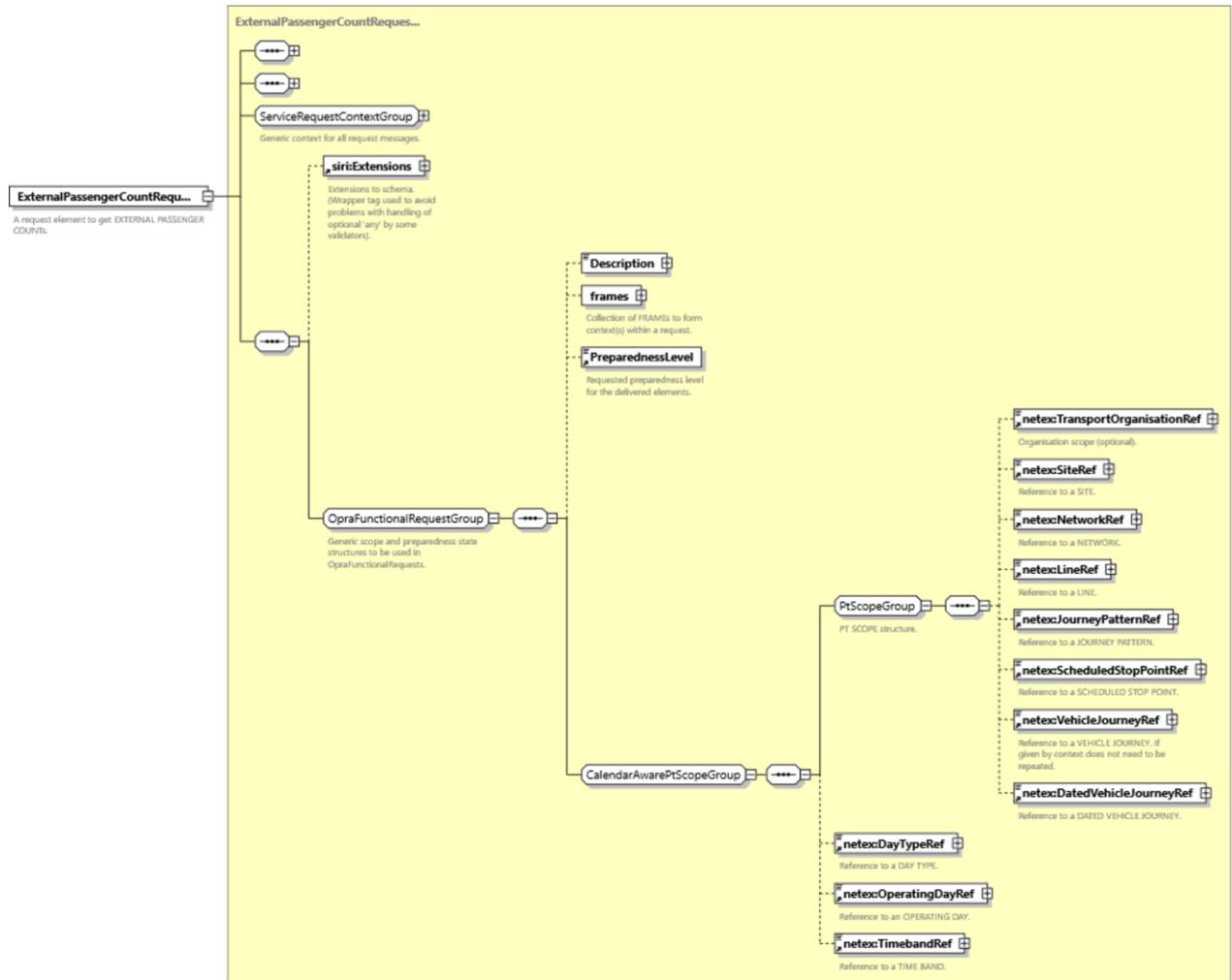


Figure 85 — External passenger count request structure

Multiple requests for multiple sources can be composed into one bulk request, even for disjunct scopes (e.g. all ticketing and onboard data from multiple sites within a network)

13.3.2.3 Payload

TICKETING BASED PASSENGER COUNT data group is organized under *ticketingBasedPassengerCounts* collection element within a RAW DATA FRAME. A *TicketingBasedPassengerCount* element aggregates multiple *ValidationEntry* elements' data and refers to the entries within its scope. The entry elements are optionally presented in a separate *validationEntries* collection, preceding *ticketingBasedPassengerCounts*.

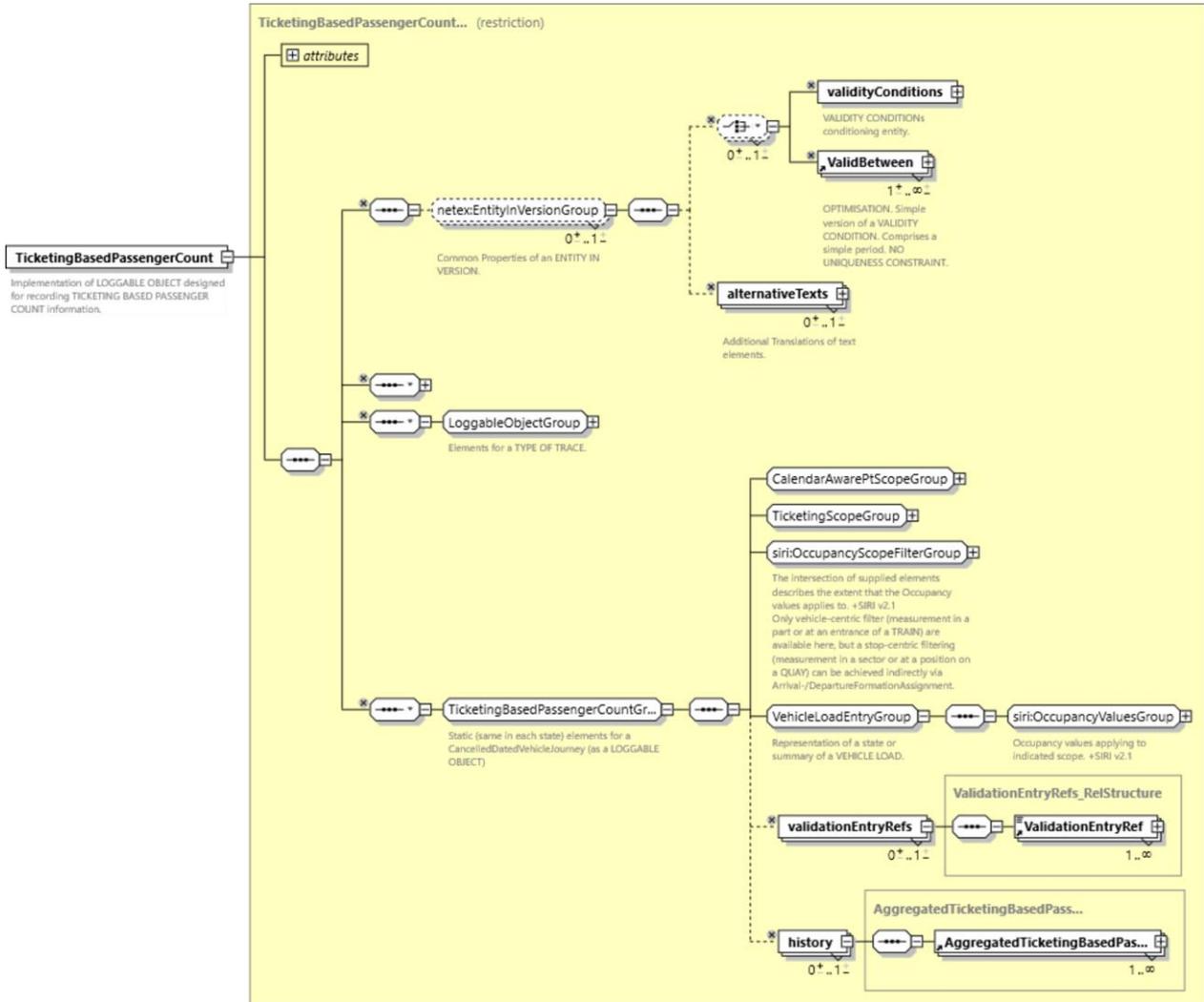


Figure 86 — Ticketing based passenger count delivery structure

The *TicketingBasedPassengerCount* element contains an optional *validationEntryRefs* collection to refer to individual *ValidationEntry* elements adding up within this indicator. It is also possible to exchange the historical build-up (eg. along the course of an OPERATING DAY, represented by multiple TIME BANDS) of this indicator through an *AggregatedTicketingBasedPassengerCount* collection within the *history* element.

ONBOARD DEVICE BASED PASSENGER COUNT data group is organized under *onboardDeviceBasedPassengerCounts* collection element within a RAW DATA FRAME. Since onboard devices can have statistical reliability and confidence, passenger counting originating from an onboard device is modelled with an ESTIMATED VEHICLE LOAD ENTRY.

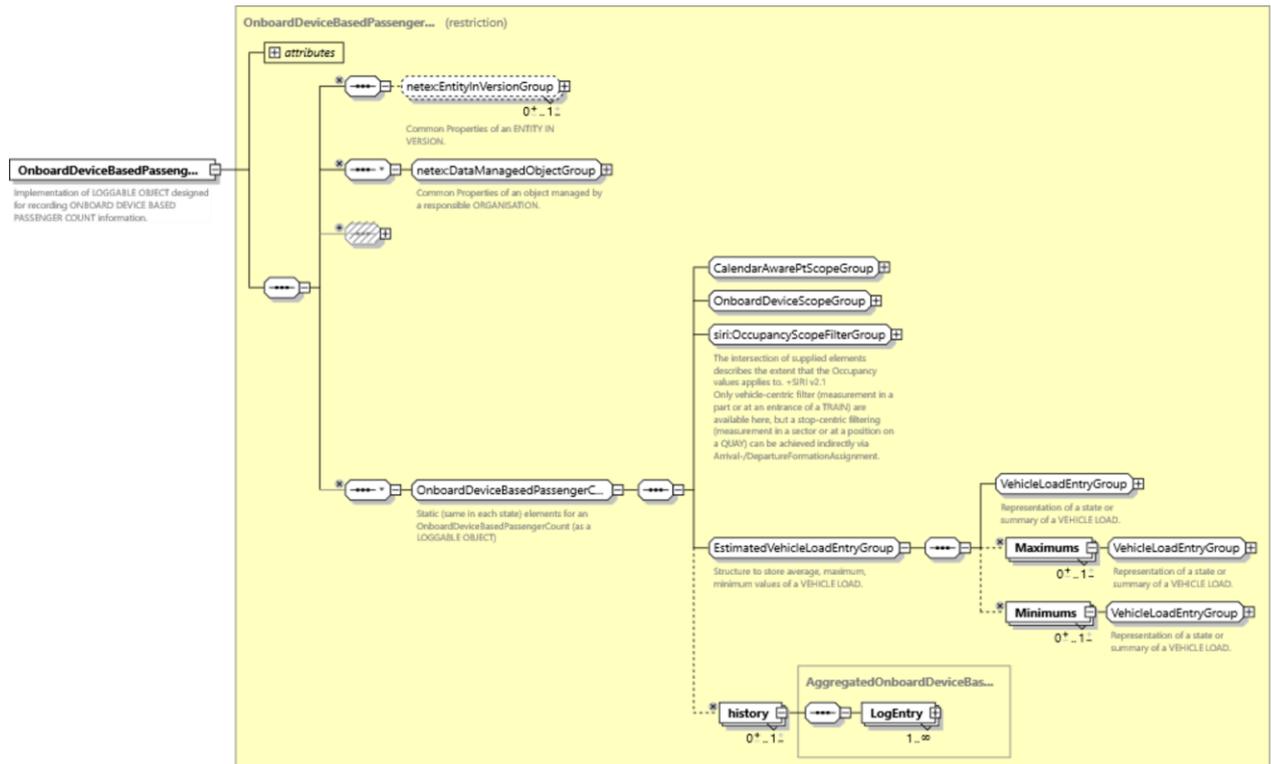


Figure 87 — Onboard device based passenger count delivery structure

It is also possible to exchange the historical build-up (e.g. along the course of an OPERATING DAY, represented by multiple TIME BANDS) of an *OnboardDeviceBasedPassengerCount* indicator through an *AggregatedOnboardDeviceBasedPassengerCount* collection within the *history* element.

EXTERNAL PASSENGER COUNT data group is organized under *externalPassengerCounts* collection element within a RAW DATA FRAME and presents one VEHICLE LOAD ENTRY container for passenger data defined by the scope of the *ExternalPassengerCount* element.

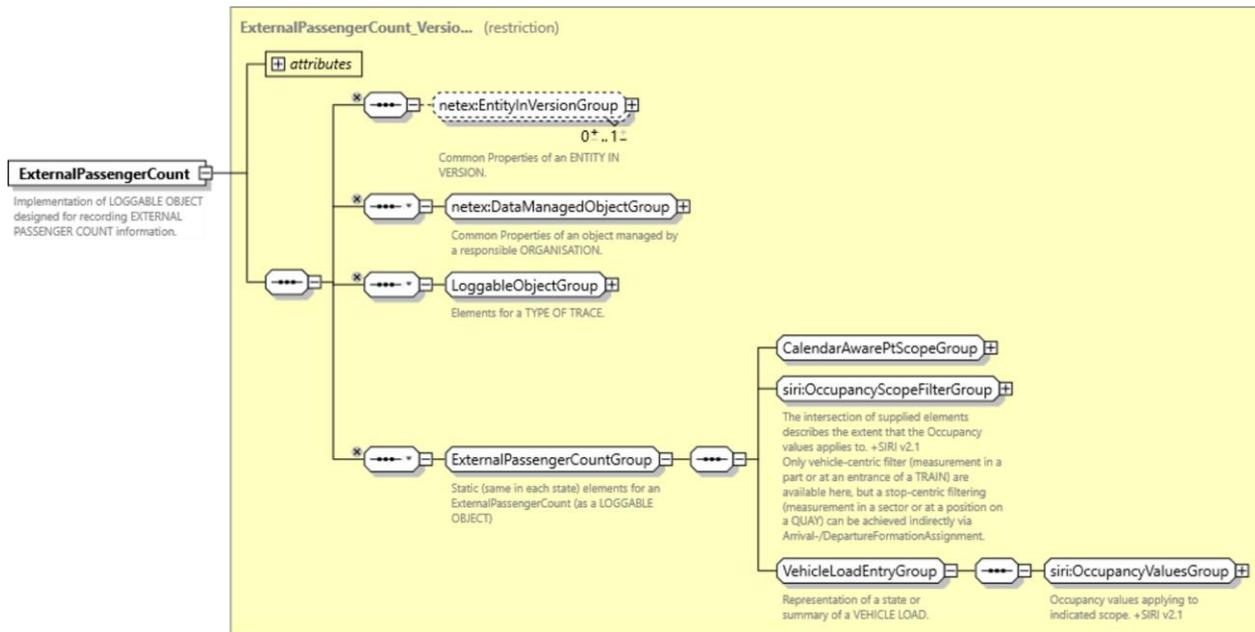


Figure 88 — External passenger count based delivery structure

13.3.2.4 Example

The example below displays TICKETING BASED PASSENGER COUNT delivery on the sample network, with individual VALIDATION ENTRIES, including references within the indicators, also exchanged.

```
<RawDataFrame>
  <PreparednessLevel>Cleansed</PreparednessLevel>
  <validationEntries>
    <ValidationEntry id="Test:VE:1_in">
      <netex:LineRef ref="Test:Line:10"/>
      <netex:ScheduledStopPointRef ref="Test:SSP:T1"/>
      <PassengerTravelEntryType>PassengerCheckIn</PassengerTravelEntryType>
    </ValidationEntry>
    <ValidationEntry id="Test:VE:1_out">
      <netex:LineRef ref="Test:Line:10"/>
      <netex:ScheduledStopPointRef ref="Test:SSP:3"/>
      <PassengerTravelEntryType>PassengerCheckOut</PassengerTravelEntryType>
    </ValidationEntry>
    <ValidationEntry id="Test:VE:2_in">
      <netex:LineRef ref="Test:Line:20"/>
      <netex:ScheduledStopPointRef ref="Test:SSP:T2"/>
      <PassengerTravelEntryType>PassengerCheckIn</PassengerTravelEntryType>
    </ValidationEntry>
    <ValidationEntry id="Test:VE:2_out">
      <netex:SiteRef ref="Test:Site:SiteB"/>
      <netex:LineRef ref="Test:Line:20"/>
      <netex:ScheduledStopPointRef ref="Test:SSP:T3b"/>
      <PassengerTravelEntryType>PassengerCheckOut</PassengerTravelEntryType>
    </ValidationEntry>
    <ValidationEntry id="Test:VE:3_in">
      <netex:SiteRef ref="Test:Site:SiteB"/>
      <PassengerTravelEntryType>PassengerCheckIn</PassengerTravelEntryType>
    </ValidationEntry>
    <ValidationEntry id="Test:VE:3_out">
      <netex:SiteRef ref="Test:Site:SiteA"/>
      <PassengerTravelEntryType>PassengerCheckOut</PassengerTravelEntryType>
    </ValidationEntry>
  </validationEntries>
</RawDataFrame>
```

```

</ValidationEntry>
<ValidationEntry id="Test:VE:4_in">
  <netex:LineRef ref="Test:Line:10"/>
  <netex:ScheduledStopPointRef ref="Test:SSP:T1"/>
  <PassengerTravelEntryType>PassengerCheckIn</PassengerTravelEntryType>
</ValidationEntry>
<ValidationEntry id="Test:VE:4_out">
  <netex:LineRef ref="Test:Line:20"/>
  <netex:ScheduledStopPointRef ref="Test:SSP:6"/>
  <PassengerTravelEntryType>PassengerCheckOut</PassengerTravelEntryType>
</ValidationEntry>
<ValidationEntry id="Test:VE:5_in">
  <netex:LineRef ref="Test:Line:10"/>
  <netex:ScheduledStopPointRef ref="Test:SSP:2"/>
  <PassengerTravelEntryType>PassengerCheckIn</PassengerTravelEntryType>
</ValidationEntry>
<ValidationEntry id="Test:VE:5_out">
  <netex:LineRef ref="Test:Line:10"/>
  <netex:ScheduledStopPointRef ref="Test:SSP:4"/>
  <PassengerTravelEntryType>PassengerCheckOut</PassengerTravelEntryType>
</ValidationEntry>
</validationEntries>
<ticketingBasedPassengerCounts>
  <TicketingBasedPassengerCount>
    <netex:LineRef ref="Test:Line:10"/>
    <PassengerTravelEntryType>PassengerCheckIn</PassengerTravelEntryType>
    <siri:AlightingCount>3</siri:AlightingCount>
    <siri:BoardingCount>3</siri:BoardingCount>
    <validationEntryRefs>
      <ValidationEntryRef ref="Test:VE:1_in"/>
      <ValidationEntryRef ref="Test:VE:1_out"/>
      <ValidationEntryRef ref="Test:VE:4_in"/>
      <ValidationEntryRef ref="Test:VE:4_out"/>
      <ValidationEntryRef ref="Test:VE:5_in"/>
      <ValidationEntryRef ref="Test:VE:5_out"/>
    </validationEntryRefs>
  </TicketingBasedPassengerCount>
  <TicketingBasedPassengerCount>
    <netex:SiteRef ref="Test:Site:SiteB"/>
    <siri:AlightingCount>1</siri:AlightingCount>
    <siri:BoardingCount>1</siri:BoardingCount>
    <validationEntryRefs>
      <ValidationEntryRef ref="Test:VE:2_out"/>
      <ValidationEntryRef ref="Test:VE:3_in"/>
    </validationEntryRefs>
  </TicketingBasedPassengerCount>
</ticketingBasedPassengerCounts>
</RawDataFrame>

```

13.4 Generic KPI Example

This section will describe how to share KPIs which are not closely aligned to a service.

It is likely that a profile will need to be created to describe the specific details of an implementation of this generic KPI to ensure that both the producer and consumer can understand what data is being transferred.

Annex A (informative)

Use Cases

A.1 High Level Generic use cases

This is a particular category, which gathers generic use cases for OpRa potentially relevant to data format and update criteria, and in general all the UCs relevant to XSD format design and future implementation of exchanging data.

Table A.1 — Use case: GENERIC-000-001

Use case: GENERIC-000-001	
Name	Concept definition
Description	Need for clear and unambiguous definitions of concepts to be exchanged. It is expected that the used concepts are fully consistent with Transmodel (and therefore with the ones from NeTEx and SIRI).
Source (Country)	France
Main Actors	All
Raw data needed	NA (need for UML schema and definitions)

Table A.2 — Use case: GENERIC-000-002

Use case: GENERIC-000-002	
Name	Stable data format
Description	Need for a well-defined and stable data format to exchange operational raw data, It is expected that the data format is consistent with other standardized existing format (i.e. NeTEx and SIRI). The format should be sufficiently precise and well defined so it can be requested by authorities in call for tenders (and validated as part of the service delivery).
Source (Country)	France
Main Actors	All
Raw data needed	NA (need for XSD schema)

Table A.3 — Use case: GENERIC-000-003

Use case: GENERIC-000-003	
Name	Data format used for operation raw data description, machine readable and semantic based

Description	It should be possible to use a software tool to import and analyse operation raw data. Therefore, non-machine readable and non-semantic based format (paper, scanned images, pdf, etc.) cannot be used.
Source (Country)	France
Main Actors	All
Raw data needed	NA (need for XSD schema)

Table A.4 — Use case: GENERIC-000-004

Use case: GENERIC-000-004	
Name	Data comparison
Description	It should be possible to compare datasets: - from one delivery to another; - from one area to another; - from different PTO deliveries.
Source (Country)	France
Main Actors	All
Raw data needed	NA

Table A.5 — Use case: GENERIC-000-005

Use case: GENERIC-000-005	
Name	Data subset
Description	It should be possible to describe a subset of the operational raw data of an operator. For example, in numerous situations, operators are operating on a multiple region basis but need to report on a regional base. Therefore, as a minimum, geographical area-based subsets should be supported.
Source (Country)	France
Main Actors	PTO, PTA
Raw data needed	NA

Table A.6 — Use case: GENERIC-000-006

Use case: GENERIC-000-006	
Name	Raw data

Description	Data are expected to be as “raw” as possible and elaborated data should be accompanied by the raw data they are based on.
Source (Country)	France
Main Actors	PTO, PTA
Raw data needed	NA

Table A.7 — Use case: GENERIC-000-007

Use case: GENERIC-000-007	
Name	Up to date data
Description	Data need to be up to date: this also means that validity conditions of datasets should be provided. Furthermore, there shall not be any gap between datasets.
Source (Country)	France
Main Actors	PTO, PTA
Raw data needed	NA

Table A.8 — Use case: GENERIC-000-008

Use case: GENERIC-000-008	
Name	Modular applicability
Description	Utilization of the exchange format shall allow modular data content, enabling continuously growing data depth and detail after initial deployment.
Source (Country)	Hungary
Main Actors	PTO, PTA
Raw data needed	NA

Table A.9 — Use case: GENERIC-000-009

Use case: GENERIC-000-009	
Name	Data visualization best practice
Description	Recommendations for efficient presentation of operational raw data so that raw data can be digested and compared at once. For example, which visualisation technique (i.e. diagram) gives the best presentation for a specific dataset. That would be useful for implementation of dashboards.

Source (Country)	Slovenia
Main Actors	PTO, PTA
Raw data needed	all

A.2 Service Offer use cases

A.2.1 General

This category gathers all the themes and use cases relevant to the Offer of Transport for PT Service (e.g. Spatial and time coverage, offered seats, etc.). This category is further divided into following sub-categories:

- Planned Service Offer (as result of Strategic and Tactical planning phases);
- Actual (measured) service Offer.

A.2.2 Planned Service Offer

Table A.10 — Use case: OFFER-001-001

Use case: OFFER-001-001	
Name	Total length of planned distance to be run by PT Vehicle.
Description	<p>It considers the total planned distance by the PTO to provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the planned distance of each PTO respect the reference Service contract.</p> <p>Formulae: N.A.</p> <p>To be evaluated:</p> <ul style="list-style-type: none"> - per network - per line - per origin/destination (i.e. TM JOURNEY PATTERN) - per area (i.e. region) - per vehicle <p>Shall differentiate regular service and replacement service (especially when it involves a change of mode: for example coach instead of train)</p> <p>Shall differentiate dead-run and service kilometres.</p>
Source (Country)	France, Italy, Slovenia, Hungary
Main Actors	PTO, PTA
UC	Public Transport Service dimensions
MMTIS Use Case Type	Historical
Raw data needed	TM Raw data: DAY_TYPE, DAY_TYPE_ASSIGNMENT, LINES, SERVICE_JOURNEY_PATTERN, SERVICE JOURNEY, SERVICE_LINK, JOURNEY

	<p>ACCOUNTING.</p> <p>NOTE: another option is to use ROUTE LINKs, in this case the use case is similar to OFFER-001-007 with an additional use of JOURNEYs</p> <p>Exchanged data: the total of all the journey length, for all journeys running on the period (line, area, etc.) and separating service, dead-run and replacement services counts.</p>
--	--

Table A.11 — Use case: OFFER-001-002

Use case: OFFER-001-002	
Name	Planned number of offered Fleet Vehicle seats
Description	<p>It considers the total vehicles capacity (seating, standing, special place and wheelchair) by the PTO to provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the planned offered seats offered by each PTO with respect the reference Service contract.</p> <p>Formulae: Number of planned offered seats (seated and standings)</p> <p>To be evaluated:</p> <ul style="list-style-type: none"> - globally; - per peak/off-peak hours; - per line; - per origin/destination (i.e. TM JOURNEY PATTERN); - per hour/SERVICE JOURNEY.
Source (Country)	France, Italy, Slovenia
Main Actors	PTO, PTA
UC	Service dimensions (lines, journeys, seats, etc.)
Raw data needed	<p>TM Raw data: DAY_TYPE, DAY_TYPE_ASSIGNEMENT, LINEs, (JOURNEY_PATTERN), SERVICE JOURNEY, VEHICLE_TYPE, JOURNEY ACCOUNTING</p> <p>Exchanged data: the total of all offered seat, possible separated by FARE CLASS and accessibility (possibly filtered per line, peak hours, etc.).</p>

Table A.12 — Use case: OFFER-001-003

Use case: OFFER-001-003	
Name	Mean Fleet Vehicle seats per offered planned distance
Description	<p>It considers the loading factor capacity of the vehicles considering the number of planned offered seats (seated and standing) normalized on the total planned distance by the PTO to provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal:</p>

Use case: OFFER-001-003	
	Evaluate the loading factor capacity of each PTO respect the reference Service contract. Formulae: Mean planned offered seats (seated and standing) / Total planned distance, calculated on different time frame (e.g. 9:00–10:00, etc.)
Source (Country)	Italy
Main Actors	PTO, PTA
UC	Public Transport Service Load factor
Raw data needed	TM Raw data: DAY_TYPE, DAY_TYPE_ASSIGNMENT, LINES, SERVICE_JOURNEY_PATTERN, SERVICE_JOURNEY, SERVICE_LINK, VEHICLE_TYPE, JOURNEY ACCOUNTING NOTE: THIS IS MAINLY A DIVISION OF THE RESULT OF OFFER-001-001 by OFFER-001-002

Table A.13 — Use case: OFFER-001-004

Use case: OFFER-001-004	
Name	Planned VEHICLE JOURNEYS
Description	It considers the planned number of VEHICLE JOURNEYS of the Public Transport Service in the reference period. Goal: Evaluate the offered VEHICLE JOURNEY of each PTO with respect to the reference Service contract. Formulae: Number of offered VEHICLE JOURNEYS, calculated on different time frame (e.g. 9:00–10:00, etc.) NOTE: Services and dead runs could be distinguished
Source (Country)	Italy, Hungary
Main Actors	PTO, PTA
UC	Service dimensions (lines, journeys, seats, etc.)
Raw data needed	TM Raw data: DAY_TYPE, DAY_TYPE_ASSIGNMENT, LINES, SERVICE_JOURNEY_PATTERN, SERVICE_JOURNEY, JOURNEY ACCOUNTING Exchanged data: the total of the journeys in the considered period, separated by service/dead-run.

Table A.14 — Use case: OFFER-001-005

Use case: OFFER-001-005

Use case: OFFER-001-005	
Name	Planned commercial speed
Description	<p>It considers the planned Fleet commercial speed of the PTO to provide the Public Transport Service in the reference period.</p> <p>Goal: Evaluate the offered commercial speed by each PTO respect the reference Service contract.</p> <p>Formulae: Total planned distance to be performed / Total planned travel time, calculated on different time frame (e.g. 9:00-10:00, etc.), considering only travel time between without STOP_POINT waiting time (MIT)</p>
Source (Country)	Italy
Main Actors	PTO, PTA
UC	Commercial speed
Raw data needed	<p>TM Raw data: DAY_TYPE, DAY_TYPE_ASSIGNMENT, LINES, SERVICE_JOURNEY_PATTERN, SERVICE JOURNEY, SERVICE_LINK, JOURNEY ACCOUNTING, TARGET_PASSING_TIME, (may also use VEHICLE JOURNEY RUN TIME), SERVICE CALENDAR</p> <p>Exchanged data: mean speed (i.e. total of Durations divided by total of Distance attributes of JOURNEY ACCOUNTING) for journeys running (starting or ending ?) in the considered time period (SERVICE CALENDAR).</p>

Table A.15 — Use case: OFFER-001-006

Use case: OFFER-001-006	
Name	Total number of STOP POINTs in the Network
Description	<p>It considers the number of total STOPs in the Network relevant to the Public Transport Service, with reference to PT contract and specific service</p> <p>Goal: Evaluate the available STOPs.</p> <p>Formulae: Total number of STOPs: per line, JOURNEY_PATTERN</p>
Source (Country)	Italy
Main Actors	PTO, PTA
UC	Service dimensions (lines, journeys, seats, etc.)
Raw data needed	<p>TM Raw data: DAY_TYPE, DAY_TYPE_ASSIGNMENT, LINES, (JOURNEY_PATTERN), SERVICE JOURNEY, VEHICLE_TYPE, JOURNEY ACCOUNTING, SCHEDULED_STOP_POINTS, JOURNEY ACCOUNTING, STOP PLACE, TYPE OF STOP PLACE, STOP PLACE ASSIGNMENT</p> <p>NOTE: STOP PLACES could be considered for NETWORK and SSP for JP and VJ.</p> <p>Exchanged data: Total number of STOP PLACES of the network or LINE, total of</p>

Use case: OFFER-001-006	
	SCHEDULED STOP POINTs per JOURNEY PATTERN

Table A.16 — Use case: OFFER-001-007

Use case: OFFER-001-007	
Name	Total distance of ROUTEs
Description	Considers the ROUTE distance (per line and/or per origin/destination) Goal: Evaluate network size Formulae: Sum of ROUTE LINK distances
Source (Country)	France
Main Actors	PTO, PTA
UC	Service dimensions (lines, journeys, seats, etc.)
Raw data needed	TM Raw data: ROUTE, ROUTE LINK, POINT PROJECTION, LINK PROJECTION, LINES, SERVICE PATTERN, DEAD RUN PATTERN Complementary data: infrastructure network (road or rails), on which PROJECTIONs project to (unless the ROUTE LINKs already holds the real link distance on the infrastructure). Exchanged data: total of km per line (selected per LINE, GROUP OF LINE/NETWORK), possibly separating SERVICE and DEAD RUN.

Table A.17 — Use case: OFFER-001-008

Use case: OFFER-001-008	
Name	Provide rescheduling information
Description	Rescheduling, including timetable updates, cancelations, replacements, and additional service and associated reason (roadwork, maintenance, special event, etc.) shall be provided.
Source (Country)	France, Hungary
Main Actors	PTO, PTA
UC	Public Transport Service spatial coverage
	Public Transport Service temporal coverage

	Origin/Destination zones connections
Raw data needed	<p>TM Raw data: CHANGE OF JOURNEY TIMING, CHANGE OF JOURNEY PATTERN, JOURNEY CANCELLATION, PARTIAL JOURNEY CANCELLATION, JOURNEY CREATION, DATED VEHICLE JOURNEY, DATED PASSING TIMES and associated STOP PLACES (requiring JP, POINT in JP, SSP, STOP ASSIGNMENT), PT SITUATION, SITUATION CAUSE and REASON.</p> <p>Complementary data: Possibly linked to a SIRI SX situation.</p> <p>Exchanged data: updated DATED PASSING TIMES connected to STOP PLACES, referencing the DATED VEHICLE JOURNEY, and providing the SITUATION REASON (code).</p> <p>NORMAL DATED PASSING TIMES should also be available in some way (except if it is a JOURNEY CREATION).</p>

Table A.18 — Use case: OFFER-001-009

Use case: OFFER-001-009	
Name	Provide the detailed planned service
Description	The planned service (i.e. network description and timetable description) shall be provided Shall be provided under a well-documented standard and profile (as NeTEx),
Source (Country)	France
Main Actors	PTO, PTA
UC	Service dimensions (lines, journeys, seats, etc.)
Raw data needed	<p>TM Raw data: every data defined in the NeTEx European Passenger Information profile.</p> <p>Exchanged data: a set of XML (NeTEx) files conforming to the profile.</p>

Table A.19 — Use case: OFFER-001-010

Use case: OFFER-001-010	
Name	Provide the expected vehicle type (and train + train components)
Description	Report the planned vehicle type per VEHICLE JOURNEY
Source (Country)	France, Hungary
Main Actors	PTO, PTA
UC	Service dimensions (lines, journeys, seats, etc.)
Raw data needed	<p>TM Raw data: VEHICLE TYPE, TRAIN, TRAIN COMPONENT, SERVICE JOURNEY</p> <p>Exchanged data: VEHICLE TYPE, TRAIN, TRAIN COMPONENT linked to SERVICE JOURNEYS (SERVICE JOURNEYS are expected to be previously exchanged, typically thanks to UC OFFER-001-009....)</p>

Table A.20 — Use case: OFFER-001-011

Use case: OFFER-001-011	
Name	Provide the number of available vehicle
Description	Report the number of vehicle in the fleet (global and per vehicle type)
Source (Country)	France
Main Actors	PTO, PTA
UC	Service dimensions (lines, journeys, seats, etc.)
Raw data needed	TM Raw data: VEHICLE, VEHICLE TYPE, VEHICLE Exchanged data: Total of VEHICLEs per VEHICLE TYPE

A.2.3 Actual (measured) service Offer

Table A.21 — Use case: OFFER-002-001

Use case: OFFER-002-001	
Name	Total Fleet Vehicle performed distance
Description	<p>It considers the total performed distance by the PTO to provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the performed distance of each PTO respect the reference Service contract.</p> <p>Formulae: N.A.</p> <p>To be evaluated:</p> <ul style="list-style-type: none"> - per network; - per line; - per origin/destination (i.e. TM JOURNEY PATTERN); - par area (i.e. region); - per vehicle. <p>Shall differentiate regular service and replacement service (especially when it involves a change of mode: for example coach instead of train). Shall differentiate dead-run and service kilometres.</p>
Source (Country)	Italy, France, Slovenia, Hungary
Main Actors	PTO, PTA
UC	Public Transport Service performed distance
	Public Transport Service dimensions
MMTIS Use Case Type	Observed

Use case: OFFER-002-001	
Raw data needed	<p>TM Raw data: DAY_TYPE, DAY_TYPE_ASSIGNMENT, SERVICE_JOURNEY, MONITORED VEHICLE JOURNEY, VEHICLE MONITORING, DETECTED OPERATION, VEHICLE DETECTING, SERVICE_LINK, JOURNEY ACCOUNTING</p> <p>NOTE: in most situations, this information will come from measurement done by the vehicle equipment. So the basic information will be done from the vehicle measures (DETECTED OPERATION, VEHICLE DETECTING), and then summed and filter using the associated JOURNEY, LINE, DAY TYPE, etc.</p> <p>Exchanged data: the total of all the journey length, for all journeys running on the period (line, area, etc.) and separating service, dead-run and replacement services counts.</p>

Table A.22 — Use case: OFFER-002-002

Use case: OFFER-002-002	
Name	Measured number of offered Fleet Vehicle seats
Description	<p>It considers measured vehicles capacity (seating, standing, special place and wheelchair) by the PTO to provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the measured offered seats offered by each PTO respect the reference Service contract.</p> <p>Formulae: Number of measured offered seats (seated and standing)</p> <p>To be evaluated: - globally; - per peak/off-peak hours; - per line; - per origin/destination (i.e. TM JOURNEY PATTERN).</p>
Source (Country)	France, Italy, Slovenia
Main Actors	PTO, PTA
UC	Service dimensions (lines, journeys, seats, etc.)
Raw data needed	<p>VEHICLE MONITORING, VEHICLE_TYPE, VEHICLE, VEHICLE DETECTING, DAY_TYPE, DAY_TYPE_ASSIGNMENT, JOURNEY ACCOUNTING</p> <p>NOTE: Same OFFER-001-002 but using VEHICLE and VEHICLE DETECTING to link with VEHICLE TYPE. May also use MONITORED FACILITY and more specifically MONITORED VEHICLE EQUIPMENT FACILITY</p> <p>Exchanged data: the total of all offered seat, possible separated by FARE CLASS and accessibility (possibly filtered per line, peak hours, etc.).</p>

Table A.23 — Use case: OFFER-002-003

Use case: OFFER-002-003

Use case: OFFER-002-003	
Name	Mean Fleet Vehicle seats per offered performed distance
Description	<p>It considers the loading factor capacity of the vehicles considering the number of measured offered seats (seated and standing) normalized on the total performed distance by the PTO to provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the loading factor capacity of each PTO respect the reference Service contract.</p> <p>Formulae: Mean measured offered seats (seated and standing) / Total performed distance, calculated on different time frame (e.g. 9:00–10:00, etc.)</p>
Source (Country)	Italy
Main Actors	PTO, PTA
UC	Public Transport Service Load factor
Raw data needed	<p>VEHICLE MONITORING, VEHICLE, VEHICLE_TYPE, DAY_TYPE, DAY_TYPE_ASSIGNMENT, SERVICE_JOURNEY, MONITORED VEHICLE JOURNEY, JOURNEY, ACCOUNTING, SERVICE_LINK</p> <p>NOTE: this is mainly a division of the result of OFFER-002-001 by OFFER-002-002.</p>

Table A.24 — Use case: OFFER-002-004

Use case: OFFER-002-004	
Name	Actual VEHICLE JOURNEYS
Description	<p>It considers the actual number of VEHICLE JOURNEYS to provide the Public Transport Service in the reference period.</p> <p>Goal: Evaluate the actual VEHICLE JOURNEYS of each PTO with respect to the reference Service contract.</p> <p>Formulae: Number of actual VEHICLE JOURNEYS, calculated on different time frame (e.g. 9:00–10:00, etc.)</p> <p>NOTE: Services and dead runs could be distinguished.</p>
Source (Country)	Italy, Hungary
Main Actors	PTO, PTA
UC	Service dimensions (lines, journeys, seats, etc.)
Raw data needed	<p>TM Raw data: VEHICLE MONITORING, DAY_TYPE, DAY_TYPE_ASSIGNMENT, SERVICE_JOURNEY, MONITORED VEHICLE JOURNEY, SERVICE_LINK, JOURNEY ACCOUNTING</p> <p>Exchanged data: the total of the monitored journeys in the considered period, separated by service/dead-run.</p>

Table A.25 — Use case: OFFER-002-005

Use case: OFFER-002-005	
Name	Actual commercial speed
	<p>It considers the actual commercial speed of the PTO to provide the Public Transport Service in the reference period.</p> <p>Goal: Evaluate the actual commercial speed of each PTO respect the reference Service contract.</p> <p>Formulae: Total Performed distance / Total measured travel time, calculated on different time frame (e.g. 9:00-10:00, etc.), considering only travel time between without STOP_POINT waiting time (MIT)</p>
Source (Country)	Italy
Main Actors	PTO, PTA
UC	Commercial speed
Raw data needed	<p>TM Raw data: DAY_TYPE, DAY_TYPE_ASSIGNMENT, SERVICE_JOURNEY, MONITORED VEHICLE JOURNEY, VEHICLE MONITORING, DETECTED OPERATION, VEHICLE DETECTING, SERVICE_LINK, JOURNEY ACCOUNTING, TARGET_PASSING_TIME</p> <p>Exchanged data: mean speed (i.e. mean of speed provided by VEHICLE DETECTING) for journeys running (starting or ending) in the considered period.</p>

Table A.26 — Use case: OFFER-002-010

Use case: OFFER-002-010	
Name	Provide the actual vehicle type (and train + train components)
Description	Report the actual vehicle type per VEHICLE JOURNEY (should be different from planed information as soon as the number of seats, and number of seats for disabled persons or accessibility services are different from the planned version),
Source (Country)	France, Hungary
Main Actors	PTO, PTA
UC	Service dimensions (lines, journeys, seats, etc.)
Raw data needed	<p>TM Raw data: VEHICLE MONITORING, VEHICLE TYPE, TRAIN, TRAIN COMPONENT, VEHICLE (and or VEHICLE MODEL), DATED VEHICLE JOURNEY</p> <p>Exchanged data: VEHICLE TYPE, TRAIN, TRAIN COMPONENT linked to SERVICE JOURNEYS on specific dates (SERVICE JOURNEYS are expected to be previously exchanged, typically thanks to UC OFFER-001-009...)</p>

Table A.27 — Use case: OFFER-002-011

Use case: OFFER-002-011	
Name	Provide the number of vehicle used during operation
Description	Report the number vehicle used during operation (global and per vehicle type)
Source (Country)	France
Main Actors	PTO, PTA
UC	Service dimensions (lines, journeys, seats, etc.)
Raw data needed	<p>TM Raw data: VEHICLE, VEHICLE TYPE, VEHICLE, MONITORED VEHICLE JOURNEY, LOGICAL VEHICLE, VEHICLE ASSIGNMENT</p> <p>Exchanged data: Total of VEHICLEs used during operation, per VEHICLE TYPE, for the considered period.</p>

A.3 Service Externality use cases

This category gathers all the themes and use cases relevant to PT Service Externality, that imply cost or benefit that affects an external party who did not choose to incur that cost or benefit (pollution emissions, safety, etc.).

Table A.28 — Use case: EXTERNALITY-001

Use case: EXTERNALITY-001	
Name	Total number of incidents
Description	<p>It considers the total number of accidents occurred in the reference period (typically the year). Accidents can be related to DRIVER INCIDENT or VEHICLE INCIDENTS and ALARM</p> <p>Goal: Evaluate the interferences of the Public Transport Service with other external mobility</p> <p>Formulae: Number of incidents (vehicle and driver) Number of vehicle incidents Number of driver incidents</p>
Source (Country)	Italy
Main Actors	PTO
UC	Public Transport Service safety
Raw data needed	<p>TM Raw TM data: VEHICLE INCIDENT, DRIVER INCIDENT, ALARM, DATED VEHICLE JOURNEY, OPERATIONAL DAY</p> <p>Exchanged data: the total number of VEHICLE INCIDENT, DRIVER INCIDENT and ALARM for the considered period.</p>

Table A.29 — Use case: EXTERNALITY-002

Use case: EXTERNALITY-002	
Name	Total number of vehicle incidents on the total length run by all vehicles.
Description	<p>It considers the total number of vehicle incidents, and the total performed distance by the PTO to provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the occurred vehicle incidents respect the performed distance of each PTO respect the reference Service contract.</p> <p>Formulae: Total number of vehicle incidents / Total performed distance</p>
Source (Country)	Italy
Main Actors	PTO
UC	Public Transport Service safety
Raw data needed	<p>VEHICLE INCIDENT, DAY_TYPE, DAY_TYPE_ASSIGNMENT, SERVICE_JOURNEY, MONITORED VEHICLE JOURNEY, VEHICLE MONITORING, DETECTING_VEHICLE, SERVICE_LINK, JOURNEY ACCOUNTING</p> <p>Divide EXTERNALITY-001 by OFFER-002-001</p>

Table A.30 — Use case: EXTERNALITY-003

Use case: EXTERNALITY-003	
Name	Total number of vehicle incidents on the total number of used fleet vehicles
Description	<p>It considers the total number of vehicle incidents and the total number of used vehicles by the PTO to provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the occurred vehicle incidents respect the used vehicles by each PTO respect the reference Service contract.</p> <p>Formulae: Total number of vehicle incidents / Total number fleet vehicles</p>
Source (Country)	Italy
Main Actors	PTO
UC	Public Transport Service safety

Raw data needed	VEHICLE INCIDENT, DAY_TYPE, DAY_TYPE_ASSIGNMENT, SERVICE_JOURNEY, MONITORED VEHICLE JOURNEY, VEHICLE MONITORING, DETECTING_VEHICLE, JOURNEY ACCOUNTING Divide EXTERNALITY-001 by OFFER-002-011
-----------------	---

Table A.31 — Use case: EXTERNALITY-004

Use case: EXTERNALITY-004	
Name	Total number of vehicle incidents on the total travelled time Total number of driver incidents on the total travelled time
Description	It considers the total number of vehicle/driver incidents and the total travelled time by the PTO to provide the Public Transport Service in the reference period (typically the year). Goal: Evaluate the occurred vehicle/driver incidents respect the travelled time of each PTO respect the reference Service contract. Formulae: Total number of vehicle incidents / Total travelled time Total number of driver incidents / Total travelled time
Source (Country)	Italy
Main Actors	PTO
UC	Public Transport Service safety
Raw data needed	VEHICLE INCIDENT, DRIVER INCIDENT, DAY_TYPE, DAY_TYPE_ASSIGNMENT, SERVICE_JOURNEY, MONITORED VEHICLE JOURNEY, VEHICLE MONITORING, DETECTING_VEHICLE, JOURNEY ACCOUNTING Divide EXTERNALITY-001 by OFFER-002-011

Table A.32 — Use case: EXTERNALITY-005

Use case: EXTERNALITY-005	
Name	Total pollution emissions
Description	It considers the total pollution emissions by the PTO to provide the Public Transport Service in the reference period (typically the year). Goal: Evaluate the total pollution emissions of each PTO respect the reference Service contract. Formulae: N.A.
Source (Country)	Italy
Main Actors	PTA

Use case: EXTERNALITY-005	
UC	Public Transport Service pollution emission
Raw data needed	<p>VEHICLE TYPE, DAY_TYPE, DAY_TYPE_ASSIGNMENT, SERVICE_JOURNEY, MONITORED VEHICLE JOURNEY, VEHICLE MONITORING, DETECTING_VEHICLE, JOURNEY ACCOUNTING</p> <p>Not in Transmodel; approximated using the VEHICLE TYPE information (pollution expected per vehicle type combined by the total distance run by each vehicle type, compare with OFFER-002-011). However, the environmental description of the VEHICLE is not in Transmodel and has to be provided from “outside”. If so, this could be solved by a use case being: “Total distance run per vehicle type for a specific period of time” Report polluting information on a per vehicle type/km and per pollutant base (passenger can be easily deduced by combination with other data).</p>

Table A.33 — Use case: EXTERNALITY-006

Use case: EXTERNALITY-006	
Name	Safety and security issues
Description	<p>To report safety and security related incidents (clearly identified from other incidents)</p> <p>TO BE DETAILED</p>
Source (Country)	France
Main Actors	All
UC	Public Transport Service safety
Raw data needed	<p>[Note] PROVIDED BY EXTERNALITY-001 is ALARM are included (with a separate count). But I would suggest to change that UC on “Total number of DRIVER INCIDENT, VEHICLE INCIDENTS or ALARM of a specific type” = > TM Raw TM data: VEHICLE INCIDENT, DRIVER INCIDENT, ALARM, DATED VEHICLE JOURNEY, OPERATIONAL DAY</p> <p>Exchanged data: the total number of DRIVER INCIDENT, VEHICLE INCIDENTS or ALARM of a specific type for a time period.</p>

Table A.34 — Use case: EXTERNALITY-007

Use case: EXTERNALITY-007	
Name	Events (<i>situations</i>) on the network
Description	<p>To report events on the network:</p> <ul style="list-style-type: none"> - on a per event base; - per line base; - time and day-type base.

	Planned and unplanned events should be clearly identified.
Source (Country)	France
Main Actors	All
UC	Public Transport Service safety
Raw data needed	<p>TM Raw TM data: PT SITUATION, PT SITUATION AFFECTED SCOPE, PT SITUATION CONSEQUENCE COPE, SITUATION CAUSE, SITUATION REASON</p> <p>Exchanged data: the total number of PT SITUATION, global, per line, and per situation type of consequence (note: needs a Type Of PT Situation Consequence not in TM, but Ok in SIRI)</p>

A.4 Service Economy use cases

A.4.1 General

This category gathers all the themes and use cases relevant to the Economy of Transport for PT Service in terms of incoming funds (sold tickets, etc.) and expenses (costs, etc.). This category is further divided into following sub-categories:

- Income Service Economy;
- Expenses Service Economy.

A.4.2 Income Service Economy

Table A.35 — Use case: ECONOMY-001-001

Use case: ECONOMY-001-001	
Name	PT service costs
Description	<p>It considers all the costs to provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the production costs of each PTO respect the reference Service contract. Moreover, Fuel cost, Amortization value of vehicles, Maintenance cost for vehicles, Labour (drivers) cost, Insurance cost (drivers, passengers), Cost for stop- and parking- places.</p> <p>Formulae: N.A.</p>
Source (Country)	Italy
Main Actors	PTO, PTA
UC	Public Transport Service costs
Raw data needed	<p>Costs</p> <p>NOTE: on what can we base “cost” definition and cost categorization.</p>

	NOT IN TRANSMODEL.
--	--------------------

Table A.36 — Use case: ECONOMY-001-002

Use case: ECONOMY-001-002	
Name	Total costs divided by total planned length to be run by all Vehicles.
Description	<p>It considers all the costs to provide the Public Transport Service respect the performed distance (km).</p> <p>Goal: Evaluate the production costs respect the total distance performed by all the vehicles in the reference period (typically the year).</p> <p>Formulae: Total costs / total performed distance</p>
Source (Country)	Italy, Slovenia
Main Actors	PTO, PTA
UC	Public Transport Service costs
Raw data needed	<p>DAY_TYPE, DAY_TYPE_ASSIGNMENT, LINES, SERVICE_JOURNEY_PATTERN, SERVICE_JOURNEY, SERVICE_LINK, JOURNEY ACCOUNTING, costs</p> <p>Divide ECONOMY-001-001 by OFFER-001-001</p>

Table A.37 — Use case: ECONOMY-001-003

Use case: ECONOMY-001-003	
Name	Total costs divided by a per single used vehicles basis
Description	It considers all the costs to provide the Public Transport Service divided by a per single used vehicles basis. Goal: Evaluate the production costs respect the used vehicles to provide the service in the reference period (typically the year). Formulae: Total costs / total number of used vehicles
Source (Country)	Italy
Main Actors	PTO, PTA
UC	Public Transport Service costs
Raw data needed	DAY_TYPE, DAY_TYPE_ASSIGNMENT, SERVICE_JOURNEY, MONITORED VEHICLE JOURNEY, VEHICLE MONITORING, JOURNEY ACCOUNTING, costs Divide ECONOMY-001-001 by OFFER-002-011

Table A.38 — Use case: ECONOMY-001-004

Use case: ECONOMY-001-004	
Name	Total costs divided by total number of passengers and performed distance
Description	It considers all the costs to provide the Public Transport Service divided by total number of passengers and performed distance. Goal: Evaluate the production costs respect total number of transported passengers and total performed distance in the reference period (typically the year). Formulae: Total costs / (total number of passengers * total distance)
Source (Country)	Italy
Main Actors	PTO, PTA
UC	Public Transport Service costs
Raw data needed	DAY_TYPE, DAY_TYPE_ASSIGNMENT, LINEs, SERVICE_JOURNEY_PATTERN, SERVICE JOURNEY, SERVICE_LINK, VEHICLE_TYPE, VEHICLE, JOURNEY ACCOUNTING, (VALIDATION), costs Combine ECONOMY-001-001 with DEMAND-001-001 and OFFER-001-001

Table A.39 — Use case: ECONOMY-001-005

Use case: ECONOMY-001-005	
Name	Information on reimbursements
Description	Information on reimbursements (due to delays or cancellations) with associated reason, on a per line and temporal basis.
Source (Country)	France
Main Actors	PTO, PTA
UC	Public Transport Service costs
Raw data needed	TM Raw TM data: REFUNDING, PT SITUATION, PT SITUATION CONSEQUENCE, SITUATION CAUSE, SITUATION REASON Exchanged data: List of REFUNDINGS with associated CAUSE/REASON.

A.4.3 Expenses Service Economy

Table A.40 — Use case: ECONOMY-002-001

Use case: ECONOMY-002-001	
Name	Total revenues
Description	It considers all the revenues of PTO in the reference period (typically the year). Goal: Evaluate the revenues of each PTO respect the reference Service contract. Formulae: N.A.
Source (Country)	Italy
Main Actors	PTO, PTA
UC	Public Transport Service incoming funds
Raw data needed	revenues NOT IN TRANSMODEL.

Table A.41 — Use case: ECONOMY-002-002

Use case: ECONOMY-002-002	
Name	Total revenues divided by total length run by all Vehicles.
Description	It considers all the revenues to provide the Public Transport Service respect the performed distance (km). Goal: Evaluate the revenues respect the total distance performed by all the vehicles in the reference period (typically the year). Formulae: Total revenues / total performed distance

Use case: ECONOMY-002-002	
Source (Country)	Italy
Main Actors	PTO, PTA
UC	Public Transport Service incoming funds
Raw data needed	DAY_TYPE, DAY_TYPE_ASSIGNMENT, LINEs, SERVICE_JOURNEY_PATTERN, SERVICE JOURNEY, SERVICE_LINK, JOURNEY ACCOUNTING, revenues Divide ECONOMY-002-001 by OFFER-002-001

Table A.42 — Use case: ECONOMY-002-003

Use case: ECONOMY-002-003	
Name	Total revenues divided on a per single used vehicles basis
Description	It considers all the revenues to provide the Public Transport Service divided by a per single used vehicles basis. Goal: Evaluate the revenues respect the used vehicles to provide the service in the reference period (typically the year). Formulae: Total revenues / total number of used vehicles
Source (Country)	Italy
Main Actors	PTO, PTA
UC	Public Transport Service incoming funds
Raw data needed	DAY_TYPE, DAY_TYPE_ASSIGNMENT, SERVICE_JOURNEY, MONITORED VEHICLE JOURNEY, VEHICLE MONITORING, JOURNEY ACCOUNTING, revenues Divide ECONOMY-001-002 by OFFER-002-011

Table A.43 — Use case: ECONOMY-002-004

Use case: ECONOMY-002-004	
Name	Total revenues divided by total number of passengers and performed distance
Description	It considers all the revenues to provide the Public Transport Service divided by total number of passengers and performed distance. Goal: Evaluate the revenues respect total number of transported passengers and total performed distance in the reference period (typically the year). Formulae: Total revenues / (total number of passengers * total distance)
Source (Country)	Italy
Main Actors	PTO, PTA

UC	Public Transport Service incoming funds
Raw data needed	DAY_TYPE, DAY_TYPE_ASSIGNMENT, LINEs, SERVICE_JOURNEY_PATTERN, SERVICE JOURNEY, SERVICE_LINK, VEHICLE_TYPE, VEHICLE, JOURNEY ACCOUNTING, (VALIDATION), revenues Combine ECONOMY-001-001 with DEMAND-002-001 and OFFER-002-001

Table A.44 — Use case: ECONOMY-002-005

Use case: ECONOMY-002-005	
Name	Total revenues coming from sold fare products
Description	It considers all the revenues coming from sold fare products. Goal: Evaluate the revenues respect total number of sold tickets in the reference period (typically the year). Formulae: Sold fare products total revenues / (total number of sold fare products)
Source (Country)	Italy, Slovenia
Main Actors	PTO, PTA
UC	Public Transport Service incoming funds
Raw data needed	TM Raw TM data: CUSTOMER PURCHASE PACKAGE, SALES TRANSACTION, CUSTOMER PURCHASE PACKAGE ELEMENT, SALES OFFER PACKAGE ELEMENT, SERVICE ACCESS RIGHT, FARE PRODUCT Exchanged data: Total of amounts SALES TRANSACTION, total corresponding instances of FARE PRODUCT.

Table A.45 — Use case: ECONOMY-002-006

Use case: ECONOMY-002-006	
Name	Total revenues coming from sold fare products divided by a per single used vehicles basis
Description	It considers all the revenues coming from sold fare products to provide the Public Transport Service divided by a per single used vehicles basis. Goal: Evaluate the revenues coming from sold fare products respect the used vehicles to provide the service in the reference period (typically the year). Formulae: Sold fare products total revenues / total number of used vehicles
Source (Country)	Italy
Main Actors	PTO, PTA
UC	Public Transport Service incoming funds

Raw data needed	DAY_TYPE, DAY_TYPE_ASSIGNMENT, SERVICE_JOURNEY, MONITORED VEHICLE JOURNEY, VEHICLE MONITORING, JOURNEY ACCOUNTING, revenue fare product, sold fare products Combine ECONOMY-002-005 with OFFER-002-011
-----------------	---

Table A.46 — Use case: ECONOMY-002-007

Use case: ECONOMY-002-007	
Name	Total revenues coming from sold fare products divided by total number of passengers and total length run by vehicles.
Description	It considers all the revenues coming from sold fare products to provide the Public Transport Service divided by total number of passengers and total length run by all vehicles. Goal: Evaluate the revenues coming from sold fare products respect total number of transported passengers and total performed distance in the reference period (typically the year). Formulae: $\text{Sold fare products total revenues} / (\text{total number of passengers} * \text{total distance})$
Source (Country)	Italy
Main Actors	PTO, PTA
UC	ECONOMY-002-007
Raw data needed	DAY_TYPE, DAY_TYPE_ASSIGNMENT, LINEs, SERVICE_JOURNEY_PATTERN, SERVICE_JOURNEY, SERVICE_LINK, VEHICLE_TYPE, VEHICLE, JOURNEY ACCOUNTING, (VALIDATION), revenue fare product, sold fare products Combine ECONOMY-001-001 with DEMAND-002-001 and OFFER-002-001

Table A.47 — Use case: ECONOMY-002-008

Use case: ECONOMY-002-008	
Name	Ratio between total revenues and total costs
Description	It considers all the costs to provide the Public Transport Service and the total revenues in the reference period (typically the year). Goal: Evaluate the ration between revenues and costs of each PTO respect the reference Service contract. Formulae: $\text{Total revenues} / \text{Total costs}$
Source (Country)	Italy
Main Actors	PTO, PTA
Raw data needed	revenues / costs Divide ECONOMY-001-002 by ECONOMY-001-001

Table A.48 — Use case: ECONOMY-002-009

Use case: ECONOMY-002-009	
Name	Ratio between total revenues coming from sold fare products and total costs
Description	It considers all the costs to provide the Public Transport Service and the total revenues total costs in the reference period (typically the year). Goal: Evaluate the ratio between revenues total costs and costs of each PTO respect the reference Service contract. Formulae: Sold fare products total revenues / Total costs
Source (Country)	Italy
Main Actors	PTO, PTA
UC	Public Transport Service incoming funds
Raw data needed	sold fare products / costs Divide ECONOMY-001-005 by ECONOMY-001-001

Table A.49 — Use case: ECONOMY-002-010

Use case: ECONOMY-002-010	
Name	Report distance and number of trip per travel card
Description	Report the mean distance and mean number of trip per season ticket type (any product allowing several journeys). Goal: Evaluate the real use of travel cards, entitlement cards, season fare products, etc. Formulae:
Source (Country)	France
Main Actors	PTO, PTA
UC	Public Transport Service incoming funds
Raw data needed	TM Raw TM data: CUSTOMER PURCHASE PACKAGE, SALES TRANSACTION, CUSTOMER PURCHASE PACKAGE ELEMENT, SALES OFFER PACKAGE ELEMENT, SERVICE ACCESS RIGHT, FARE PRODUCT (filtered by season ticket/card, etc.). TRIP PATTERN, PT TRIP, LEG, PT RIDE, MONITORED TRIP PATTERN (liked to above CUSTOMER PURCHASE PACKAGE). Exchanged data: total number of MONITORED TRIP PATTERN per season ticket, card, etc. and associated total PT RIDE distances.

Table A.50 — Use case: ECONOMY-002-011

Use case: ECONOMY-002-011	
Name	Total revenues coming from sold fare products on a per line (or group of lines) bases
Description	It considers all the revenues coming from sold fare products to provide the Public Transport Service on a per line basis. Goal: Evaluate the distribution of revenues coming from sold fare products respect the lines (or group of lines). Formulae: $\text{Sold fare products total revenues} / (\text{number of lines})$
Source (Country)	France, Slovenia
Main Actors	PTO, PTA
UC	Public Transport Service incoming funds
Raw data needed	TM Raw TM data: CUSTOMER PURCHASE PACKAGE, SALES TRANSACTION, CUSTOMER PURCHASE PACKAGE ELEMENT, SALES OFFER PACKAGE ELEMENT, SERVICE ACCESS RIGHT, FARE PRODUCT (filtered by season ticket/card etc). TRIP PATTERN, PT TRIP, LEG, PT RIDE, MONITORED TRIP PATTERN (linked to above CUSTOMER PURCHASE PACKAGE). Exchanged data: Total of amount of CUSTOMER PURCHASE PACKAGE per lines used in PT RIDES.

Table A.51 — Use case: ECONOMY-002-012

Use case: ECONOMY-002-012	
Name	Total income from unattended luggage transported
Description	Income from unattended luggage transported (per concessionaire, per sub-contractors). Goal: To obtain the income from unattended luggage transported in the reporting period (typically the month and year) for the purpose of request for compensation in frame of a concessionaire contract. Formulae: Total income from unattended luggage transported
Source (Country)	Slovenia
Main Actors	PTO, PTA
UC	Public Transport Service incoming funds
Raw data needed	Same as ECONOMY-002-005 with a filter on FARE PRODUCT. New TYPE OF FARE PRODUCT (unattended luggage) is needed. NOTE: could be generalized to any kind of Fare product.

Table A.52 — Use case: ECONOMY-002-013

Use case: ECONOMY-002-013	
Name	Total income from public sources
Description	Income from public sources (i.e. schools, army; generally any public funding). Goal: To obtain the income from any public sources (contracts with schools, army, etc.) in the reporting period (typically the month and year) for the purpose of request for compensation in frame of a concessionaire contract. Formulae: Total income from public sources
Source (Country)	Slovenia
Main Actors	PTO, PTA
UC	Public Transport Service incoming funds
Raw data needed	Incomes from public sources NOT in TRANSMODEL. (AT) To get a total estimation (ballpark figure) of PTO economy under concession this is needed. Should we introduce CONCESSION (or SERVICE CONTRACT)?

Table A.53 — Use case: ECONOMY-002-014

Use case: ECONOMY-002-014	
Name	Value of Fleet Vehicles
Description	It considers the financial Initial value of vehicle and current value of vehicle Goal: Evaluate the value of vehicles by each PTO respect the reference Service contract. Formulae: Initial value of vehicle, Current value of vehicle.
Source (Country)	Slovenia
Main Actors	PTO
UC	Public Transport Service incoming funds
Raw data needed	Needs VEHICLES to have a "Price" (initial and current) attribute.... (to be defined as something like "total aggregate purchase price including all equipment") TM Raw TM data: VEHICLE, VEHICLE TYPE Exchanged data: Total of VEHICLES prices (initial and current).

Table A.54 — Use case: ECONOMY-002-015

Use case: ECONOMY-002-015	
Name	Actual number of passengers luggage items
Description	Provision of actual number of passenger luggage items per line Goal: to convey income generated from passenger luggage items
Source (Country)	Slovenia
Main Actors	PTO, PTA
UC	Public Transport Service incoming funds
Raw data needed	NOTE: in TM only have LUGGAGE ALLOWANCE as part of FARE PRODUCT. New TYPE OF FARE PRODUCT is needed. It should be recorded at the level of MONITORED VEHICLE JOURNEY. PASSENGER TRAVEL ENTRY, VALIDATED ACCESS, CUSTOMER PURCHASE PACKAGE, SALES TRANSACTION, CUSTOMER PURCHASE PACKAGE ELEMENT, SALES OFFER PACKAGE ELEMENT

Table A.55 — Use case ECONOMY-002-016

Use case: ECONOMY-002-016	
Name	Actual number of unattended luggage items
Description	Provision of number of unattended luggage transported per line (customer pays for the scheduled luggage delivery) Goal: To convey income from unattended luggage transported. Formulae: Total number of unattended luggage transported
Source (Country)	Slovenia
Main Actors	PTO, PTA
UC	Public Transport Service incoming funds
Raw data needed	NOTE: in TM only have LUGGAGE ALLOWANCE as part of FARE PRODUCT. New TYPE OF FARE PRODUCT (unattended luggage) is needed. It should be recorded at the level of MONITORED VEHICLE JOURNEY. CUSTOMER PURCHASE PACKAGE, SALES TRANSACTION, CUSTOMER PURCHASE PACKAGE ELEMENT, SALES OFFER PACKAGE ELEMENT

A.5 Service Efficiency use cases

Table A.56 — Use case: EFFICIENCY-001

Use case: EFFICIENCY-001	
Name	Fleet Vehicles age

Description	It considers the number of vehicles on a per 5 years age range (e.g. older than 5 years,10 years, etc) used by the PTO to provide the Public Transport Service in the reference period (typically the year). Goal: Evaluate the technical fleet efficiency of each PTO respect the reference Service contract. Formulae: N.A.
Source (Country)	Italy, Slovenia, Hungary
Main Actors	PTO, PTA
UC	Public Transport Fleet dimensions
Raw data needed	TM Raw TM data: VEHICLE, VEHICLE TYPE NOTE: Needs VEHICLES to have an “Age” attribute Exchanged data: Mean vehicle age, and distribution per slice of N years (N between 1 and 25)

Table A.57 — Use case: EFFICIENCY-002

Use case: EFFICIENCY-002	
Name	Fleet Vehicles age distribution
Description	It considers the number of vehicles on a per 5 years age range (e.g. older than 5 years,10 years, etc) respect the total number of fleet vehicles, used by the PTO to provide the Public Transport Service in the reference period (typically the year). Goal: Evaluate the technical fleet efficiency of each PTO respect the reference Service contract. Formulae: Number of vehicles in a specific age range / total number of vehicles
Source (Country)	Italy, Slovenia
Main Actors	PTO, PTA
UC	Public Transport Fleet dimensions
Raw data needed	NOTE: Needs VEHICLES to have an “Age” attribute. TM Raw TM data: VEHICLE, VEHICLE TYPE Exchanged data: Mean vehicle age, and distribution per slice of N years (N between 1 and 25).

Table A.58 — Use case: EFFICIENCY-003

Use case: EFFICIENCY-003	
Name	Total Fleet Vehicle performed distance by employee
Description	<p>It considers the total performed distance normalized by total number of employees of each PTO lto provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the efficiency of each PTO considering the number of total employees and in the respect the reference Service contract.</p> <p>Formulae: Performed distance / number of employees</p>
Source (Country)	Italy
Main Actors	PTO, PTA
UC	Public Transport Service performed distance
Raw data needed	<p>TM Raw TM data: see OFFER-002-001 + EMPLOYEE, ORGANIZATION, OPERATOR</p> <p>Exchanged data: OFFER-002-001 divided by the total number of EMPLOYEE</p>

Table A.59 — Use case: EFFICIENCY-004

Use case: EFFICIENCY-004	
Name	Total Fleet Vehicle performed distance by driver
Description	<p>It considers the total performed distance normalized by total number of drivers of each PTO to provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the efficiency of each PTO considering the number of total drivers and in the respect the reference Service contract.</p> <p>Formulae: Performed distance / number of drivers</p>
Source (Country)	Italy
Main Actors	PTO, PTA
UC	Public Transport Service performed distance
Raw data needed	<p>DAY_TYPE, DAY_TYPE_ASSIGNMENT, SERVICE_JOURNEY, MONITORED VEHICLE JOURNEY, VEHICLE MONITORING, DETECTING_VEHICLE, SERVICE_LINK, JOURNEY ACCOUNTING, LOGICAL DRIVER</p> <p>TM Raw TM data: see OFFER-002-001 + EMPLOYEE, DRIVER, ORGANIZATION, OPERATOR</p> <p>Exchanged data: OFFER-002-001 divided by the total number of DRIVER</p>

Table A.60 — Use case: EFFICIENCY-005

Use case: EFFICIENCY-005	
Name	Mean Fleet Vehicle performed distance
Description	<p>It considers the mean performed distance of each vehicle of each PTO to provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the efficiency in the usage of vehicles of each PTO respect the reference Service contract.</p> <p>Formulae: Performed distance / number of vehicles on a per contract basis.</p>
Source (Country)	Italy
Main Actors	PTO, PTA
UC	Public Transport Service performed distance
Raw data needed	<p>DAY_TYPE, DAY_TYPE_ASSIGNMENT, SERVICE_JOURNEY, MONITORED VEHICLE JOURNEY, VEHICLE MONITORING, DETECTING_VEHICLE, SERVICE_LINK, JOURNEY ACCOUNTING, JOURNEY ACCOUNTING</p> <p>Divide OFFER-002-001 by OFFER-002-011 ... or OFFER-001-001 by OFFER-001-011</p>

Table A.61 — Use case: EFFICIENCY-006

Use case: EFFICIENCY-006	
Name	PTO employees categories distribution
Description	<p>It considers the category of the employees for each vehicle of each PTO to provide the Public Transport Service in the reference period (typically the year).</p> <p>Goal: Evaluate the employees distribution in the different categories (drivers, administrative, engineering, etc.) of each PTO respect the reference Service contract.</p> <p>Formulae: Percentage and absolute distribution of employees for all contractual categories</p>
Source (Country)	Italy, Slovenia
Main Actors	PTA
UC	Public Transport Operator dimensions
	employees*, employees categories* (AT) TM Raw TM data: In TM we have EMPLOYEE and QUALIFICATION

Table A.62 — Use case: EFFICIENCY-007

Use case: EFFICIENCY-007	
Name	Percentage of drivers over the total number of employees
Description	It considers the total number of employees and drivers for each vehicle of each PTO to provide the Public Transport Service in the reference period (typically the year). Goal: Evaluate the percentage of drivers on total number of employees of each PTO respect the reference Service contract. Formulae: Total number of drivers / Total number of employees
Source (Country)	Italy
Main Actors	PTA
UC	Public Transport Service performed distance
	Public Transport Operator dimensions
Raw data needed	employees*, physical drivers* TM Raw TM data: EMPLOYEE, DRIVER, QUALIFICATION, ORGANIZATION, OPERATOR Exchanged data: Total number of DRIVER and EMPLOYEE per QUALIFICATION,

Table A.63 — Use case: EFFICIENCY-008

Use case: EFFICIENCY-008	
Name	Total distance performed on a per vehicle basis
Description	It considers the total performed distance by each vehicle to provide the Public Transport Service in the reference period (typically the year). Goal: Evaluate the distance performed by each vehicle for maintenance purposes. Formulae: Total number of km for each fleet vehicle
Source (Country)	Italy
Main Actors	PTO
UC	Public Transport Service performed distance

Use case: EFFICIENCY-008	
Raw data needed	<p>DAY_TYPE, DAY_TYPE_ASSIGNMENT, SERVICE_JOURNEY, MONITORED VEHICLE JOURNEY, VEHICLE MONITORING, DETECTING_VEHICLE, SERVICE_LINK, JOURNEY ACCOUNTING</p> <p>Multiple way of calculation: need to be aligned with OFFER-002-001 and OFFER-002-011</p> <p>TM Raw TM data: VEHICLE, VEHICLE TYPE, VEHICLE, MONITORED VEHICLE JOURNEY, LOGICAL VEHICLE, VEHICLE ASSIGNMENT</p> <p>+DAY_TYPE, DAY_TYPE_ASSIGNMENT, SERVICE_JOURNEY, VEHICLE MONITORING, DETECTED OPERATION, VEHICLE DETECTING, SERVICE_LINK, JOURNEY ACCOUNTING</p> <p>Exchanged data: Total of JOURNEY ACCOUNTING distance performed for each single vehicle.</p>

Table A.64 — Use case: EFFICIENCY-009

Use case: EFFICIENCY-009	
Name	Mean Time Between Failure (MTBF) on a per vehicles basis.
Description	<p>It considers the occurred and resolve date of each Failure for each vehicle used to provide the Public Transport Service in the reference period (typically the year). Goal: Evaluate the maintenance efficiency of each vehicles to monitor the maintenance purposes. Formulae: MTBF</p>
Source (Country)	Italy
Main Actors	PTO
UC	Public Transport Fleet dimensions
Raw data needed	<p>TM Raw TM data: VEHICLE, VEHICLE TYPE, VEHICLE, MONITORED VEHICLE JOURNEY, LOGICAL VEHICLE, VEHICLE INCIDENT</p> <p>Exchanged data: For each VEHICLE, divide the period by total number of VEHICLE INCIDENT (possibly filtered by type or incident).</p>

Table A.65 — Use case: EFFICIENCY-009

Use case: EFFICIENCY-010	
Name	Mean delay on a per ROUTE, SERVICE JOURNEY or STOP basis.

Description	It considers the occurred delay at a specific point (STOP_POINT) for each line of the Public Transport Service in the reference period (day, month). Goal: Evaluate the efficiency of the Service in terms of delays. Formulae: Mean (Arrival Time — Scheduled Time)
Source (Country)	Italy, Hungary, France
Main Actors	PTO
UC	Service delays and inefficiencies
Raw data needed	TM Raw TM data: DELAY, PT SITUATION CONSEQUENCE, PT SITUATION, SITUATION CAUSE, PT SITUATION CONSEQUENCE SCOPE, ROUTE, SERVICE JOURNEY, SCHEDULED STOP POINT, STOP PLACE (through assignment). Exchanged data: Mean DELAY per SERVICE JOURNEYS using the specified ROUTE and mean for all SERVICE JOURNEYS of the ROUTE; Mean delay per STOP PLACE unused on the ROUTE; A precise definition of delay has to be provided, since it can be provided at stop level, at start and end, can be valid for only a part of the Journey, etc.

Table A.66 — Use case: EFFICIENCY-011

Use case: EFFICIENCY-011	
Name	Number of late services
Description	Should be available: <ul style="list-style-type: none"> - on a per stop base (especially first and last stops); - on an origin/destination base; - on a peak/off-peak base; - on an area (i.e. region or city) base. The definition of late shall be very clearly stated: <ul style="list-style-type: none"> - maximum delay to be paid on time (may depend on mode); - there may be a specific definition for headway services; - cancelled services shall be clearly identified; - additional services shall be clearly identified.
Source (Country)	France, Hungary
Main Actors	All
UC	Service delays and inefficiencies

Use case: EFFICIENCY-011	
Raw data needed	<p>More detailed version of EFFICIENCY-011</p> <p>TM Raw TM data: DELAY, PT SITUATION CONSEQUENCE, PT SITUATION, SITUATION CAUSE, PT SITUATION CONSEQUENCE SCOPE, ROUTE, SERVICE JOURNEY, SCHEDULED STOP POINT, STOP PLACE (through assignment), TIME DEMAND TYPE, TOPOGRAPHIC PLACE.</p> <p>Exchanged data: Number of delayed service (any delay even for a single stop of the service, is “marking” the service as delayed), with a delay above a specific threshold, and a special count for cancellation, for all services:</p> <ul style="list-style-type: none"> - stopping at one or a set of STOP PLACES; - having specific origin and destination STOP PLACES; - crossing a specific TOPOGRAPHIC PLACE (having STOP PLACES inside it); - running in a specific TIME DEMAND TYPE.

Table A.67 — Use case: EFFICIENCY-012

Use case: EFFICIENCY-012	
Name	Causes for delays
Description	<p>Delay and late service shall be categorized depending on their cause:</p> <ul style="list-style-type: none"> - incidents due to the infrastructure, - due to external events (weather, passenger, traffic congestion, etc.), - due to the operator’s management and decision, - to strikes or social event,
Source (Country)	France, Hungary
Main Actors	All
UC	Service delays and inefficiencies
Raw data needed	<p>TM Raw TM data: DELAY, PT SITUATION CONSEQUENCE, PT SITUATION, SITUATION CAUSE, PT SITUATION CONSEQUENCE SCOPE, ROUTE, SERVICE JOURNEY, SCHEDULED STOP POINT, STOP PLACE (through assignment), TIME DEMAND TYPE, TOPOGRAPHIC PLACE.</p> <p>Exchanged data: Number of delayed service (any delay even for a single stop of the service, is “marking” the service as delayed), with a delay above a specific threshold, and a special count for cancellation, for all services:</p> <ul style="list-style-type: none"> - stopping at one or a set of STOP PLACES; - having specific origin and destination STOP PLACES; - crossing a specific TOPOGRAPHIC PLACE (having STOP PLACES inside it); - running in a specific TIME DEMAND TYPE.

Table A.68 — Use case: EFFICIENCY-013

Use case: EFFICIENCY-013	
Name	Number of full and empty vehicles

Description	Number of full or nearly full, and empty or nearly empty vehicles. On a per line (or origin/destination), per peak/off-peak, and vehicle type base
Source (Country)	France
Main Actors	All
UC	
Raw data needed	<p>TM Raw TM data: VEHICLE MONITORING, TYPE OF VEHICLE DETECTING, VEHICLE DETECTING LOG ENTRY, VEHICLE, DATED VEHICLE JOURNEY, VEHICLE JOURNEY (= > up to LINE), TIME DEMAND TYPE</p> <p><i>The VEHICLE DETECTING should have a full/empty or percentage of occupancy attribute.</i></p> <p><i>A clear definition needs to be provided to manage situation where a VEHICLE is full/empty for only part of the JOURNEY (proposed: for Full, a single occurrence of full is sufficient to tag the service Full, for Empty a threshold, like percentage of empty stops, has to be provided).</i></p> <p><i>Also, note that a Journey may be full at some points and empty at others...</i></p> <p>Exchanged data: Number of full and empty vehicle for each LINE and for a specific TIME DEMAND TYPE or full DAY.</p>

Table A.69 — Use case: EFFICIENCY-014

Use case: EFFICIENCY-014	
Name	Number of cancelled journeys
Description	Report the number of cancelled journeys, and association reasons (number of cancelation per reason type), on a predefined period base.
Source (Country)	France
Main Actors	All
UC	Service delays and inefficiencies
Raw data needed	<p>TM Raw TM data: JOURNEY CANCELLATION, PARTIAL JOURNEY CANCELLATON, PT SITUATION CONSEQUENCE, PT SITUATION, SITUATION CAUSE, OPERATIONAL EVENT, SERVICE JOURNEY</p> <p>Exchanged data: List of cancelled of partially cancelled and associate SITUATION CAUSE/REASON and/or OPERATIONAL EVENT</p>

Table A.70 — Use case: EFFICIENCY-015

Use case: EFFICIENCY-015	
Name	PTO employees age categories distribution

Description	It considers the age category of the employees for each PTO to provide the Public Transport Service in the reference period (typically the year). Goal: Evaluate the employees age distribution in the different categories (drivers, administrative, engineering, etc.) of each PTO respect the reference Service contract. Formulae: Percentage and absolute distribution of employees age for all contractual categories
Source (Country)	Slovenia
Main Actors	PTA, PTO
UC	Public Transport Operator dimensions
	EMPLOYEE need to have an "Age" attribute. TM Raw TM data: EMPLOYEE, ORGANIZATION, OPERATOR Exchanged data: Age distribution of employees (by predefined slices of ages).

Table A.71 — Use case: EFFICIENCY-016

Use case: EFFICIENCY-016	
Name	Maximal number of needed daily drivers
Description	It considers the maximal number of drivers needed for daily operation of the Public Transport Service in the reference period (typically the year). Goal: Evaluate the measured demand respect the reference Service contract. Formulae: Measured number of maximal needed daily drivers
Source (Country)	Slovenia
Main Actors	PTO, PTA
UC	Public Transport Operator dimensions
Raw data needed	TM Raw TM data: DRIVER, EMPLOYEE, ORGANIZATION, OPERATOR, DRIVER ASSIGNMENT, OPERATING DAY Exchanged data: Maximal number of DRIVERS for amongst OPERATIONAL DAYS.

Annex B (informative)

Sample network

To provide a good illustration for the data generation, a sample network has been designed to serve as a basis for examples.

B.1 Network

The network comprises of two lines, which intersect at multiple points (see figure). Line 10 runs from T1 to T3a, calling at SCHEDULED STOP POINTS T1, 1a, 2, 3, 4, T3a. Line 20 runs from T2 to T3b and calls at T2, 1b, 2, 5, 6. SCHEDULED STOP POINT 1a and 1b belong to the same SITE SiteB, while T3a and T3b are located at SITE SiteA.

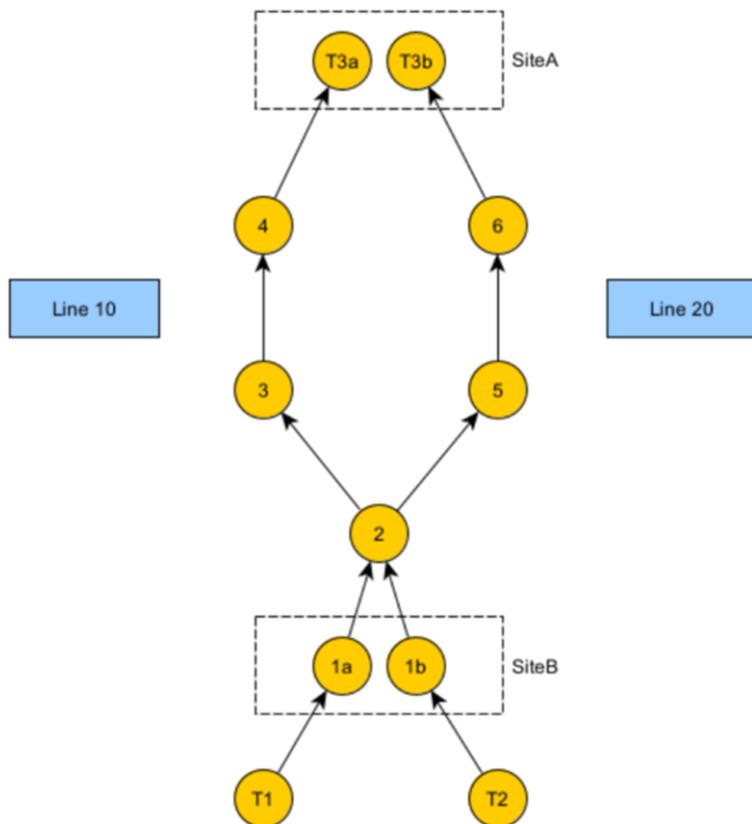


Figure B.1 — Diagram of sample network used for the use cases

To provide real-life demand, the following assumptions were made for the network:

SiteA: industrial area with a typical 3-shift demand

SiteB: Commercial area with shopping characteristics

SCHEDULED STOP POINTS 2: Administrative centre

T1-T2: Large residential areas

SCHEDULED STOP POINTS 3, 4, 5 and 6: Suburban residential areas

B.1.1 Estimated ridership

The following values were made up to simulate the traffic on Line 10:

Line 10	Journey Patterns			
	T1 -> T3a		T3a -> T1	
	Alight	Board	Alight	Board
T1		500	500	
1a	60	80	80	60
2	140	160	160	140
3	70	50	50	70
4	50	30	30	50
T3a	500			500
Total	820	820	820	820

Table B.1 — Sample values for passenger traffic

Through the course of the day, the alighters and boarders in each time band were created according to the following tables

T1->T3a	0-6	6-9	9-12	12-15	15-18	18-21	21-24	Total
Alight								
T1								
1a	5	10	10	20	10	5		60
2		30	60	40	10			140
3		10		20	25	10	5	70
4		10		15	15	5	5	50
T3a	20	200		200		80		500
Total	25	260	70	295	60	100	10	820

Table B.2 — Number of alighters by timeband

T1->T3a	0-6	6-9	9-12	12-15	15-18	18-21	21-24	Total
Board								
T1	25	200	60	165		50		500
1a		10	10	35	20		5	80
2		30		70	40	15	5	160
3		10		15		25		50
4		10		10		10		30
T3a								
Total	25	260	70	295	60	100	10	820

Table B.3 — Number of boarders by timeband

B.1.2 Timetable

Line 10 starts at the following hours:

Direction T1 -> T3a: 4:30, 5:15, 6:00, 6:30, 7:00, 7:30, 8:00, 8:30, 9:00, 10:00, 11:00, 12:00, 12:30, 13:00, 13:30, 14:00, 14:30, 15:00, 16:00, 17:00, 18:00, 19:00, 20:00, 21:00, 22:00

Direction T3a -> T1: 4:50, 5:35, 6:20, 6:50, 7:20, 7:50, 8:20, 8:50, 9:20, 10:20, 11:20, 12:20, 12:50, 13:20, 13:50, 14:20, 14:50, 15:20, 16:20, 17:20, 18:20, 19:20, 20:20, 21:20, 22:20

B.1.3 Operational log

Through 3 days, the following operational events were registered. Delays shorter than 5 minutes were not registered.

T1 -> T3a	2025-03-16	2025-03-17	2025-03-18
4:30			
5:15			
6:00			
6:30			Cancelled due to vehicle breakdown
7:00	Delay 5'		
7:30		Delay 5'	
8:00			
8:30			
9:00			
10:00			
11:00			
12:00			
12:30			
13:00		Delay 10'	
13:30	Delay 5'		
14:00			
14:30			
15:00			
16:00			
17:00		Replaced due to vehicle breakdown	
18:00			Delay 10'
19:00			

20:00			
21:00			
22:00			

Table B.4 — Operational log used for worked examples

Annex C (informative) Version - Model Element

C.1 Version – Model Element

C.1.1 General

IMPORTANT: The **Version** Element is DEPRECATED from NeTEx 2.0 onwards. This, is only for the XML **Version** element (<Version ... >) and not the XML **version** attribute (as in <Tag id="xxx" version="yyy" ... >) whose usage is now emphasised in the XSD.

The use of the VERSION element was just to state descriptive details about a particular version of an individual element. In data exchanges, this information is usually of little interest and it is only the description of the validity of the whole VERSION FRAME that is pertinent. The deprecation of the Version Element is due to its very limited usage, and multiple misunderstandings and misuses encountered in these few usages (version of version, version of profiles, wrong id of version, etc.). The concept of the VERSION object remains but is not explicitly implemented as a **Version** element. Instead, a proper usage of the version attribute is strongly recommended. The version of a dataset remains fully managed and described by the VERSION FRAME and its specialisations. The following lines of this section are maintained only for backward compatibility.

A VERSION is a group of operational data instances which share the same VALIDITY CONDITIONS. A version belongs to a unique VERSION FRAME and is characterised by a unique TYPE OF VERSION; e.g. 'NETWORK VERSION for Line 12 starting from 2000-01-01'.

An instance of a VERSION can be used to describe the reason for a change or set of changes and record further details about it. Often however explicit instances of VERSION will not be created, but rather just be implied by the use of version numbers on instances of ENTITIES IN VERSION.

Classification	Name	Type	Cardinality	Description
::>	::>	<i>DataManagedObject</i>	::>	VERSION inherits from of DATA MANAGED OBJECT
«PK»	id	<i>VersionIdType</i>	1:1	Identifier of VERSION.
	StartDate	<i>xsd:dateTime</i>	0:1	Start date of validity of VERSION.
	EndDate	<i>xsd:dateTime</i>	0:1	End date of validity of VERSION.
«enum»	Status	<i>VersionStatusEnum</i>	0:1	Status of VERSION. See allowed values below.
	Description	<i>MultilingualString</i>	0:1	Description of VERSION.
«enum»	VersionType	<i>VersionTypeEnum</i>	0:1	Fixed version type. See allowed values below.
«FK»	TypeOfVersionRef	<i>TypeOfVersionRef</i>	0:1	Reference to an open TYPE OF VERSION. +v1.1
«FK»	DerivedFromVersionRef	<i>VersionRef</i>	0:1	Reference to a VERSION from which this VERSION is derived. The VersionRef type holds a Ref being of VersionIdType , thus being a version number.

Table C.1 — Version - Element

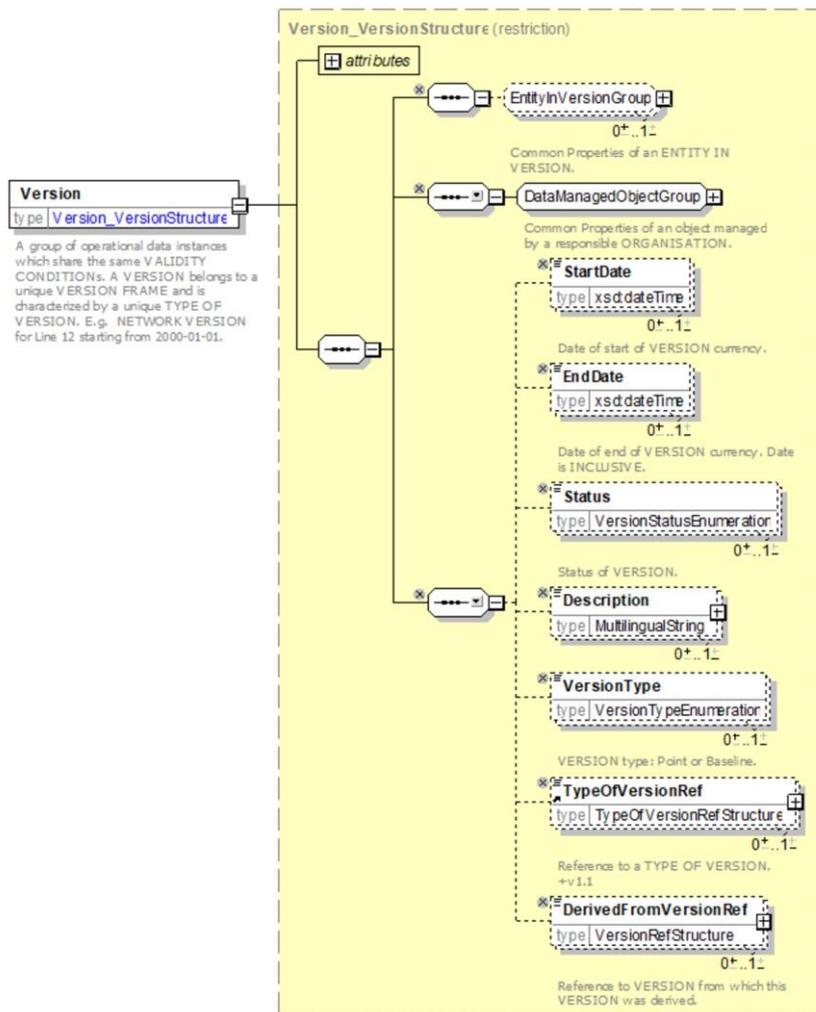


Figure C.1 — Version - XSD

C.1.2 VersionType - Allowed Values

The following table shows the allowed values for **VersionType** (*VersionTypeEnumeration*).

Value	Description
Baseline	This version is comprehensive and comprises all the needed objects to use it.
Point	This version is an update of a baseline version and may only contain a subset of the object from the baseline.

Table C.2 — VersionType- Allowed Values

Bibliography

- [1] Commission Delegated Regulation (EU) 2017/1926 of 31 May 2017 supplementing Directive 2010/40/EU of the European Parliament and of the Council with regard to the provision of EU-wide multimodal travel information services
- [2] RFC 1766. Tags for the Identification of Languages