
**Intelligent transport systems —
Framework for collaborative Telematics
Applications for Regulated commercial
freight Vehicles (TARV) —**

**Part 1:
Framework and architecture**

*Systèmes intelligents de transport — Cadre pour applications
télématiques collaboratives pour véhicules de fret commercial
réglementé (TARV) —*

Partie 1: Cadre et architecture



STANDARDSISO.COM : Click to view the full PDF of ISO 15638-1:2012



COPYRIGHT PROTECTED DOCUMENT

© ISO 2012

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword	v
Introduction.....	vi
1 Scope	1
2 Conformance	1
3 Normative references	2
4 Terms and definitions	3
5 Symbols (and abbreviated terms).....	7
6 General overview and framework	10
6.1 Objective	10
6.2 National variations	10
6.3 Mandatory, optional and cooperative issues	11
6.4 Specification of service provision	11
6.5 Architecture options	11
6.6 Approval of service providers.....	11
7 Concept of operations	12
7.1 General	12
7.2 Statement of the goals and objectives of the system	12
7.3 Strategies, tactics, policies, and constraints affecting the system	12
7.4 Organisations, activities, and interactions among participants and stakeholders.....	12
7.5 Clear statement of responsibilities and authorities delegated.....	12
7.6 Operational processes for the system	12
7.7 Appointment of a approval authority (regulatory)	13
7.8 Role of service provider.....	13
7.9 User.....	13
7.10 Application service.....	13
8 Conceptual architecture framework	14
8.1 General	14
8.2 Actors	14
8.3 Service definition.....	16
8.4 Role model architecture.....	18
9 Conceptual architecture elaboration.....	29
10 Taxonomy.....	37
11 The communications architecture	38
12 Interoperability and the TARV-ROAM 'facilities' layer.....	38
12.1 Interoperability with other cooperative ITS systems	38
12.2 TARV-ROAM 'facilities layer' architecture	41
12.3 ROAM framework and architecture	42
12.4 OSGi® (open services gateway initiative)	50
12.5 TARV-ROAM layered architecture and the role of OSGi®.....	58
12.6 Host management centre (HMC).....	61
12.7 Local data tree (LDT)	63
12.8 TARV supported LDTs	69
12.9 Distributed directory service (DDS).....	71
12.10 Typical use case examples	71

13	Privacy issues	74
13.1	General issues of privacy	74
13.2	Personal privacy	74
13.3	Commercial privacy	75
13.4	Communications privacy	75
13.5	TARV-ROAM privacy	75
14	Quality of service requirements	76
15	Test requirements	76
16	Marking, labelling and packaging	77
17	Declaration of patents and intellectual property	77
Annex A	(Informative) International examples of regulated services	78
Bibliography		107

STANDARDSISO.COM : Click to view the full PDF of ISO 15638-1:2012

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 15638-1 was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*.

ISO 15638 consists of the following parts, under the general title *Intelligent transport systems — Framework for collaborative Telematics Applications for Regulated commercial freight Vehicles (TARV)*:

— *Part 1: Framework and architecture*

The following parts are to be published:

— *Part 2: Common platform parameters using CALM*

— *Part 3: Operating requirements, 'Approval Authority' procedures, and enforcement provisions for the providers of regulated services*

— *Part 4: System security requirements*

— *Part 5: Generic vehicle information*

— *Part 6: Regulated applications*

— *Part 7: Other applications*

Subsequent parts of ISO 15638 will provide definitions for specific TARV application services.

Introduction

Many ITS technologies have been embraced by commercial transport operators and freight owners, in the areas of fleet management, safety and security. Telematics applications have also been developed for governmental use. Such regulatory services in use or being considered vary from country to country, but include electronic on-board recorders, vehicle charging, digital tachograph, on-board mass monitoring, vehicle access monitoring, hazardous goods tracking and e-call. Additional applications with a regulatory impact being developed include, fatigue management, speed monitoring and heavy vehicle charging based on mass, location, distance and time.

In such an emerging environment of regulatory and *commercial applications* (4.15), it is timely to consider an overall *architecture* (4.7) (business and functional) that could support these functions from a single platform within a commercial freight vehicle that operates within such regulations. International standards will allow for a speedy development and *specification* (4.40) of new applications that build upon the functionality of a generic *specification* platform. A suite of standards deliverables is required to describe and define the *framework* (4.20) and requirements so that the on-board equipment and *back office* (4.9) systems can be commercially designed in an open market to meet common requirements of *jurisdictions* (4.24).

This suite of standards addresses and defines the *framework* (4.20) for a range of cooperative telematics applications for *regulated commercial freight vehicles* (4.37) (such as access monitoring, driver fatigue management, speed monitoring, on-board mass monitoring and charging). The overall scope includes the concept of operation, legal and regulatory issues, and the generic cooperative provision of services to *regulated commercial freight vehicles* (4.37), using an on-board ITS platform. The *framework* (4.20) is based on a (multiple) *service provider* (4.39) oriented approach provisions for the approval and auditing of *service providers* (4.40).

This suite of standards deliverables:

- provides the basis for future development of cooperative telematics applications for *regulated commercial freight vehicles* (4.37). Many elements to accomplish this are already available. Existing relevant standards will be referenced, and the *specifications* (4.41) will use existing standards (such as CALM) wherever practicable.
- allows for a powerful platform for highly cost-effective delivery of a range of telematics applications for *regulated commercial freight vehicles* (4.37).
- presents a business *architecture* (4.7) based on a (multiple) *service provider* (4.39) oriented approach.
- addresses legal and regulatory aspects for the approval and auditing of *service providers* (4.40).

This suite of standards deliverables is timely as many governments (Europe, North America, Asia and Australia/New Zealand) are considering the use of telematics for a range of regulatory purposes. Ensuring that a single in-vehicle platform can deliver a range of services to both government and industry through open standards and competitive markets is a strategic objective.

This part of the ISO 15638 provides the overall *framework* (4.20) description and *architecture* (4.7) for TARV, including the detailed *architecture* (4.7) *specification* (4.40) of the facilities layer.

NOTE 1 The definition of what comprises a 'regulated' vehicle is regarded as an issue for National decision, and may vary from country to country. This suite of standards deliverables does not impose any requirements on nations in respect of how they define a regulated commercial freight vehicle.

NOTE 2 The definition of what comprises a 'regulated' service is regarded as an issue for National decision, and may vary from country to country. This suite of standards deliverables does not impose any requirements on nations in respect

of which services for regulated commercial freight vehicles countries will require, or support as an option, but provides standardised sets of requirements descriptions for identified services to enable consistent and cost efficient implementations where implemented.

NOTE 3 *Cooperative ITS (4.14)* applications, in this context, are defined as the use of an in-vehicle *ITS* platform to meet both commercial and regulatory needs from a (functionally) single on-board platform.

STANDARDSISO.COM : Click to view the full PDF of ISO 15638-1:2012

STANDARDSISO.COM : Click to view the full PDF of ISO 15638-1:2012

Intelligent transport systems — Framework for collaborative Telematics Applications for Regulated commercial freight Vehicles (TARV) —

Part 1: Framework and architecture

1 Scope

This part of ISO 15638 provides the following for cooperative telematics applications for *regulated commercial freight vehicles* (4.37):

- a) A *framework* (4.20) for the provision of cooperative telematics application services for regulated commercial freight vehicles;
- b) A description of the concept of operation, regulatory aspects and options and the role models;
- c) A conceptual *architecture* (4.7) using an on-board platform and wireless communications to a *regulator* (4.25) or his agent;
- d) References for the key documents on which the *architecture* (4.7) is based;
- e) Details of the *architecture* (4.7) of the facilities layer;
- f) A taxonomy of the organisation of generic procedures;
- g) Common terminology for the ISO 15638 family of standards.

This part of ISO 15638 is based on a (multiple) *service provider* (4.39) oriented approach.

ISO 15638 has been developed for use in the context of regulated commercial freight vehicles. There is nothing however to prevent a jurisdiction extending or adapting the scope to include other types of regulated vehicles, as it deems appropriate.

NOTE The specific 'approval' procedures for specific application services are a matter for the jurisdiction (4.24) and are outside the scope of this (or any) part of 15638. However approval authorities (4.6) are encouraged to use the guidance of ISO 17000 and ISO/IEC 17065:2012 when developing and implementing such procedures.

2 Conformance

This part of ISO 15638 defines a general architecture (4.7), and has no specific conformance tests defined herein. Some aspects defined within may have conformance tests defined in other parts of ISO 15638.

Conformance declarations for the various parts of a CALM-compliant system shall be based on the relevant CALM-related international standards.

Conformance to any other international standard or specification (4.40) referenced in this part of ISO 15638 shall be ascertained according to the requirements of the referenced deliverable.

Conformance to this part of ISO 15638 is therefore a matter of self declaration of compliance, or by submission to a test house to ascertain that the provisions of the clauses of this part of ISO 15638 have been adhered to.

3 Normative references

The following referenced documents are indispensable for the application 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.

ISO/TR 12859, *Intelligent transport systems — System architecture — Privacy aspects in ITS standards and systems*

ISO 15638-2¹, *Intelligent transport systems — Framework for collaborative Telematics Applications for Regulated commercial freight Vehicles (TARV) — Common platform parameters using CALM*

ISO 15638-3², *Intelligent transport systems — Framework for collaborative Telematics Applications for Regulated commercial freight Vehicles (TARV) — Operating requirements, 'Approval Authority' procedures, and enforcement provisions for the providers of regulated services*

ISO 15638-5³, *Intelligent transport systems — Framework for collaborative Telematics Applications for Regulated commercial freight Vehicles (TARV) — Generic vehicle information*

ISO/TS 15638-6⁴, *Intelligent transport systems — Framework for collaborative Telematics Applications for Regulated commercial freight Vehicles (TARV) — Regulated applications*

ISO 15638-7⁵, *Intelligent transport systems — Framework for collaborative Telematics Applications for Regulated commercial freight Vehicles (TARV) — Other applications*

ISO 21210, *Intelligent transport systems — Communications access for land mobiles (CALM) — IPv6 Networking*

ISO 21217, *Intelligent transport systems — Communications access for land mobiles (CALM) — Architecture*

ISO 21218, *Intelligent transport systems — Communications access for land mobiles (CALM) — Medium service access points*

ISO 24102, *Intelligent transport systems — Communications access for land mobiles (CALM) — Management*

ETSI TS 102 665, *Digital Enhanced Cordless Telecommunications (DECT); DECT access to IP networks*

NOTE 1 Subsequent parts of ISO 15638 will provide definitions for specific TARV application services.

NOTE 2 See Bibliography for references to other parts of ISO 15638 which are in various stages of ballot, but not yet published at the date of publication of this International Standard.

¹ To be published.

² To be published.

³ To be published.

⁴ To be published. Full International Standard approval procedures are in process.

⁵ To be published.

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

NOTE This clause contains all definitions generally used within the ISO 15638 suite of standards.

4.1

applicant

party which has applied for approval as a *service provider* (4.39)

4.2

application service

service provided by a *service provider* (4.39) accessing data from the *IVS* (4.23) of a *regulated commercial freight vehicle* (4.37) via a wireless communications network

4.3

appoint/appointment/appointed

assign officially to take responsibility for a role

4.4

approval

formal affirmation that an *applicant* (4.1) has satisfied all the requirements for *appointment* (4.3) as a *service provider* (4.39)

4.5

approval agreement

written agreement made between an *approval authority (regulatory)* (4.6) and a *service provider* (4.39)

NOTE An *approval agreement* (4.5) recognises the fact that a '*service provider*' (4.39) having satisfied the '*approval authority (regulatory)*' (4.6) requirements for *appointment* (4.3) as a '*service provider*', is *appointed* (4.3) in that capacity, and sets out the legal obligations of the parties with respect to the on-going role of the '*service provider*'.

4.6

approval authority (regulatory)

organisation (usually independent) which conducts *approval* (4.4) and ongoing *audit* (4.8) for *service providers* (4.39)

4.7

architecture

formalised description of the design of the structure of *TARV* and its *framework* (4.20)

4.8

audit

review of a party's capacity to meet, or continue to meet, the initial and ongoing *approval agreements* (4.5) as a *service provider* (4.39)

4.9

back office

generic term for the computing and communication facilities of a *service provider* (4.39) or an *approval authority (regulatory)* (4.6) or *jurisdiction regulator* (4.25)

4.10

basic vehicle data

data that shall be maintained/provided by all *IVS* (4.23), regardless of *jurisdiction* (4.24)

4.11

Controller Area Networking bus

CAN bus

network designed for use in automobiles; it uses a single terminated twisted pair cable; is multi master; maximum signal frequency used is 1 Mbit/sec; length is typically 40M at 1Mbit/sec up to 10KM at 5Kbits/sec; it has high reliability with extensive error checking; typical maximum data rate achievable is 40KBytes/sec; maximum latency of high priority message <120 µsec at 1Mbit/sec

NOTE CAN is unusual in that the entities on the network, called nodes, are not given specific addresses. Instead, it is the messages themselves that have an identifier which also determines the messages' priority. For this reason there is no theoretical limit to the number of nodes although in practice it is ~64.

4.12

certification authority (digital)

organization which issues digital certificates for use by other parties (specifically in the context of communications security)

4.13

cmpn

OSGi® (open services gateway initiative) service platform specifications

[OSGi®]

4.14

cooperative ITS

C-ITS

ITS applications in *regulated commercial freight vehicles* (4.37) for both regulatory and commercial purposes that require the exchange of data between uncontracted parties using ITS stations

4.15

commercial applications

ITS applications in *regulated commercial freight vehicles* (4.37) for commercial (non-regulated) purposes

Example Asset tracking, vehicle and engine monitoring, cargo security, driver management etc.

4.16

condition

set of rules determined by the *jurisdiction* (4.24) to trigger the generation of reports

EXAMPLE Compliance or non-compliance reports, *exception reports* (4.19), condition reports, events, alarms and passage reports.

4.17

core application data

basic vehicle data (4.10) plus any additional data required to provide an implemented *regulated application service* (4.36)

4.18

enrolment

official registration to participate

4.19

exception report

report that is generated according to the *condition(s)* (4.16) in an application, and forwarded to a *jurisdiction* (4.24) by a *service provider* (4.39)

4.20

framework

particular set of beliefs or ideas referred to in order to describe a scenario or solve a problem

4.21**global navigation satellite system****GNSS**

comprises several networks of satellites that transmit radio signals containing time and distance data that can be picked up by a receiver, allowing the user to identify the location of its receiver anywhere around the globe

4.22**global positioning system****GPS**

instantiation of *GNSS* (4.21) controlled by the US Department of Defence

4.23**in-vehicle system (IVS)**

ITS-station and connected equipment on board a vehicle

4.24**jurisdiction**

government, road or traffic authority which owns the *regulatory applications* (4.35)

Example

Country, state, city council, road authority, government department (customs, treasury, transport), etc.

4.25**jurisdiction regulator****regulator**

agent of the *jurisdiction* (4.24) *appointed* (4.3) to regulate and manage TARV within the domain of the jurisdiction

NOTE

May or may not be the *approval authority (regulatory)* (4.6).

4.26**map**

spatial dataset that defines the road system

4.27**on-board unit****OBU**

integrated telematics unit installed on board which provides the specified telematics functionality required for the *IVS* (4.23)

4.28**OSGi® Bundles**

OSGi® components made by the developers

4.29**OSGi® Execution environment**

defines what methods and classes are available in a specific platform

4.30**OSGi® Life-cycle**

application programming interface (API) to install, start, stop, update, and uninstall bundles

4.31**OSGi® Modules**

layer that defines how a bundle can import and export code

4.32**OSGi® Security**

layer that handles the security aspects

4.33

OSGi® Services

connect bundles in a dynamic way by offering a publish-find-bind model for plain old JAVA® objects

4.34

prime service provider

service provider (4.39) who is the first contractor to provide *regulated application services* (4.36) to the *regulated commercial freight vehicle* (4.37), or a nominated successor on termination of that initial contract

NOTE The *prime service provider* is also responsible for maintaining the installed *IVS* (4.23); if the *IVS* was not installed during the manufacture of the vehicle, the *prime service provider* is also responsible for installing and commissioning the *IVS*.

4.35

regulated/regulatory application

approval arrangement utilised by *jurisdictions* (4.24) for granting certain categories of commercial vehicle rights to operate in regulated circumstances subject to certain *conditions* (4.16)

NOTE Each *jurisdiction* may use their own terminology including, but not limited to, permit, application, scheme, concession, exemption, gazettal and notice.

4.36

regulated application service

TARV application service (4.2) that is mandated by a regulation imposed by a *jurisdiction* (4.24), or an option supported by a *jurisdiction*

4.37

regulated commercial freight vehicle

vehicle (often but not always designed to haul commercial freight) that is subject to regulations determined by the *jurisdiction* (4.24) as to its use on the road system of the jurisdiction in regulated circumstances, subject to certain *conditions* (4.16), and in compliance with specific regulations for that class of vehicle

NOTE At the option of jurisdictions this may require the provision of information via *TARV* or provide the option to do so.

4.38

regime for open application management

ROAM

facilities and open application execution and its management for *TARV* systems

4.39

service provider

party which is certified by an *approval authority (regulatory)* (4.6) as suitable to provide regulated or commercial *ITS application services* (4.2)

4.40

specification

explicit and detailed description of the nature and functional requirements and minimum performance of equipment, service or a combination of both

4.41

tamper

conduct towards *IVS* (4.23) or a *service provider's* (4.39) system which is intended to prevent the *IVS* or the *service provider's* system from functioning correctly

4.42

Unified Modeling Language

UML

graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system

NOTE *UML* offers a standard way to write a system's blueprints, including conceptual things such as business processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components, and is standardised as ISO/IEC 19501.

4.43

uniform resource identifier

URI

string of characters used to identify a name or a resource on the Internet

NOTE Such identification enables interaction with representations of the resource over a network (typically the World Wide Web) using specific protocols; schemes specifying a concrete syntax and associated protocols define each URI.

4.44

uniform resource locator

URL

uniform resource identifier [URI (4.43)] that specifies where an identified resource is available and the mechanism for retrieving it

NOTE In popular usage and in many technical documents and verbal discussions it is often incorrectly used as a synonym for *URI* (4.43) (The best-known example of the use of *URLs* is for the addresses of web pages on the World Wide Web, such as <http://www.example.com/>).

4.45

user

individual or party that enrolls in and operates within a regulated or commercial *application service* (4.2)

Example Driver, transport operator, freight owner, etc.

5 Abbreviated terms

For the purposes of this part of ISO 15638 and the ISO 15638 suite of standards deliverables, the following abbreviated terms apply.

API

application programming interface

app

application programme

CALM

communications access for land mobiles

CAD

core application data (4.17)

CAN

controller area network (4.11)

C-ITS

cooperative intelligent transport systems

CDS

charging data services

DDS

distributed directory service

DMT

device management tree

DSRC

dedicated short range communication

FA

interface between the facilities layer and the ITS-S applications entity

[ISO 21217]

FOAM

framework for open applications

[project CVIS]

G

gravitational force

GNSS

global navigation satellite system (4.21)

GPS

global positioning system (4.22)

HMC

host management centre

HMI

human/machine interface

I2I

infrastructure to infrastructure

ID

identity

IETF

internet engineering task force

IN

interface between the access layer and the networking and transport layer

[ISO 21217]

ITS

intelligent transport system

IVS

in-vehicle system (4.23)

JAR

JAVA® archive retrieval (file format)

LDT

local data tree

LDM

local dynamic map

LTE

long term evolution (mobile phone generation after 3G)

MA

interface between the communication and station management entity and the ITS-S applications entity

[ISO 21217]

MF

interface between the communication and station management entity and the facilities layer

[ISO 21217]

MI

interface between the communication and station management entity and the access layer

[ISO 21217]

MN

interface between the communication and station management entity and the networking and transport layer

[ISO 21217]

MS

interface between the communication and station management entity and the security entity

[ISO 21217]

NF

interface between the networking and transport layer and the facilities layer

[ISO 21217]

OMA

open mobile alliance

OB

on-board unit (4.27)

OEM

original equipment manufacturer

OSGi®

open services gateway initiative

RAM

random access memory

ROAM

regime for open application management (4.38)

RSE

roadside equipment

RSI

roadside Infrastructure

SAP

service access point

SF

interface between the security entity and the facilities layer

[ISO 21217]

SI

interface between the security entity and the access layer

[ISO 21217]

SN

interface between the security entity and the networking and transport layer

[ISO 21217]

SOA

service oriented architecture

SSO

single sign-on

TARV

telematics applications for regulated commercial freight vehicles

UML

Unified Modeling Language (4.42)

(ISO 19501)

URI

uniform resource identifier (4.43)

URL

uniform resource locator (4.44)

UTC

universal time coordinated

V2I

vehicle to infrastructure (communication)

V2V

vehicle to vehicle communication

6 General overview and framework

6.1 Objective

This Clause describes a generic *framework (4.20)* for the provision of cooperative telematics *application services (4.2)* for *regulated commercial freight vehicles (4.37)*. Clause 7 provides the general concept of operations for which this *architecture (4.7)* is designed. Clauses 8 and 9 provide a *framework*, role definition and elaboration of the *architecture* at a conceptual level. Clause 10 provides the taxonomy of the *architecture (4.7)* and Clause 11 defines the communications *architecture*. Clause 12 defines the facilities layer and its interoperability.

Annex A provides some informative examples from telematics systems for regulated commercial freight vehicles, cited from Jurisdictions around the world.

6.2 National variations

6.2.1 As stated in the scope, the definition of what comprises a 'regulated' vehicle is regarded as an issue for National decision, and may vary from country to country.

6.2.2 The instantiation of interoperable on-board platforms for regulated commercial freight vehicles with common features is expected to vary from country to country, as will the provision of regulated, or supported, services.

6.2.3 It is possible that some countries will mandate the use of such a platform, others will offer it as an option to meet the requirements of the regulation with minimum administration and paperwork (providing a good business case for operators to fit and use the equipment).

6.2.4 Some countries may implement a single, government operated, controlled, or contracted *service provider* (4.39) which is the single communication manager between the vehicle and the service. Other countries may provide a market based solution with multiple *service providers* competing for the business of vehicle operators.

6.3 Mandatory, optional and cooperative issues

6.3.1 As stated in 6.2.1, the definition of what comprises a 'regulated' service is regarded as an issue for National decision, and may vary from country to country. Further, services may be 'required' by a *regulator* (4.25), or may be supported by a regulator, but not required. (There may for example be a choice between using electronic means to plan, approve and monitor the movement of a hazardous cargo journey, or to use traditional paper request, approval and monitoring).

6.3.2 The *IVS* (4.23) may also support the provision of other commercial *application services* (4.2) that are not required by the *regulator* (4.25).

6.3.3 This suite of standards deliverables does not impose any requirements on Nations in respect of which services for regulated commercial freight vehicles countries will require, or which they will support as an option, but this suite of standards deliverables will provide a generic common *architecture* (4.7) within which countries can achieve their own objectives in respect of *application services* (4.2) for *regulated commercial freight vehicles* (4.37), and provide standardised sets of requirements descriptions for identified services to enable consistent and cost efficient implementations where instantiated.

6.3.4 Cooperative *ITS* (4.14) applications, in this context, is the use of a common on-board platform to meet both regulated and commercial service provision.

6.4 Specification of service provision

Cooperative *ITS* (4.14) applications for regulated commercial freight vehicles (both regulated services and commercial services) are specified in terms of the service provision, and not in terms of the hardware and software.

6.5 Architecture options

Architecturally, it needs to be possible for a vehicle operator to use the services of different *service providers* (4.40) in different geographical areas, or for the provision of different services within the same geographical area. In these circumstances, where there is a market of competing service providers, the most likely solution may be expected to be that the *user* (4.45) will choose a single service provider who will install and maintain the *in-vehicle system* (4.23) into the regulated commercial freight vehicle and deliver all services that the user to which the user chooses to subscribe. In future years however, the in-vehicle system may be a vehicle original equipment *specification* (4.40) option, inbuilt at the time of manufacture of the vehicle, with service provider selection being a subsequent user choice (much as we select an internet service provider today). Other options are possible and should be able to be supported within the conceptual *architecture* (4.7).

6.6 Approval of service providers

As determined by the regulatory *jurisdiction* (4.24), the *service provider* (4.39) may need to be certified by the *regulator* (4.25), and so some form of 'Approval authority (regulatory) (4.6)' role forms an essential part of the *architecture* (4.7), but the role may and will be instantiated differently by different *jurisdictions*.

7 Concept of operations

7.1 General

This Clause describes the characteristics of a proposed system from the viewpoint of an individual who will use that system. Its objective is to communicate the quantitative and qualitative system characteristics to all stakeholders.

This part of ISO 15638 describes the roles and responsibilities of the classes and actors involved in the provision of regulated services for regulated commercial freight vehicles using telematics.

This part of ISO 15638 recognises that there will be variations between *jurisdictions* (4.24). It does not attempt, nor recommend, homogeneity between *jurisdictions*, simply it is designed to provide common standard features to enable equipment of common *specification* (4.40) to be used, and the common features of service provision to be able to be referenced by a jurisdiction in its regulatory and/or legislative regime simply by reference to an International Standards deliverable (requiring it to specify in detail only the particular additional requirements of a jurisdiction).

A 'concept of operations' (CONOPS) generally evolves from a concept and is a description of how a set of capabilities may be employed to achieve desired objectives.

7.2 Statement of the goals and objectives of the system

The overall objective of *TARV* is the assessment, monitoring etc. of regulated commercial freight vehicles to meet the requirements of the jurisdiction within it is operating, using telematics.

This is achieved by the provision of *application services* (4.2) for specific aspects of the control and management of *regulated commercial freight vehicles* (4.37). These services are provided by agreement with the *user* (4.45), and using an approved *service provider* (4.39) to meet the requirements of the jurisdiction using *in-vehicle system* (4.23) with communications capability between the vehicle and the *service provider*, and access to relevant data from the regulated commercial freight vehicle.

7.3 Strategies, tactics, policies, and constraints affecting the system

Strategies, tactics, policies and constraints, and indeed, the services that are regulated as mandatory or optionally supported, may vary from *jurisdiction* (4.24) to *jurisdiction*. (Clause 6 provides detail of the options of such aspects.

7.4 Organisations, activities, and interactions among participants and stakeholders

The classes, attributes and key relationships are described in Clause 8, and are some high level conceptual architectural detail is elaborated in Clause 9. Clause 10 provides the taxonomy of the *architecture* (4.7) and Clause 11 defines the communications *architecture*. Clause 12 defines the facilities layer and its interoperability.

7.5 Clear statement of responsibilities and authorities delegated

Clause 6 describes the high level options and issues. The actors, their responsibilities and authorities are described in Clause 8 below, and the roles are described in Clause 8 and in this Clause (Clause 7).

7.6 Operational processes for the system

The following description of operational processes is at a high abstracted level (above that of any particular application service). Specific services may have additional requirements not described herein, but guidance and *specification* (4.40) for some aspects may be obtained from ISO 15638-6 and ISO 15638-7.

7.6.1 Service requirements definition

A jurisdiction passes legislation/regulation to require or support the provision of a particular *application service* (4.2). The legislation/regulation needs to provide clear and unambiguous definition of what is required.

7.7 Appointment of an approval authority (regulatory)

The jurisdiction creates or *appoints* (4.3) an authority to approve and *audit* (4.8) the process. The structure of that authority is a matter for the jurisdiction and it may be a separate appointed organisation, or a department of the *jurisdiction*. (4.24). Within the context of this part of ISO 15638, it is the actor 'role' of 'approval authority' that is important, not its structure, ownership or business model.

An *approval authority (regulatory)* (4.6) may only preside over the instantiation and operation of one particular *application service* (4.2), or may preside over the instantiation and operation of some or all *application services* for regulated commercial freight vehicles (at the discretion of the jurisdiction).

The *approval authority (regulatory)* (4.6) will approve *service providers* (4.40), *IVS* (4.23), and will provide *audit* (4.8) as described in Clause 6 above, in accordance with the requirements of the *jurisdiction* (4.24).

NOTE: The TARV approval authority is described throughout as the '*approval authority (regulatory)* (4.6) to clearly distinguish it from a certification authority which issues digital certificates (especially in the context of communications security).

7.8 Role of service provider

The *service provider* (4.39) shall offer to users to provide the application service needed to meet the requirements of the legislation/regulation of the *jurisdiction* (4.24). The *service provider* (4.39) may also provide additional commercial services so long as they do not impair, impede or interfere with, the provision of the enrolled regulated application service(s).

7.9 User

The *user* (4.45) is most usually the operator of the regulated commercial freight vehicle, but in some cases may be the driver. He/she will enrol with the jurisdiction to have his service provided automatically by wireless communications. He/she will appoint an approved *service provider* (4.39) to provide the *regulated application service* (4.36) for the regulated commercial freight vehicle (or driver where appropriate).

It is the responsibility of the operator of the vehicle to enrol, and to have its vehicle equipped to enable it to provide the service (regardless of whether the *user* (4.45) of the service is the vehicle operator or the driver of the vehicle). So long as operator uses certified *service providers* (4.40), installers and maintainers, the operator may then assume that the application service will be provided in accordance with the legislation/regulations.

The *user* (4.45) will be responsible to pay any fees for the provision of the service agreed with the *service provider* (4.39), to the *service provider*. The means by which this is achieved is a subject for the commercial marketplace and is outside the scope of this part of ISO 15638.

7.10 Application service

The *service provider* (4.39) will provide the *regulated application service* (4.36) (and may also provide other commercial services so long as they do not impair, impede or interfere with, the provision of the enrolled regulated application service(s)).

In some *jurisdictions* (4.24) the *service provider* (4.39) application service, with authorisation from the *user* (4.45), may also collect permit fees, licence fees and other fees required to be paid to the *jurisdiction*, and forward these to the *jurisdiction*. The commercial provisions for such transactions are a matter between the jurisdiction and the *service provider* and are outside the scope of this part of ISO 15638.

8 Conceptual architecture framework

8.1 General

Clause 7 above provided the generic concept of operations which these actors and classes enact in order to provide the application service(s). In order to specify a generic *framework* (4.20) standard of the ITS service platform, this framework standardisation deliverable identifies core actors and classes as described in 8.2 to 8.4 below, which are described as elements which are independent of any specific application.

8.2 Actors

This part of ISO 15638 defines a role model where the roles and responsibilities of four key actor classes are defined to provide an entity known as an 'application service':

- the 'Jurisdiction(s)' (4.24);
- the 'Users' (4.76);
- the 'Service Provider(s)' (4.40);
- the 'Approval authority (regulatory)' (4.6).

The 'role model' provides the general attributes and the responsibilities of the parties. These aspects are described within this part of ISO 15638. Figure 1 illustrates a conceptual role model *architecture* (4.7) for TARV.

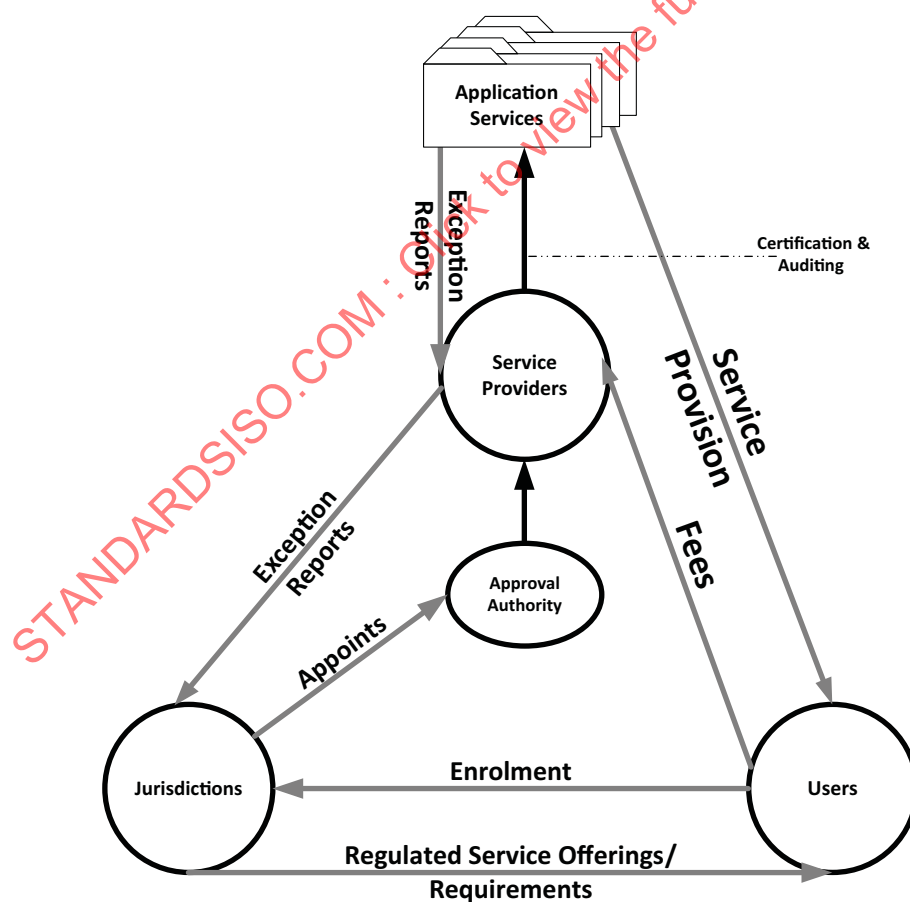


Figure 1 — Role model conceptual architecture

However the basic rules of procedure are also required and shall be as defined in ISO 15638-3 (TARV – Operating requirements, 'Approval authority' approval procedures, and enforcement provisions for the providers of regulated services) which provides common generic *specifications* (4.41) for these issues in respect of *regulated application services* (4.36). Individual service provision aspects shall be as provided in ISO 15638-5 (TARV – Essential and core application data', ISO 15638-6 (TARV regulated applications) and ISO 15638-7 (TARV- Other applications).

Using a *UML* (4.42) approach, the relationships between the classes can be represented as shown in Figure 2.

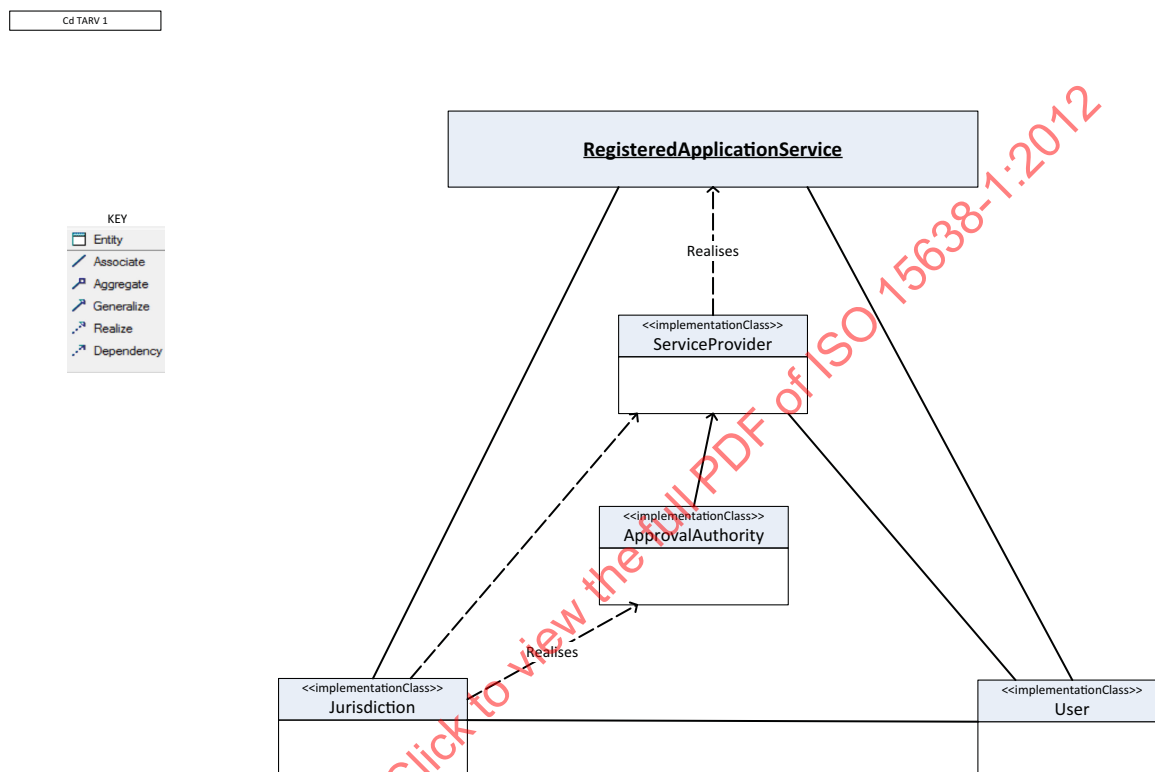


Figure 2 — UML model overview of the classes

The ISO 15638 family of standards deliverables comprises:

- 15638-1 TARV-Framework and architecture (this part)
- 15638-2 TARV-Common platform parameters using *CALM*
- 15638-3 TARV-Operating requirements, 'Approval Authority' procedures, and enforcement provisions for the providers of regulated services
- 15638-4 TARV-System security requirements (*expected completion TS 2012, IS 2013*)
- 15638-5 TARV-Essential and core application data
- 15638-6 TARV-Regulated applications (*expected completion TS 2012, IS 2013*)
- 15638-7 TARV-Other applications.

Clause 9 provides more detailed architectural elaboration of the role models.

The communications aspects of the concept can also be graphically represented as in Figure 3 below.

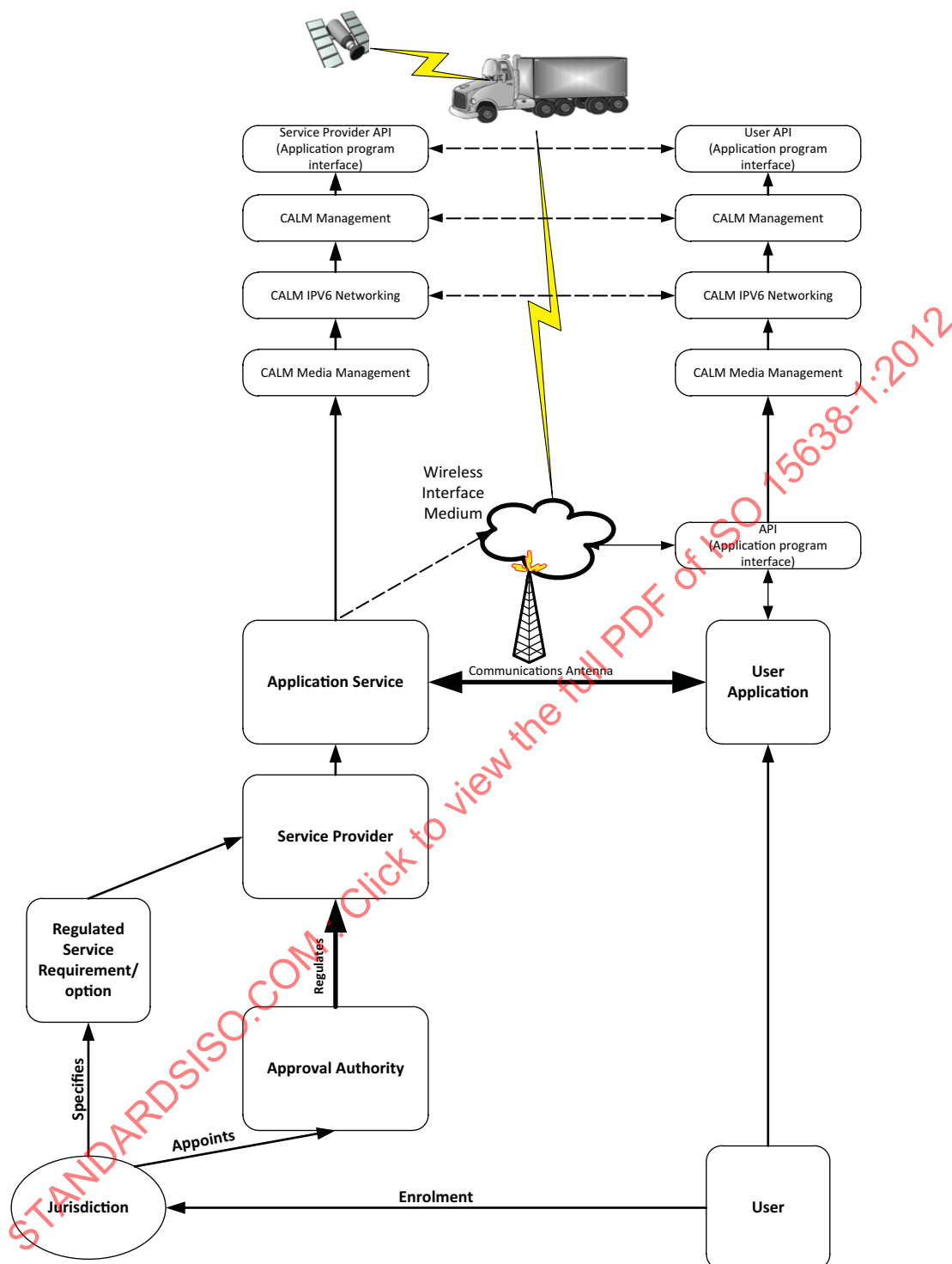


Figure 3 — Service provision and its communications

The communications *architecture* (4.7) is defined in greater detail in ISO 15638-2.

8.3 Service definition

For any specific service, regulated or commercial, mandatory or optional, the definition of the service that a *Service Provider* (4.40) has to support shall be specified and provided, including a given service level. Specific application service examples can be found in ISO 15638-6 and ISO 15638-7.

The service definition for each application service comprises:

- a) a clear description of the service provided and its inputs, outputs and results;
- b) basic vehicle data content and quality that an *IVS* (4.23) has to deliver;
- c) *core application data* (4.17) content to meet the requirements of the jurisdiction;
- d) any additional application specific data content for the provision of that particular service;
- e) service elements (such as “retrieve data from *OBU*”, “map data to a map with access conditions”, “report non-compliance”, etc.);
- f) rules for the *approval* (4.4) of *IVSs* (4.23), ‘*application service providers* (4.40)’ and ‘*application services* (4.2)’.

Unlike commercial services, regulated services, be they required or optional, are a special case, because the jurisdiction has to trust a *service provider* (4.39) to ensure that the *regulated application service* (4.36) is properly provided in accordance with the regulation. ‘*Rules for approval* (4.4)’ of *service providers* and *IVSs* (4.23) are therefore required, and there will in most instantiations be a ‘*Approval authority (regulatory)* (4.6)’ to regulate the service provision. ISO 15638-3 (TARV—Operating requirements, ‘Approval Authority’ procedures, and enforcement provisions for the providers of regulated services) provides both ‘*generic approval* (4.4) *procedures*’ and ‘*Basic service requirements for Service Providers*’ that are generic and independent of a specific application.

(for example, shall have secure processing, shall have *map* (4.26)-matching capability, shall keep records in a transparent and auditable way, shall read *IVS* (4.23) data at defined intervals etc.). ISO 15638-3 provides *specifications* (4.41) for these issues.

ISO 15638-2 shall provide *specifications* (4.41) for communications aspects and the communications *architecture* (4.7) is defined in greater detail in ISO 15638-2.

ISO 15638-4 shall provide *specifications* (4.41) for security issues.

Application service provision shall in part provided by ‘*core application data* (4.17)’ , as defined in ISO 15638-5, and in part by requirements specific only to a particular regulated service application. *Basic vehicle data* (4.10) is calculated and stored by all *IVSs* (4.23), regardless of *jurisdiction* (4.24). Additional specific data requirements of the *jurisdiction*, together with the *basic vehicle data* are known as the *core application data* (4.17).

Examples of *Basic vehicle data* (4.10) include aspects such as: a unique vehicle identifier, a unique *IVS* (4.23) identifier, time and location data. ISO 15638-5 shall define the common data concepts for vehicle information known within the suite of ISO 15638 standards deliverables as *basic vehicle data* (4.10). Some of this data is permanent for the lifetime, or a contract span, other information is dynamic and journey specific (such as location, freight and driver information). ISO 15638-3 shall define aspects such as identifying tasks, and who has responsibilities for them, such as: shall have secure data processing and storage, shall be physically and environmentally robust, shall be able to communicate with the *service provider* (4.39) etc.

Most services will also require data that is specific to the application. These are specific data concepts to enable the provision of a particular *application service* (4.2). For example, data from a tachograph for remote tachograph monitoring. ISO 15638-6 shall provide high level *specifications* (4.41) for these issues, in respect of commonly agreed generic requirements for *regulated applications* (4.35), but is designed to operate in an environment where the jurisdiction determines its exact requirements and the application *service provider* (4.39) determines and designs the system that provides the application service to the *user* (4.45).

ISO 15638-7 shall provide generic *specifications* (4.41) for the standardised *specification* of future regulatory and commercial *application services* (4.2) which can share access to the on-board platform.

8.4 Role model architecture

8.4.1 General

This Clause considers the roles of the actors defined in 8.2 and their interrelationship in greater detail, and their relationship to the provision of the applications service(s).

8.4.2 Jurisdictions

The *jurisdiction* (4.24) is the body that has official power to make legal decisions and impose regulations. How this operates will vary from country to country according to their constitution or legal structure. Countries may have a single *jurisdiction*, or may delegate such authorities to their constituent states, or, as in the case of Europe, independent states may concede part of their independent National jurisdiction to a common jurisdiction union (e.g. European Union) to achieve common goals and interoperability within common *conditions* (4.16), while retaining independent jurisdiction in other matters.

Regardless of the differences between *jurisdictions* (4.24), what is common for the purposes of this part of ISO 15638, is the concept that at any specific location, and time, there is a single jurisdiction that has official power to make legal decisions and impose regulations in respect of the regulation of commercial freight transport.

While the specific *regulated application services* (4.36) that are offered or imposed on *regulated commercial freight vehicles* (4.37) will vary from *jurisdiction* (4.24) to *jurisdiction*, the generic requirements to offer or impose such *regulated application services* are largely similar.

The *jurisdictions* (4.24) are the owners of the *regulated applications* (4.35). These may be required by regulation, or may be offered by a *jurisdiction* as an option to demonstrate compliance to a regulation, according to the choice of the *jurisdiction* and the regulations that it enforces.

Within the context of this part of ISO 15638, the role of the *jurisdiction* (4.24) is to:

- define the regulated application services (4.36);
- define if they are mandatory or optional;
- pass legislation to determine and regulate;
- manage and regulate the provision of the regulated application services.

Without prescribing the domestic arrangements within any *jurisdiction* (4.24), the management and regulation of the provision of the *regulated application services* (4.36) can be architecturally described as:

- laws and regulations;
- adopted standards;
- adjudication and mediation;
- auditing;
- *approval* (4.4) of equipment;
- *approval* of *service providers* (4.40);
- *approval* of *application services* (4.2);
- trusted third party;

involving five further classes/subclasses of actors in addition to the jurisdiction:

- The jurisdiction;
- The service provider;
- The *IVS* (4.23) equipment installer (subclass);
- The *IVS* equipment maintainer (subclass);
- The *approval authority (regulatory)* (4.6);
- The user.

Single entities may perform the roles of multiple classes of actor (a *service provider* (4.39) for example may also install and maintain the *IVS* (4.23)). Other actors will also be embraced within these key roles (such as a communications provider), but these may be regarded as additional subclasses that support one of the key actor roles.

At the specific *jurisdiction* (4.24) level this *architecture* (4.7) can be elaborated in greater detail, and specifically to the instantiation of *TARV* within that *jurisdiction*. For the purposes of this part of ISO 15638, however, abstracting to the level of Figure 1 and Figure 2, provides a generic common *framework* (4.20) that can be instantiated with variations from *jurisdiction* to *jurisdiction*, yet remain a generic common *framework* (4.20) to which equipment can be built and *application services* (4.2) specified.

8.4.3 Service provider(s)

A *service provider* (4.39), within the context of this part of ISO 15638, can be described as a party which is certified by the *approval authority (regulatory)* (4.6) as suitable to provide regulated or commercial *ITS* services.

The *service providers* (4.40) may be provided or subcontracted by the *jurisdiction* (4.24), but are more likely to be by the use of third party commercial companies which provide *ITS* services. It is expected that in many cases, and particularly in the early years, the *service providers* will also install the *IVS* (4.23) into the users' vehicles, although in the future the *IVS* (4.23) platforms may be an option for installed equipment at manufacture, or may be mandated equipment at manufacture, according to the *jurisdiction*, and separation of the provision and maintenance of equipment and service provision may also be possible if allowed by the *jurisdiction*.

The *service provider* (4.39) will provide the application service, interacting wirelessly with the vehicle to collect relevant data from the *IVS* (4.23), process the data and provide the *jurisdictions* (4.24) with *exception reports* (4.19) and any other relevant and required data, according to the requirements of the application(s), and provide relevant data to the *user* (4.45).

As stated in 6.2, some countries may implement a single, government operated, controlled, or contracted *service provider* (4.39) which is the single communication manager between the vehicle and the service. Other countries may provide a market based solution with multiple *service providers* competing for the business of vehicle operators. In instantiation, the role of the *service provider* is more complex where there may be multiple *service providers*, and even more complex where a *user* (4.45) may use multiple *service providers* for different services. The *architecture* (4.7) defined in this part of ISO 15638 has at least to encompass these three possibilities. Fortunately, the *architecture* for the most complex situation is also appropriate with its simpler alternatives.

This, the most complex of the options to instantiate, is sometimes called the *CDS* - Charging Data Services model. The model is simple, even if the instantiation options may make instantiation more complex. In this model *service providers* (4.40) compete in an open market to provide data based services for which they charge the *user* (4.45) a fee for the service provision. See Figure 4. The functionality of the model to provide *application services* (4.2) still operates where there the *service provider* is mandated by the *jurisdiction* (4.24) (and there is no open market), and operates regardless of whether the application service is an offered option or is a requirement mandated by regulation.

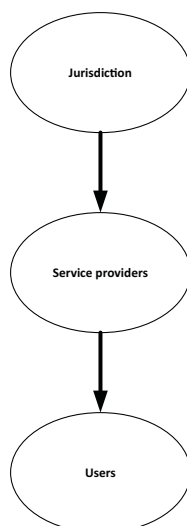


Figure 4 — CDS model

What is important to understand is that in this model, which fits clearly within the models shown in Figure 1 and Figure 2, responsibility for collection of data rests with *service providers* (4.40).

The *service providers* (4.40) will most frequently charge the users for the service provided. It is also possible in this model that the *service provider* also collects fees required by the regulation from the *user* (4.45) (such as fees for permits, road use payment, additional policing, possibly even fees for violations) on behalf of the *jurisdiction* (4.24) and forward these payments to the *jurisdiction*, thus minimising the costs of the *jurisdiction* to maintain its regime. It is also possible that in some *jurisdictions* for some *regulated applications* (4.35) the *jurisdiction* may bear the cost of providing the service.

The responsibility for determination of such fees (or the scope within which the *service provider* (4.39) can set its fees) has to rest with the jurisdiction and will depend on the legislation and regulation imposed by the *jurisdiction* (4.24).

Multiple *service providers* (4.40) can also transmit raw or collated data to the back-office of the *jurisdiction* (4.24), or to transport departments of the *jurisdiction*, for which they may expect to receive fees from the authority, or it may be a *condition* (4.16) of their licence. In these cases privacy aspects need to be carefully considered by both the *jurisdiction* and *service provider* (4.39).

Approval (4.4) and auditing of *service providers* (4.40), to meet the requirements of the *jurisdiction* (4.24) is needed to guarantee required (and clearly defined) levels and quality of service.

In the open market context, *service providers* (4.40) are third party commercial companies who provide ITS services based on the applications. Where a *user* (4.45) employs a single *service provider* (4.39) to provide all of the regulated services (and possibly some additional commercial services) it may be expected that the *service providers* may also install and maintain IVSs (4.23) in the user's vehicles. However, in a situation where the IVS is provided as part of the original equipment *specification* (4.40) of the vehicle, or in a situation where a *user* elects to employ multiple *service providers* either to provide different services, or in different geographic areas, or a combination of both, we also need to introduce the concept of the 'equipment installer' and the 'equipment maintainer' as essential actor roles in the *architecture* (4.7), even if they are technically sub-classes of the *service provider*.

NOTE The choices may not necessarily be only those of the *user* (4.45). In some *jurisdictions* (4.24), in order to maintain control and assure quality of service to its *regulated application services* (4.36), and because of issues of liability, a *jurisdiction* may limit the option of the *user* to that of a single or limited number of *service providers* (4.40).

Whatever the combination used, once contracted, the *service provider* (4.39) will be responsible to collect relevant data from the IVS (4.23), process the data and provide the *jurisdiction(s)* (4.24) with reports according

to the requirements of the *application service* (4.2) provided (as specified by the jurisdiction in respect of *regulated application services* (4.36)).

In most cases, although collecting data from the vehicle constitutes a crucial part of the service provision, the end results are sorted and evaluated by and at the *service provider* (4.39) and communicated to the *jurisdiction* (4.24) (as demanded by the regulation) and to the *user* (4.45) (as agreed in the contract between the *service provider* and the user).

While it is desirable that *service providers* (4.40) are also permitted to provide commercial services to users using the same *IVS* (4.23), it will be necessary for the *service provider* to gain approval of the *regulator* (4.25) or its *approval authority (regulatory)* (4.6) to assure that the provision of a non-regulated service does not affect the quality of the service provision of any regulated services.

The technical requirements for a *service provider* should be performance based. That is, the *jurisdiction regulator* (4.25) defines required outputs and it is up to each potential service provider to establish, to the satisfaction of the *jurisdiction regulator*, or its *approval authority (regulatory)* (4.6), that its equipment and related back-office systems deliver the required outputs. The *jurisdiction regulator* (4.25), and its *approval authority (regulatory)*, should not normally specify the particular equipment and systems required. Thus, competing companies whose equipment and systems differ significantly may be certified, as long as they deliver the required outputs.

This enables *service providers* (4.40) to have the flexibility to take full advantage of innovative, cutting edge *ITS* technologies when designing and developing their equipment and systems, and to evolve those systems over time as technology advances. Coupled with market competition between *service providers* (where permitted by the *jurisdiction* (4.24)), this flexibility will ensure that the technology keeps pace with world-wide advances in broader *ITS* technologies.

8.4.4 Application services

Application services (4.2), whether regulated or commercial, therefore need clear definition in terms of the requirements on the *service provider* (4.39). Guidelines and example *specifications* (4.41) are provided in 15638-6 and 15638-7.

It is important to understand the difference between *approval* (4.4) of the *application service* (4.2) provider and *approval* of the application service. Where there is only one *service provider* providing the service across the *jurisdiction* (4.24), or where a *user* (4.45) is tied to a single *service provider* for the provision of all of its services, the difference may at times seem somewhat academic, however there is a functional difference of significance.

The jurisdiction will want the provision of a *regulated application service* (4.36) to be identical for all users, regardless of *service provider* (4.39). It has only two ways to achieve this. It can appoint only one *service provider* to provide an application service, compelling all users to use the one *service provider* (in which case a *user* (4.45) may have to contract with multiple *service providers* (4.40) to provide different *application services* (4.2) unless the jurisdiction opts to offer a monopoly service provider). Alternately, there may be multiple *service providers* (4.40). If there are multiple *service providers* it is crucial that the both the requirements of its *application service* (4.2) and the net result are identical, regardless of *service provider*, even if harvested in slightly different ways. Another way to achieve this would be for the jurisdiction to develop the application service, and provide the same software to multiple *service providers* to use. In this circumstance, responsibilities for any defects in the software, and all upgrades or modifications, will have to lie with the jurisdiction.

ISO 15638-3 provides *specification* (4.40) for three simple generic *application services* (4.2):

- basic vehicle data (4.10);
- core application data (4.17);
- stored data.

ISO 15638-3 shall provide common requirements for operating requirements, *appointment* (4.3), election and *approval* (4.4) and these operating service commands. ISO 15638-5 shall provide common core *specification* (4.40) for *regulated application services* (4.36) that *jurisdictions* (4.24) may elect to implement. The *specification* in ISO 15638-5 therefore provide *jurisdictions* with a way to ensure that the service provided and received for these *application services* (4.2) is homogenous across its regime. However, *jurisdictions* or their *appointed* (4.3) *regulator* (4.25) or approval authority (4.6) retain responsibility to ensure that the quality of service provision meets their requirements and is consistent from service provider (4.39) to *service provider*. Standards assist by providing common requirements specification, but that alone does not ensure consistent instantiation, and it is the responsibility of the jurisdiction and its agents- the *approval authority (regulatory)* function and *jurisdiction regulator* (4.25) function - to ensure quality and consistency, however it is organisationally arranged within the *jurisdiction*.

For regulated services, both the inputs and outputs, together with the process requirements to provide the service, need to be specified in a way that is independent of any *service provider* (4.39) or *IVS* (4.23) technology. Process requirements refer to the IT system used in, for example, the collection, processing, data storage, data reporting, security and quality management procedures. The *application service* (4.2) provider system shall have sufficient transfer capability in its specified communication coverage area, and sufficient storage and processing capacity to support the number of *IVSs* for which it has been certified, so the minimum *specifications* (4.41) for these requirements need also to be defined for each application service.

It shall not be allowed to provide the destination IPv6 address of a *jurisdiction* (4.24), *prime service provider* (4.34) or *application service* (4.2) provider, intended to be used for the receipt of data, to the party requesting data. Data issued from the ITS-station of the *IVS* (4.23) shall only be provided to a predetermined IPv6 address, and shall in no circumstances whatsoever be provided directly to an address specified by the party requesting the data.

NOTE By responding to a command by sending the response (data) only to a predetermined IPv6 address, the possible phishing and other unauthorised third party access to the data is significantly reduced.

8.4.5 The IVS equipment installer

This is the actor who installs the *IVS* (4.23), into the vehicle and connects it to additional equipment that is required, so that it is able to perform the application service.

If this is part of the original equipment *specification* (4.40) for the vehicle, the *IVS* (4.23) equipment installer will be the vehicle manufacturer or his agent.

Where the *IVS* (4.23) is not part of the original equipment, the equipment installer is likely to be a *service provider* (4.39) or his agent, particularly in a situation where the market model is where a *user* (4.45) selects a single *service provider* (or is required to do so). In this environment each *service provider* offers and installs their own type of *IVS* (4.23) and has the freedom to offer different market models to recoup the cost of the equipment and its installation (similar to those *conditions* (4.16) which operate in the mobile telecommunications and satellite television markets).

In a situation where the *user* (4.45) is able to, and elects to, use multiple *service providers* (4.40), the *IVS* (4.23) equipment installer is likely to be a third party commercial company. In these circumstances, it will be up to the jurisdiction to establish a regime that ensures effective quality control, and multi-equipment and system functionality, as such a regime will depend on the nature of the particular regulations for that *jurisdiction*. (4.24)

In all circumstances where the *IVS* (4.23) is not part of the original equipment it is expected that these equipment installers will in most *jurisdictions* (4.24) have to be registered with, and approved by, the *approval authority (regulatory)* (4.6).

The *IVS* (4.23) equipment installer has the role not only to install the *IVS* communications equipment but to connect it to other equipment required in order to deliver the application service. For example, in the case of remotely monitoring an electronic tacograph, to connect the tacograph into the *IVS*; in the case of on-board weigh in motion monitoring, connecting that equipment to the *IVS*; etc., and to test the functionality of the installed equipment, and that where multiple equipment is connected, that all regulated services can be provided without detriment of one because of another.

In order to maintain control of the regime, it would seem that it has to be the *IVS* (4.23) equipment installer who is held accountable by the *approval authority (regulatory)* (4.6) for providing the data to application service(s) to the required quality, and the only way that this can be practically achieved is through the service provider(s). Therefore, the installers of any other equipment have to be responsible to the *IVS* equipment installer that their equipment functions properly (and architecturally are therefore a sub-class of the *IVS* equipment installer) and, architecturally, the *IVS* equipment installer has to be responsible to (and is therefore architecturally a sub-class of) the *service provider* (4.39).

While the requirements of the *application service* (4.2) are determined by the *jurisdiction* (4.24), the *jurisdiction* also certifies, approves and *appoints* (4.3) service provider(s) and holds them accountable for the provision of the application service. The *jurisdiction* may decide that it also has to approve *IVS* (4.23) equipment providers, or it may leave this function to the *service provider* (4.39) which it will hold to account, and give the *service provider* freedom (possibly within some limits) as to how he controls his subcontractors.

8.4.6 The IVS equipment maintainer

Once installed, the *IVS* (4.23) equipment has to be maintained. Functionality and capabilities have to be checked from time to time, and the equipment may have to be recalibrated and recertified from time to time in accordance with the regime imposed by the *jurisdictions* (4.24).

A number of business models for this can be envisaged. Maintenance may be a service provided by the service provider; it may be provided by the equipment installer; it may be provided by the vehicle maintainer; it may be provided by the vehicle inspector used for vehicle safety test *approval* (4.4); etc.

The regime allowed will depend on how the jurisdiction best believes that its regime can be implemented and maintained, and will vary from *jurisdiction* (4.24) to *jurisdiction*.

Regardless of the business model operating within a particular *jurisdiction* (4.24), as with *IVS* (4.23) equipment installers, the *IVS* equipment maintainer can also architecturally be considered as a sub-class of the *service provider* (4.39).

8.4.7 Approval authority (regulatory)

Clause 6.5 expounded that if third party *service providers* (4.40) are to provide a service determined by the jurisdiction to users, the *jurisdiction* (4.24), and for that matter the *users* (4.76), need to be assured that this service is being properly provided to the regime and requirements of the jurisdiction's regulation. The *service provider* will therefore need to be certified by the regulator, and so some form of '*Approval authority (regulatory)* (4.6)' forms an essential component of the *architecture* (4.7) (even where this function is in practice carried out by the staff of the jurisdiction).

A '*Approval authority (regulatory)* (4.6)' would most commonly be expected to be an independent organisation which certifies the '*Service providers* (4.40)', and ensures that the level of service provided by the *service providers* is maintained, although *jurisdictions* (4.24) of course have the right to make other arrangements for approval and *audit* (4.8). The concept of the (usually independent) '*Approval authority (regulatory)* (4.6)' appointed (4.3) by the jurisdiction is used throughout this *framework* (4.20) and *architecture* (4.7).

Approval (4.4) refers to the confirmation of certain characteristics of an object, person or organisation. In this context, *approval* (4.4) applies to both the *service providers* and the *IVS* (4.23) for which requirements need to be formulated. These requirements need be described as tests to be passed. Each requirement leads to a verdict (passed or failed) on which the *approval* is based. While the ISO 15638 series of standards prescribe the generic requirements for appointing an *approval authority (regulatory)* (4.6), the ISO 15638 series of standards do NOT prescribe the specific requirements to achieve *approval*, nor its procedures nor pass criteria, nor evaluation methods, which are deemed to be within the provenance of each *jurisdiction*, and not a matter for these International Standards.

8.4.8 Certification authority (digital)

Organization which shall issue digital certificates for use by other parties, particularly in the context of communications and online security.

8.4.9 Service provider approval

This is the process where an organisation is certified as being able to competently complete the tasks to be fulfilled in the *regulated application(s)* (4.35).

The prime role of the *approval authority (regulatory)* (4.6) is therefore, on behalf of the *jurisdiction* (4.24), and to its regime to:

- consider candidates to be service providers (4.40);
- test and approve that the service provider can meet the requirements necessary to provide the application service;
- approve their business model in relation to charging users (where required by the jurisdiction);
- approve and approve the service provider;
- determine the duration of the approval (4.4) and renewal options and requirements.

Approval (4.4) has a significant business impact. Based on the *approval* (4.4), it will be decided which companies will be *appointed* (4.3) as *service providers* (4.40). Such procedures therefore need to be clearly and unambivalently defined by the *jurisdiction* (4.24).

While *approval* (4.4) provides assurance that a service provider meets *approval* requirements at a point in time, a process of on-going *audits* (4.8) is also required to ensure that the *service provider* (4.39) continues to maintain the minimum level of service in accordance with the *approval* agreement.

The *audit* (4.8) requirements comprise a variety of aspects which include operational, technical and financial capabilities. The process of *audit* (4.8) is specific to the *regulated application* (4.35). But the generic common objectives of the *audit* (4.8) function form the second set of requirements for the *approval authority (regulatory)* (4.6), which are to:

- support the policy objectives to the various legislation and requirements;
- monitor compliance by *service providers* (4.40) with the standard and the *approval agreement* (4.5);
- ensure that information provided by the *service provider* is reliable, compete and accurate;
- assist in determining the integrity of information provided by the *service provider*;
- enhance transparency, integrity and public credibility of the *regulated applications* (4.35).

8.4.10 Application service approval

In addition to the *approval* (4.4) of the *application service* (4.2) provider, each application service, whether regulated, or unregulated, shall need to be tested and certified by the *approval authority (regulatory)* (4.6) to assure that:

- a) The system provides the *application service* (4.2) and its data consistent with its *specification* (4.40) and documentation;
- b) The documentation is adequate;
- c) The provision of the *application service* does not adversely impact the provision of other *regulated application services* (4.36).

8.4.11 in-vehicle system (IVS) approval

Having assured that the *service provider* (4.39) is capable to provide and is certified to provide the *application services* (4.2), the *approval authority (regulatory)* (4.6) has also to:

- type approve' the *IVS* (4.23), or if performance-based requirements are in place, perform tests to assure compliance with those standards.
- provide a regime to test and provide assurance that *IVS* (4.23) equipment is capable and properly installed in order to provide the *application services* (4.2).

These should be seen as two functionally separate tasks.

Where an *IVS* (4.23) takes the form of a discrete *OBU* (4.27), it can be 'type approved' using an independent test house. This is more complex in the case of *OEM* installed equipment, which will have to be certified as part of the vehicle type approval tests.

In respect of approving that approved equipment has been installed correctly, the *jurisdiction* (4.24)/*approval authority* (regulatory) (4.6) has a number of ways that it can do this, either by designing specific installation tests directly or assigning that role as a responsibility of a *service provider* (4.39). That decision is to be made by the *jurisdiction* (4.24) and is not defined in this part of ISO 15638.

8.4.11.1 *IVS* type approval

In terms of in-vehicle platform (*IVS* (4.23)) *approval* (4.4), this refers to processes intended to determine if the *IVS* meets minimum standards to assure the required quality.

8.4.11.1.1 *VS* instantiated as an *OBU*

In the case where the *IVS* (4.23) is an independent functioning *on-board unit* (4.27) (*OBU*) it may be viewed as a single product, independent of the vehicle into which it is fitted. It shall be tested in a testable environment where its' functionality can be tested separately from the functionality and performance of any equipment connected to it in order to provide data for the performance of an application service.

8.4.11.1.2 *IVS* instantiated not as an *OBU*

If the *IVS* (4.23) is part of the original equipment of the vehicle it is likely that there will not be a single *OBU* (4.27), but that the functionality will be provided, at least in part via the *CAN bus* (4.11) and/or from similar equipment disbursed around the vehicle. For example, the *GNSS* (4.21) data and compass function will most probably be obtained from the vehicles satellite navigation system; accelerometer data and multi-axis gyroscope from the electronic drive/stability control, etc.

In this event, the *IVS* (4.23) *approval* (4.4) will have to be integrated into the overall vehicle *approval* (4.4).

8.4.11.1.3 *IVS* attributes

The functionality of the *IVS* (4.23) is a computing device with six key attributes:

- central processing unit;
- data storage means;
- data input means;
- connectivity means to/from auxiliary equipment;
- communications means;
- power supply.

Each function needs specific tests as to fitness of purpose.

8.4.11.1.3.1 Central processing unit

The *IVS* (4.23) shall be able to prove that it is able to perform the program of operations required in order to fulfil regulated service provision. This normally implies the combination of

- a processor;
- volatile memory (*RAM/DRAM/SRAM* etc.);
- recognised operating system (e.g. *LINUX®*).

Functionality tests for such systems are widely available and easily devised. The speed of the processor shall be adequate enough to perform the *regulated application service* (4.36). By today's typical computer performance standards, these demands are not high and can be easily demonstrated to be satisfied.

NOTE Providers of in-vehicle platforms that may perform multiple functions in the vehicle in addition to regulated services may be advised to use high performance processors, but this should not be a requirement for the provision of currently envisaged regulated services.

Volatile memory shall be adequate to handle data processing of multiple *regulated applications* (4.35). Since non-volatile memory will also be present, by standards of typical computer performance at the time of the development of this part of ISO 15638, these demands are not high and can be easily demonstrated to be satisfied.

The testing of the central processing unit shall be completely independent of any envisaged application service.

8.4.11.1.3.2 Data storage means

The *IVS* (4.23) shall have a means of non-volatile bistable data storage that can retain the stored information even when not powered (such as hard disc, flash memory etc.).

8.4.11.1.3.3 Data input means

The *IVS* (4.23) shall have a means to receive inputs both from auxiliary equipment, and from its communications capability (in order to receive and process instructions from the service provider)

8.4.11.1.3.4 Connectivity means to/from auxiliary equipment

The *IVS* (4.23) shall have multiple interfaces to connect with auxiliary equipment using standard physical interfaces (USB2, RS232, RS422 etc.) (or in the case of *OEM* installation, access to the *CAN bus* (4.11)).

In the case of a stand-alone *OBU* (4.27), the equipment required to provide the '*basic vehicle data* (4.10)' as specified in ISO 15638-5 shall be provided within the *IVS* (4.23). (*GNSS* (4.21), accelerometer, multi-axis gyroscope, altimeter, clock, compass, etc., and probably a megapixel camera/video)

In the case of *OEM* installation the *IVS* (4.23) shall be provided with access to the vehicle generic data as specified in ISO 15638-5. (from *GNSS* (4.21), accelerometer, multi-axis gyroscope, altimeter, clock, compass, etc., and probably a megapixel camera/video) or shall provide such functionalities as defined in ISO 15638-5 that are not available to it elsewhere.

8.4.11.1.3.5 Communications means

The *IVS* (4.23) shall have, or have access to, one or more wireless means to communicate with the *service provider* (4.39). ISO 15638 architecturally envisages the communications link using the *CALM* protocols. The *CALM* protocols shall be as defined in ISO 21217 Intelligent transport systems — Communications access for land mobiles (*CALM*) — Architecture; ISO 21210 Intelligent transport systems — Communications access for land mobiles — IPv6 Networking; ISO 21218 Intelligent transport systems — Communications access for land

mobiles (*CALM*) — Medium service access points: and ISO 24102 Intelligent transport systems — Communications access for land mobiles (*CALM*) — *CALM* management, which shall form the basis of communications and networking. This shall enable the *service provider* to use any or multiple of the common means to communicate with a vehicle (UMTS/GSM, 5GHz (802.11p WAVE), 60 GHz, European or Japanese *DSRC*, infra-red, mobile broadband, satellite, etc.)

Ideally the communications would be left to the choice of the *service provider* (4.39), but in practical terms, a *jurisdiction* (4.24) may elect to determine the choice of communications technology. For example it may specify the use of satellite communications in very remote areas; or *CALM* M5 if there are multiple trailers which need to communicate with the tractor or where the unit has also to support other V2V communications; or UMTS/GSM where it requires access at any time or to also support eCall; etc.).

Jurisdictions (4.24) will have to pay special care when making such determinations because of the long term implication of their impact.

Conformance tests for communications equipment should be a combination of the normal communications test regime for the selected media combined with conformance tests for *CALM*. See ISO 15638-2.

8.4.11.1.3.6 Power supply

Normally, an in-vehicle platform (*IVS* (4.23)) will draw its electricity from the vehicle power supply. However, systems will need physically protected independent power supply(ies) in the event of the disconnection of the vehicle power supply (for example in the event of an accident), a combination of independent power supply and non-volatile memory to prevent attempts to overcome/outwit the system, and, where required, to obtain information from the vehicle where the vehicle power supply has been intentionally removed (such as during service or vehicle lay-up, or in the event of a collision disconnected automatically as a safety measure).

It may generally be assumed that the vehicle is not operating if there is no power supply functioning in the vehicle (however the vehicle may be on-tow, or piggy-back on another vehicle, or immediately post collision).

The means of achieving the power supply requirements should be a matter for the *service provider* (4.39), however the requirements should be specified by the *jurisdiction* (4.24). Aspects specified by the *jurisdiction* should be easy to demonstrate and include:

- availability of power to *IVS* (4.23) when vehicle is in operation (normally this requirement should be 100%);
- number of hours the *IVS* (4.23) can actively function when vehicle power supply is not available;
- number of hours the *IVS* (4.23) can remain in a stand-by state.

8.4.12 Other aspects

8.4.12.1 Thick/thin client

In designing *regulated applications* (4.35) the *jurisdiction* (4.24) needs to consider whether the design imposes consequences on the nature of the client/*service provider* (4.39) system. If the system requires significant processing to be undertaken on board, a so called 'thick client' system will be required and this will impact the cost of the *IVS* (4.23), and is also likely to make the system more difficult to upgrade. Where possible, system *specifications* (4.41) should be such to enable the *service provider* to decide where calculations are made. This enables the possibility of a 'thin-client' implementation in the vehicle. Wherever possible the nature of the implementation should be determined by the system provider, taking into account any requirements of the *jurisdiction*.

8.4.13 Users

It is important to understand clearly who is the '*user*' (4.76) of the system.

Firstly, it is important to be clear that as the objective of *TARV* is the provision of *application services* (4.2), the *user* (4.45) is the *user* of the application service, rather than any other aspect of the regulated commercial freight vehicle.

There are four possibilities as to who is the *user* (4.45) of the application service:

- owner of the vehicle;
- operator of the vehicle;
- driver;
- owner of the freight.

The owner of the vehicle is usually the person or organisation that has registered the vehicle with the jurisdiction's vehicle registration system. But depending on the regulations of the *jurisdiction* (4.24), which will vary around the world, the person/organisation registering the vehicle may or may not be the actual owner, and could also be a lessee, or the vehicle keeper.

A further complication is that the owner of the vehicle may not be the operator of the vehicle since the owner may lease out or rent out the vehicle, or leave the operation of the vehicle to a third party. Complication is compounded because there are also drivers [who may be the object of the regulated application] who are "owner-operators" (sometimes known as "own-account drivers"). However the most common scenario is the driver employed by or under contract to an operator (sometimes called a motor carrier).

The operator of the vehicle is the party that has the direct interest in the movement of the vehicle, both economically and physically. It is the operator who has given the instruction for the vehicle to take to the road; who has determined the destination; and in the case of a regulated commercial freight vehicle, who has determined the route; and who has obtained any permits that are required. In respect of any regulated services where fees are to be paid, it would be the operator who contracts with the *service provider* (4.39) and makes such payments. It would appear logical therefore that the operator of the vehicle is considered to be the *user* (4.45).

However, for some regulated services it is the driver who is the object of the *regulated application service* (4.36) (for example monitoring driver hours), and it is the driver who has to take the responsibility to ensure that the vehicle is not overloaded. It is the driver of the vehicle who personally provides the transport service and therefore causes the road usage by driving.

But in many cases the driver is not the owner of the vehicle but an employee of the vehicle operator. Various drivers may drive one vehicle during a period, and it is not uncommon for long-haul trucks to have two drivers on board who take it in turns to drive while the other rests. Further, in general, the individual driver is not the economic beneficiary of the transport service.

The owner of the freight benefits from the transport service. The owner of the freight is charged for the service according to the contract held with the transport *service provider* (4.39), but normally has no influence on the transport service itself. The owner of the freight usually has no influence on the choice of vehicle configuration or on the detailed route taken.

Architecturally, this could be very messy. So some simplification and clarity is introduced.

The 'user' may in most circumstances be the operator of the vehicle, but in other circumstances, be the driver of the vehicle. *Regulated application* (4.35) system *specifications* (4.41) must therefore specify and define who the *user* (4.45) of the *regulated application* system is deemed to be.

It is true to say that the 'user' is most commonly the operator of the regulated commercial freight vehicle.

In cases where the *user* (4.45) is defined as the driver, the operator of the regulated commercial freight vehicle should determine in the contract employing the driver that the driver will use the in-vehicle platform to

undertake all *regulated application services* (4.36) including driver specific services, and that the driver will provide all information in his possession that are required by the *regulated application services* (4.36).

As the driver is driving because he has been instructed to do so by the operator, any requirements that relate specifically to the driver being provided with an *application service* (4.2) (such as reporting tachograph data or other driver related data required by the regulator) may assume that he is performing this duty as part of the fulfilment of his contract with the vehicle operator to drive the vehicle on any specific journey. The 'prime' *user* (4.45) of the *application service* (4.2) is therefore always the vehicle operator, and the driver is a secondary *user*, even if any regulatory action, if taken, may be taken directly against the driver (much as he is also required to obey speed limits, drive in a safe manner, and could be prosecuted directly if he is in violation).

Users may choose to enrol into a voluntary application or may be required to enrol in a mandatory application, as determined by the *jurisdiction* (4.24), and this may vary from *jurisdiction* to *jurisdiction*.

Upon the *enrolment* (4.18) of an application, the users will engage a *service provider* (4.39) to start operations under the enrolled application and pay any required fees to/through the *service provider*.

8.4.14 Application service provision

Application services (4.2) are the means by which the *service provider* (4.39) meets the requirements of the regulation imposed by the *jurisdiction* (4.24), or for commercial services, to meet the objectives laid down in that *service specification* (4.40).

Application services (4.2) can have a variety of service offerings, from access to safety to charging, as determined by the *jurisdiction* (4.24), and may vary from *jurisdiction* to *jurisdiction*.

Applications can be either voluntary options or mandatory, as determined by the *jurisdiction* (4.24), and may vary from *jurisdiction* to *jurisdiction*.

NOTE For voluntary applications, the *jurisdictions* (4.24) will generally need to identify or create incentives for the users to participate. This would most typically show a business benefit to the *user* (4.45) who elects the electronic communications option (for example by relieving them of manual administrative procedures and paperwork).

The elaboration of these roles, or classes, is expanded in Clause 9 below.

Figure 3 showed an illustration of the provision of an *application service* (4.2) and its communications and management stack using the concepts of the CALM Standards that enable service provision over different physical wireless communications media.

Clause 9 which provides further explanation of the communications management aspects of Figure 3, and ISO 15638-2, provide more detailed elaboration and explanation of the communications *architecture* (4.7). Clause 12 provides an overview of the 'facilities' layer that sits on top of the communication stack and helps to provide data interoperability and reuse, and to manage applications and enable dynamic real time loading of new applications.

9 Conceptual architecture elaboration

Clause 7 above summarised the concept of operations that the *architecture* (4.7) serves to enable. This Clause provides more detailed elaboration and explanation of the conceptual *architecture* (4.7).

Figure 1 provided an illustration of the 'role model conceptual *architecture* (4.7)' for the provision of *regulated application services* (4.36). Figure 3 provided an illustration of 'service provision and its communications'.

Figure 2 provided a 'UML (4.42) use case model overview of the classes'.

This can be elaborated by expanding Figure 2 into Figure 5.

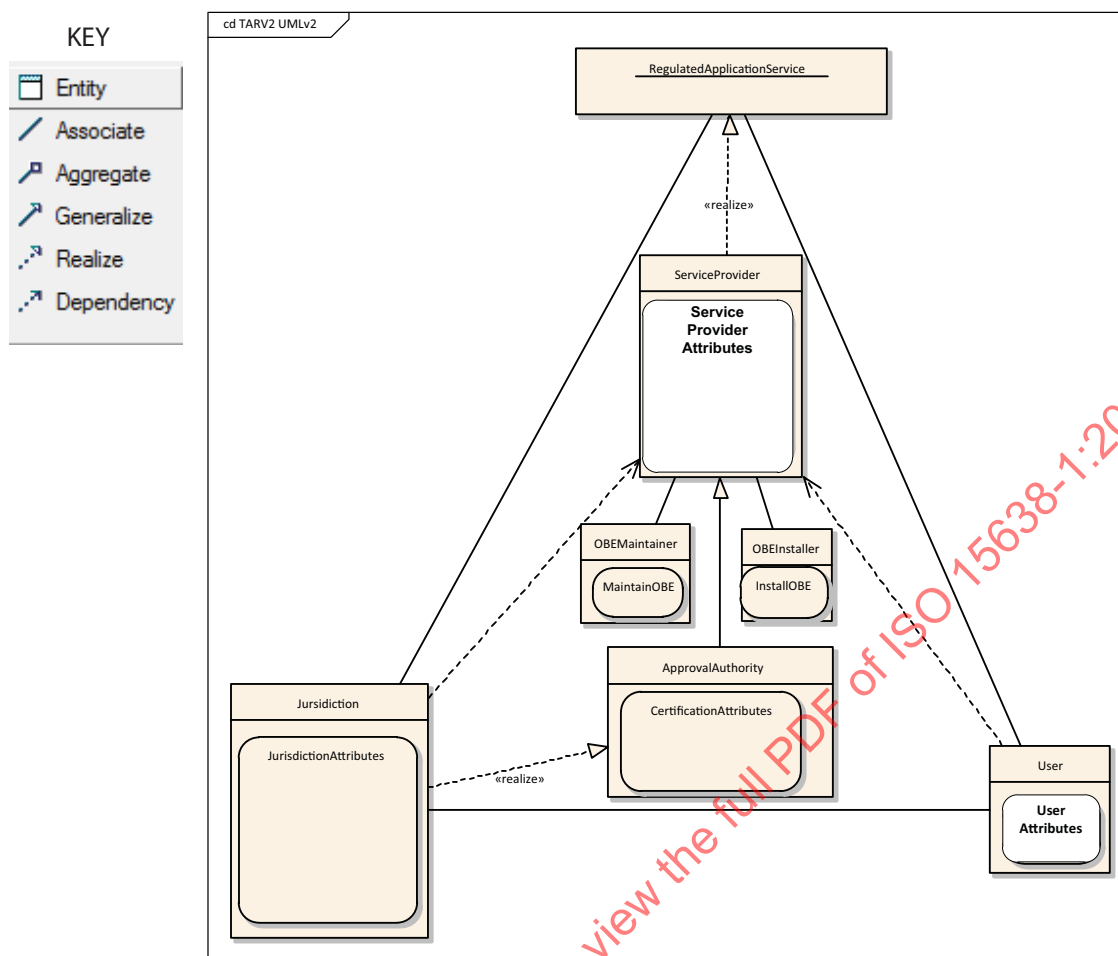


Figure 5 — UML expanded use case model of the classes

An example of the classes and their key attributes are shown in Figure 6. Although it should be noted that there will be some variations to this model due to variations in the regime of different *jurisdictions* (4.24).

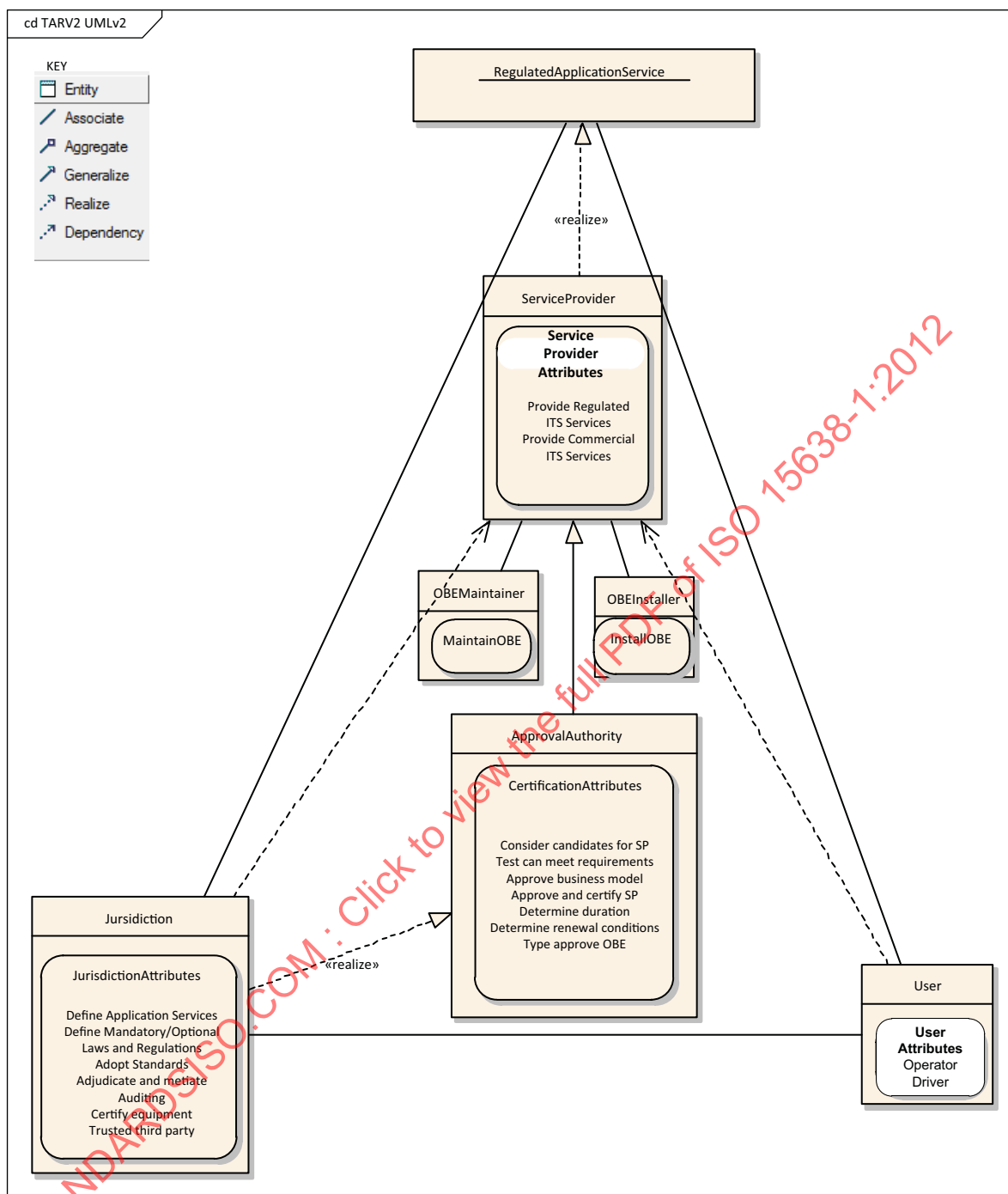


Figure 6 — UML use case model of the classes and key attributes

The illustrative figures can be formalised as a *UML* (4.42) interaction diagram as shown in Figure 7.

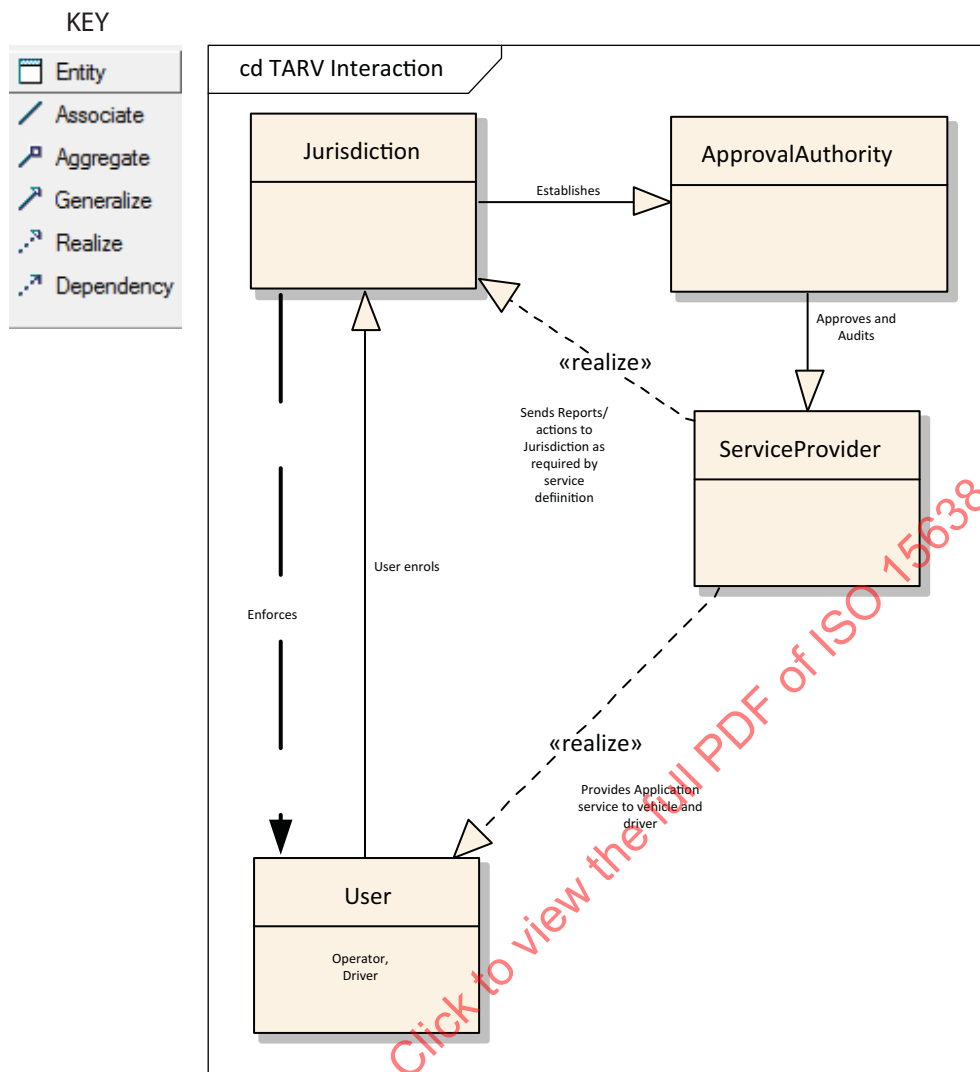


Figure 7 — UML interaction diagram for TARV

While the particular detail of communication sequences will vary from *application service* (4.2) to *application service*, a high level conceptual view of the sequence of operation is shown in Figures 8 and 9.

Figure 8 shows the commercial sequence, while Figure 9 shows an example of the transactional sequence of activity.

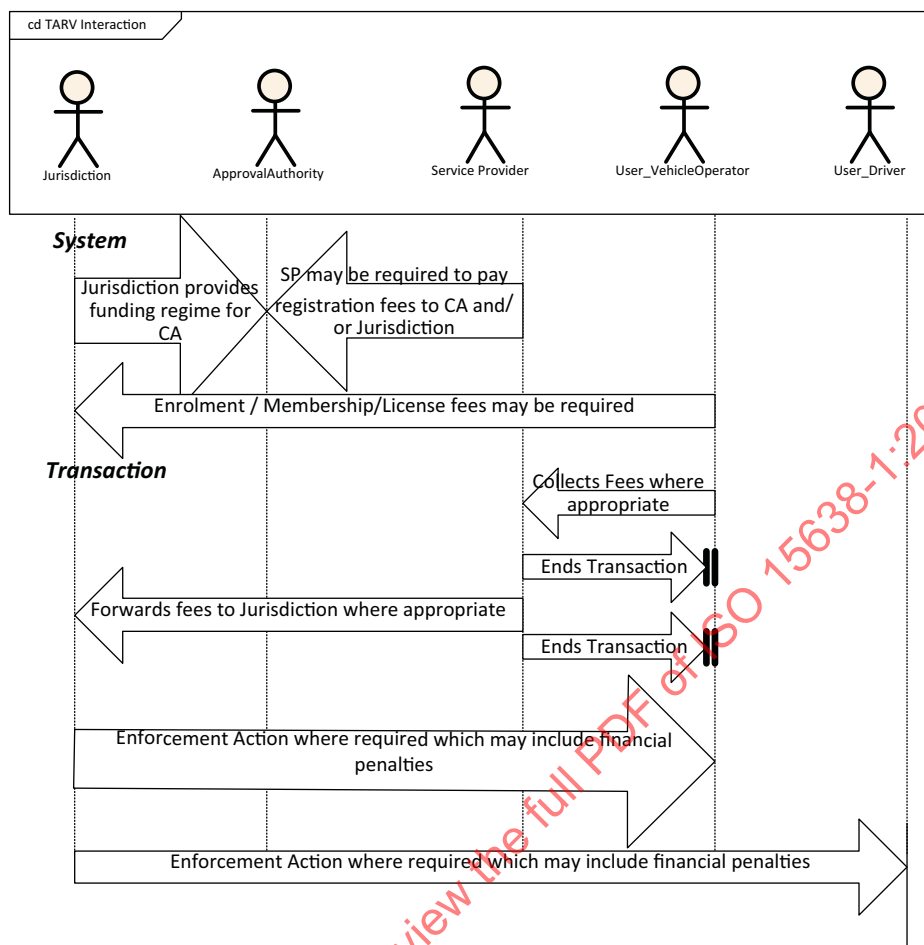


Figure 8 — UML transaction diagram showing the commercial aspects of application service provision

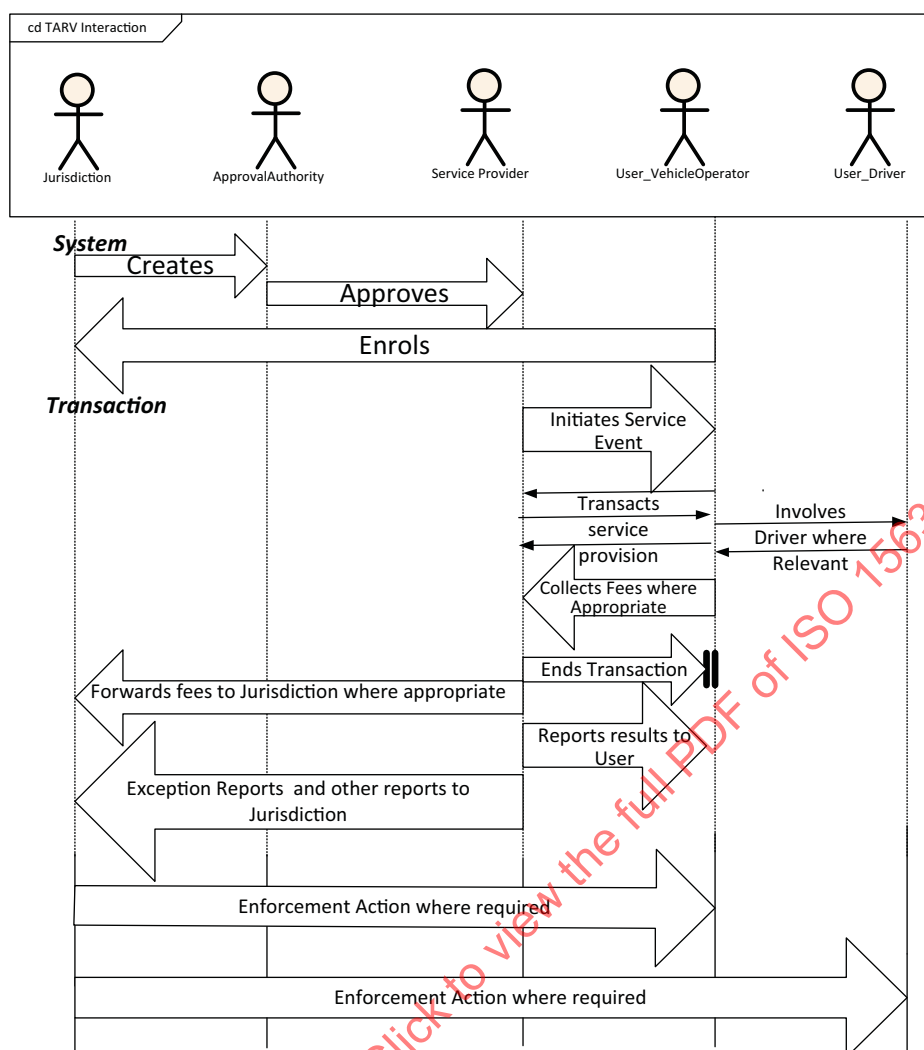


Figure 9 — UML activity diagram showing the transaction sequence of service provision

In respect of the *architecture* (4.7) for the physical communication, the reader is referred to the *specifications* (4.41) defined in ISO 15638-2, which is in accordance with ISO 21217, *CALM Architecture*, together with the *CALM* standard for the specific media that is being used. The reader is therefore referred to these documents for detail of these aspects. A summary of the *ITS station architecture* (4.7), is shown in Figure 10, which is reproduced from ISO 21217, and provides an illustration of examples of wireless links employing various access technologies.

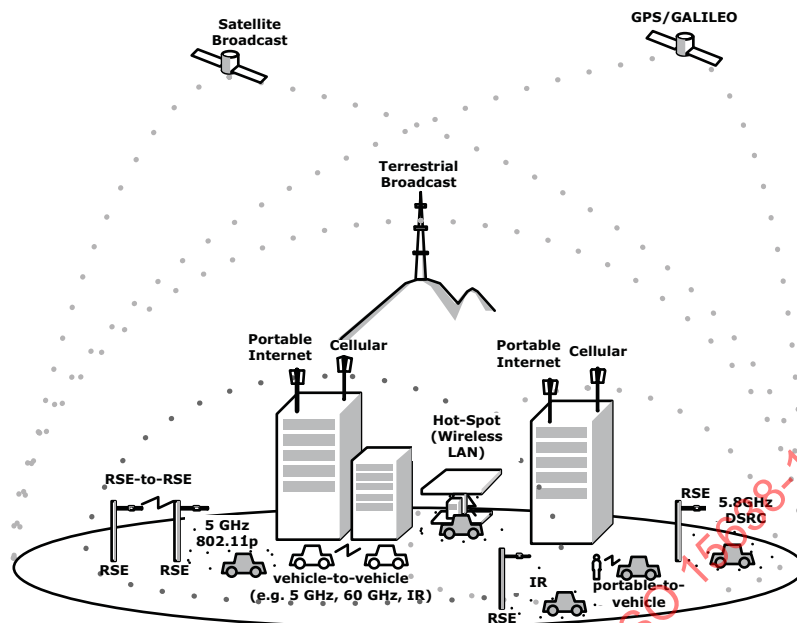


Figure 10 — Examples of wireless links employing various access technologies
(Source ISO 21217)

An overview of *ITS station communications architecture* (4.7), which is reproduced from ISO 21217, is provided in Figure 11.

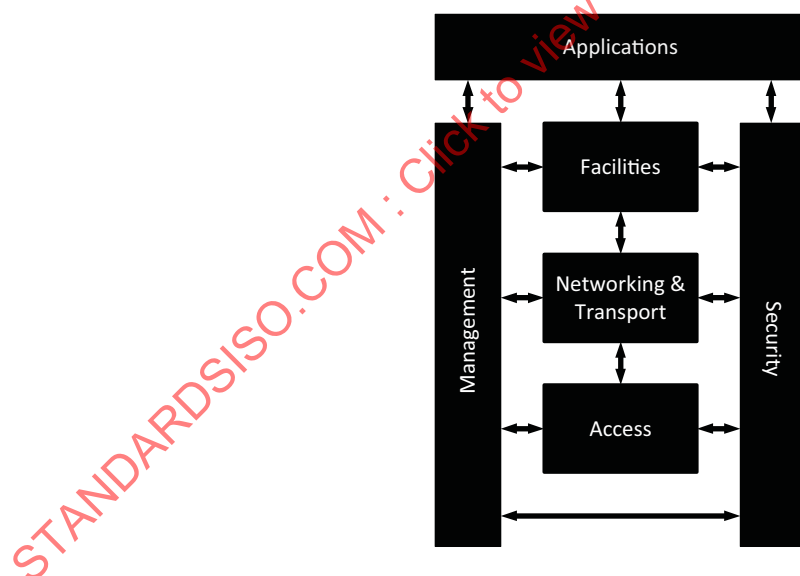


Figure 11 — ITS station architecture high level view
(Source: ISO 21217)

Figure 12 shows the general *ITS station reference architecture* (4.7), including interfaces between the various blocks with informative details. Such interfaces may be partly non-observable and thus non-testable service access points (SAPs), or observable and testable interfaces. (see Clause 5 for full text of abbreviations)

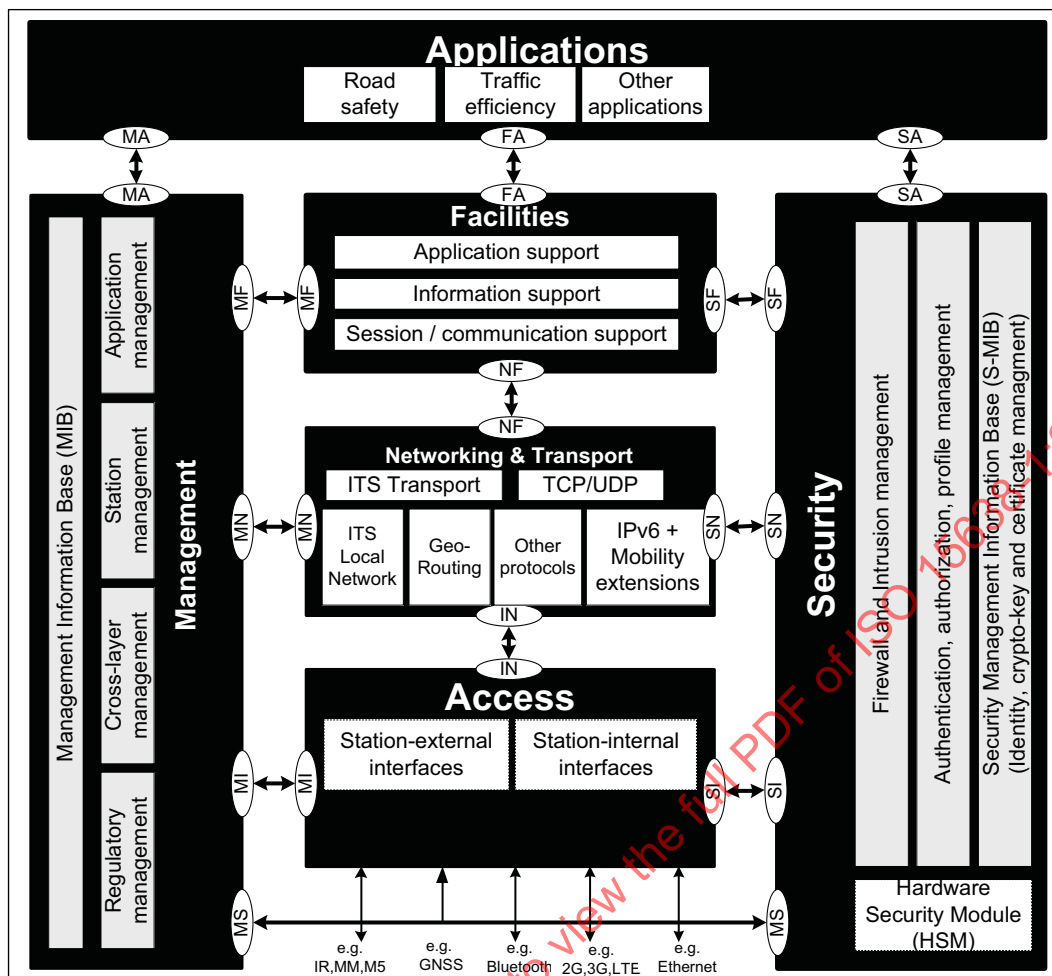


Figure 12 — ITS station architecture detailed view
(Source: ISO 21217)

For further details and explanation, see ISO 21217.

10 Taxonomy

A taxonomy of the organisation for *TARV* is provided in Figure 13.

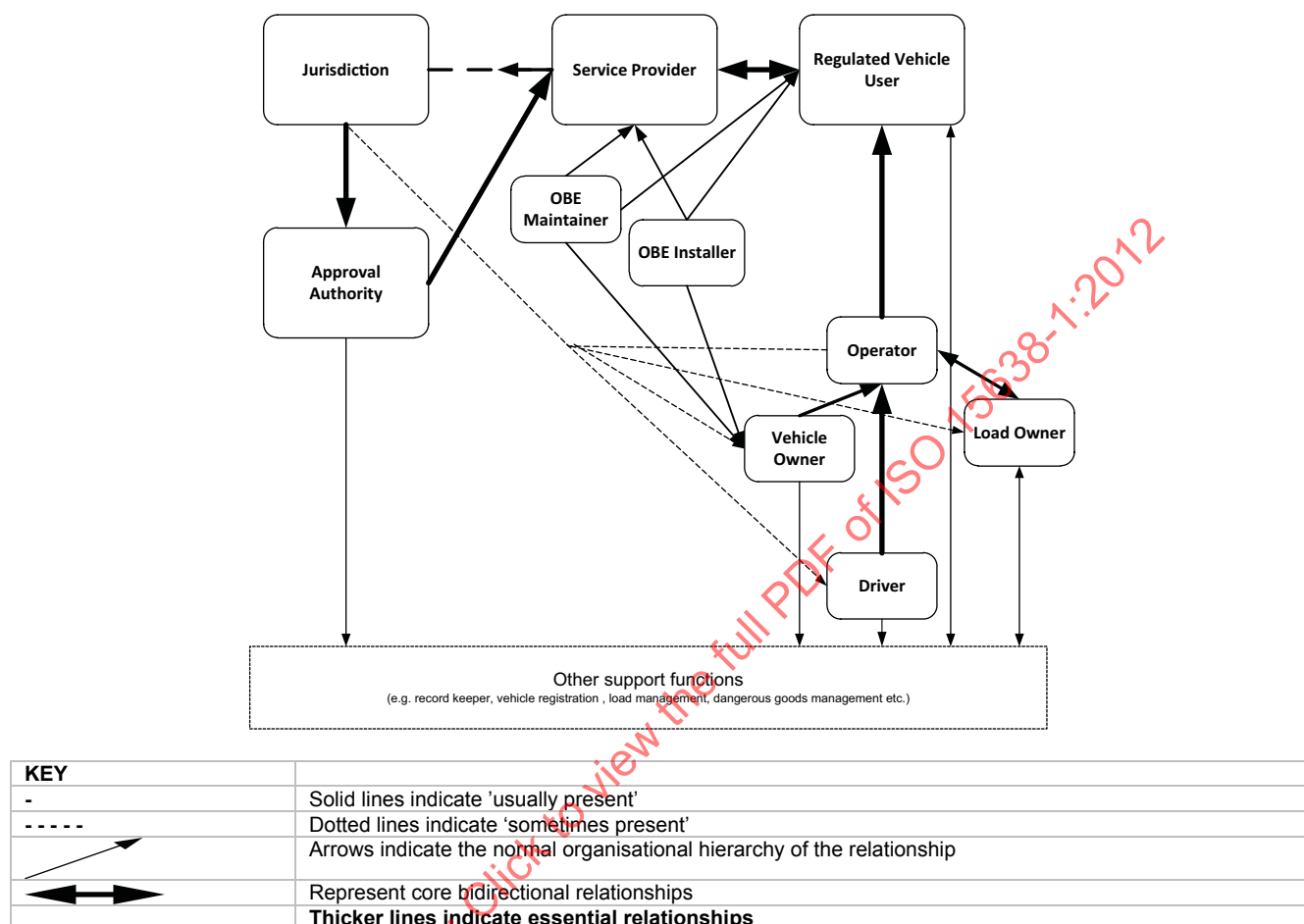


Figure 13 — TARV Taxonomy of actors and relationship of actions

The *approval authority (regulatory)* (4.6) is *appointed* (4.3) by the *jurisdiction* (4.24) and is its prime interface to the *service provider* (4.39), however the *service provider* provides reports on regulated commercial freight vehicles back to the *jurisdiction* as required by the regulation.

The *jurisdiction* (4.24) has a one way relationship with the driver, vehicle owner, or freight owner in the event of violation of the regulations and possibly registration.

The *IVS* (4.23) installer will install the equipment to the instruction of the *service provider* (4.39), but requires the consent of the vehicle owner and/or operator to effect such installation as it needs access to the vehicle to install the equipment. The *IVS* maintainer will have a relationship with both the owner of the vehicle and the *service provider* and also needs access to the vehicle.

The driver is an employee or contractor of the operator and between them they form the vehicle *user* (4.45) but in a hierarchical relationship. The operator has a relationship with the vehicle owner who is giving him permission to use of the vehicle. The operator has a bidirectional relationship with the freight owner for whom he is conveying the freight.

There may be no direct relationship between the vehicle owner and the driver, nor the vehicle owner and the freight owner, and no direct relationship between the freight owner and the driver. There may be no direct relationship between the freight owner and the vehicle owner and the *service provider* (4.39), nor between the *IVS* (4.23) maintainer and installer and the freight owner.

11 The communications architecture

The communications *architecture* (4.7) is based on the *CALM* suite of International Standards, and the key communications reference standards and aspects relevant to *TARV* shall be as defined in ISO 15638-2.

12 Interoperability and the TARV-ROAM 'facilities' layer

12.1 Interoperability with other cooperative ITS systems

The requirement for interoperability cannot be limited solely to *regulated application services* (4.36) for *regulated commercial freight vehicles* (4.37).

In a connected world, the most effective business case for an in-vehicle platform in regulated commercial freight vehicles is one where a single platform can manage multiple functions, and can communicate with other classes of vehicles. This shall be achieved using peer to peer communications between ITS-Stations (ITS-s), as defined in ISO 15638-2.

In the case of *regulated commercial freight vehicles* (4.37), the ISO 15638 suite of standards deliverables focuses on the provision of *regulated application services* (4.36) and commercial *application services* (4.2) to *regulated commercial freight vehicles* (4.37). However these communications will exist in a wider world of vehicle-to-infrastructure communications and vehicle-to-vehicle communications, and to a wider range of vehicles than only 'regulated commercial freight vehicles', such as that envisaged for C-ITS. The concept of developing radically different platform architectures for different classes of vehicle is unattractive to equipment manufacturers.

While the ISO 15638 suite of standards deliverables limits itself to *application service* (4.2) provision to *regulated commercial freight vehicles* (4.37), the provision of other V2I/V2V services using the same on-board platform needs to be considered, and while not specified within the ISO 15638 suite of standards deliverables, enabled.

Most government led *ITS* initiatives around the world recognise the needs for interoperability.

EXAMPLE In Europe the 2010 EC ITS Implementation Directive 2010/40/EU was adopted on 7 July 2010 and aims to provide a legal framework to support the actions required by the 'ITS Action Plan' to accelerate the deployment of innovative transport technologies across Europe. This Directive is an important instrument for the coordinated implementation of *ITS* in Europe. It aims to establish interoperable and seamless *ITS* services while leaving Member States the freedom to decide which systems to support. The ITS implementation Directive summarises the requirements for interoperability as:

"(11) ITS should build on interoperable systems which are based on open and public standards and available on a non-discriminatory basis to all application and service suppliers and users."

and

Once the necessary specifications for the priority actions have been adopted, the Commission shall adopt specifications ensuring compatibility, interoperability and continuity for the deployment and operational use of ITS for other actions in the priority areas.

and

'6. The specifications shall, where appropriate, be based on standards'

and

"(e) Deliver Interoperability – ensure that systems and the underlying business processes have the capacity to exchange data and to share information and knowledge to enable effective ITS service delivery."

Under this Directive the European Commission is required to adopt within the next seven years specifications (i.e. functional, technical, organisational or services provisions) to address the compatibility, interoperability and continuity of ITS solutions across the EU. The first priorities include linking the vehicle with the transport infrastructure.

This also includes the definition of necessary measures to integrate different ITS applications on an open in-vehicle platform, based on:

- the identification of functional requirements of existing or planned ITS applications
- the definition of an open-system architecture which defines the functionalities and interfaces necessary for the interoperability/interconnection with infrastructure systems and facilities
- the integration of future new or upgraded ITS applications in a 'plug and play' manner into an open in-vehicle platform
- the use of a standardisation process for the adoption of the architecture, and the open in-vehicle specifications.

Interoperability is commonly defined as '*the ability of two or more systems or components to exchange information and to use the information that has been exchanged*' (IEEE Glossary), but this definition results in significant limitation. Most particularly it encourages, for example, the adoption of a single technical solution. In a world of rapidly developing technology, it can fossilise the solution such that it becomes quickly extinct.

ISO 15638, therefore defines interoperability as a

'property of a product or system, whose interfaces are completely understood, to work with other products or systems, present or future, without any restricted access or implementation.' (Wikipedia)

This generalized definition can then be used on any system, not only information technology systems.

It is necessary to consider that 'interoperability' has three aspects:

- Technical;
- Operational;
- Contractual/commercial.

In the case of an on-board platform for the provision of *ITS* services (including *regulated application services* (4.36) for regulated commercial freight vehicles):

Technical interoperability implies the ability of two or more systems to be able to successfully communicate with each other, operating systems that can successfully interact together, and input and output that is understood by all parties and can therefore be used. (Rather than a single communication interface, single operating system and single programme).

Operational interoperability is where the underlying business processes have the capacity to exchange data and to share information and knowledge to enable effective *ITS* service delivery.

Contractual interoperability is where services can be provided in a multi *user* (4.45), multi *service provider* (4.39) environment. Within the context of TARV, one of the major obstacles to contractual interoperability is that while it makes sense to the *jurisdiction* (4.24) and *user* (4.45), it often makes little sense to the commercial interest of the installer of the equipment. Finding the business case to make equipment interoperable (which is usually a more expensive option) usually proves difficult, when the result is that a third party has the benefit of the original service provider's investment, or even worse when it provides the opportunity to take the business away from the initial *service provider*.

Technical and operational interoperability are best accomplished using open standards/specifications. Modular standards provide the best options for consensus and durability. Technical interoperability in a fast developing environment requires standards that have migration possibilities.

EXAMPLE EC 2009 ITS Action Plan. "This open system architecture would be embodied in an open in-vehicle platform, guaranteeing interoperability/interconnection with infrastructure systems and facilities. With this modular approach, additional functionalities could be integrated later for in-vehicle safety and safe HMI, personal mobility, logistics support and access to multimodal information and possibly electronic vehicle identification."

Contractual interoperability solutions in the case of *ITS* include regulation to require interoperability if it makes sense for the *jurisdiction* (4.24), or to provide interoperable equipment as part of the vehicle original equipment. This latter option is only viable if the *user* (4.45) sees the benefit and is prepared to meet the additional cost. This use case is far more likely where one or more key applications are desired by the *user* (4.45).

The *specifications* (4.41), where possible, should be performance-based, only being prescriptive in areas to achieve interoperability. This implies that, in general, only the content and quality of the required data and its exchanges should be specified. This leaves the *application service* (4.2) provider maximum freedom to provide a cost-effective solution and determine how the required data will be delivered, and will/can migrate over time. However, consistent and effective communications lie at the heart of all interoperability so some limitations will be required here.

Using open system *architecture* (4.7), implemented in an open in-vehicle platform, enabling interoperability/interconnection with infrastructure based *ITS* systems and facilities provide an optimal route. With this modular approach, additional functionalities can be progressively integrated for *regulated application services* (4.36), in-vehicle safety, vehicle-infrastructure safety systems, vehicle-vehicle safety systems, logistics support, route guidance, infotainment, and allowing safer *HMI*.

It is not for this part of ISO 15638 to impose the way that a *jurisdiction* (4.24) handles interoperability. However one solution for *jurisdictions* to consider is the use of an 'interoperability certificate'.

Using this approach every manufacturer would have to prove the interoperability of its products with those already type-approved (a card manufacturer will have to prove that its tachograph card is readable by any other type-approved digital tachograph etc.) The approach has some appeal to *jurisdictions* (4.24) as it moves the responsibility clearly to the equipment manufacturer/application systems provider. However, as devices get more complex and multi-functional, it may become impossible to know what devices and *application services* (4.2) have already been approved. On an international perspective, with different *jurisdictions* implementing different *application services* (4.2) and different generations of similar services, the burden on the manufacturer/application service provider may become intolerable and unreasonable. However it may provide short term solutions for first generation equipment.

The requirements for interoperability are recognised in the ISO 15638 suite of standards deliverables.

The first and major step of the ISO 15638 suite of standards deliverables to achieve communications interoperability shall be to adopt (via ISO 15628-2, TARV - Common platform parameters using *CALM*) a common and flexible approach to communications. In the technical *architecture* (4.7) of *CALM*, transparency between the communications layer and the application layer is provided by a middleware layer. While supporting several wireless media options, *CALM* offers common management of the network such that most applications can be provided without knowledge of the wireless communications media being used, and applications can be provided over different media according to the particular situation or as technology evolves.

This has the direct advantage that most vehicle-to-infrastructure and vehicle-to-vehicle systems currently being developed for vehicle safety systems are using the *CALM* suite of International Standards, therefore ensuring interoperability of the basic wireless communications between these systems and *application services* (4.2) for *regulated commercial freight vehicles* (4.37).

The second step to achieve interoperability is accomplished by a combination of the three common commands which shall be as specified in ISO 15638-3 (TARV – Operating requirements, 'Approval authority' approval procedures, and enforcement provisions for the providers of regulated services) 'GET BVD', 'GET CAD', and 'GET stored data', the commands to obtain *basic vehicle data* (47), and optional additional components for *core application data* (4.17), and to obtain other data stored for the application.

The third step to achieve interoperability shall be accomplished by the provision for a common *specification* (4.40) for *basic vehicle data* (4.10), and common options for *core application data* (4.17) specified in ISO 15638-5.

The fourth step to achieve interoperability shall be the core common specifications to provide a number of *regulated application services* (4.36) as defined in ISO 15638-6.

The fifth step to achieve interoperability shall be the definition of a consistent standard *framework* (4.20) to provide non-regulated *application services* (4.2) as defined in ISO 15638-7.

The sixth step to achieve interoperability shall be to rely wherever possible to *specification* (4.40) by reference to other standards for identification and for commercial vehicles.

12.2 TARV-ROAM ‘facilities layer’ architecture

12.2.1 General summary of TARV-ROAM provisions

This Clause describes and defines how the facilities layer for TARV, within the ISO 15638 suite of standards deliverables, provides an open access, yet secure runtime environment for TARV and other applications, including cooperative vehicle applications, on top of the CALM communications environment.

This part of ISO 15638 provides *specification* (4.40) (and interface to applications) of the:

- ROAM (Regime for Open Application Management) *framework* (4.20) and *architecture* (4.7);
- ROAM Facilities and management provisions
 - Access to CALM network functionality;
 - Distributed directory service facilities;
 - Access (at facilities layer) to roadside communications;
 - Shared access to an in-vehicle telematics platform;
 - Principles for security services (Detailed security provisions shall be as defined in ISO 15638-4);

and in some circumstances:

- Access to vehicle sensors and actuators.

12.2.2 Acknowledgements

This part of ISO 15638 and the design principles behind it are designed to take advantage of, and be compatible and interoperable with the CVIS FOAM (Cooperative vehicle infrastructure Systems (project) - Framework for Open Application Management) *architecture* (4.7), and ITS cooperative systems that support similar facilities and open application execution principles. TARV – ROAM accepts the generic principles of the CVIS FOAM project deliverable and adapts them to the TARV environment. It acknowledges that CVIS FOAM is the source of inspiration for much of the specifications within TARV – ROAM specifications. The authors of this part of ISO 15638 recognize and thank the European Commission for supporting the CVIS project, and particularly the subproject FOAM and for making the results available for public use.

At the time of specifying TARV – ROAM, no formal standards had been written in respect of CVIS FOAM and so no specifications can be made by reference to an extant standard. For reasons of interoperability and maintenance, later versions of this part of ISO 15638 may be revised to define sections of this part of ISO 15638 by reference to approved International Standards as these evolve.

However, in order to maximise security using equipment and systems that may be less sophisticated than those that eventually ensue for full C-ITS, additional restrictions on the communications allowed in ISO 15638

series of International Standards may provide less capable communication within the context of TARV, than will be possible with fully developed C-ITS. However, due to the use of CALM standards, the same ITS-station/IVS should be capable to support both TARV and full C-ITS applications.

Project specific terms used within the project CVIS are assumed to be proprietary, and to carry some project specific import, and so have not been used in this part of ISO 15638. Terms used in this part of ISO 15638 are offered openly and without restraint of intellectual property. However later versions of this part of ISO 15638 may adapt to use terms commonly accepted in approved International Standards that are adopted for similar subject areas.

Information concerning OSGi® (Open Services Gateway initiative) is acknowledged as being sourced or adapted from OSGi® publicly available material.

12.3 ROAM framework and architecture

12.3.1 ROAM overview

The *ROAM* (Regime for Open Application Management) *architecture* (4.7) provides the *framework* (4.20) and operational environment for developing and deploying platforms for *TARV* applications within a general *framework* (4.20) of cooperative vehicle telematics systems, and is designed not only to support *TARV* application systems (See ISO 15638-6), but also to support other commercial and safety cooperative systems for commercial vehicles beyond the scope of the *TARV regulated applications* (4.35) (See ISO 15638-7), and general cooperative vehicle systems for all classes of vehicles. It is therefore designed to be compatible and interoperable with other cooperative vehicle standards, and has used the successful results of research programmes and applications in these areas as its source of inspiration.

This *ROAM* standards deliverable provides an open end-to-end application *framework* (4.20), connecting *in-vehicle systems* (4.23), roadside infrastructure and back-end infrastructure for *TARV*. This *framework* may be regarded as a foundation and facilitator for the provision of *TARV application services* (4.2) in an international environment, without losing the discretion and authority of local *jurisdictions* (4.24) to implement different ranges of services within their domain.

ROAM provides an open execution environment in which *TARV* applications can be developed, delivered, implemented and maintained during the life cycle of both service applications and equipment. Drivers and vehicle operators will be able to rely on their integrated *in-vehicle system* (4.23) to allow *TARVs* to operate within the requirements of *jurisdictions* (4.24) within which they drive their vehicles, and gain advantages from direct co-operative management of transport safety and efficiency wherever they drive.

TARV applications for regulated commercial freight vehicles will be implemented according to the regulations of implementing *jurisdictions* (4.24), and at their discretion. A number of generic and interoperable *regulated application services* (4.36) methodologies are provided as a toolbox to *jurisdictions* in ISO 15638-6 (*TARV Regulated applications*) and subsequent parts of ISO 15638 as approved. The methodology to support commercial services for *TARVs*, and to cooperate/interoperate with the provision of general safety services for all classes of vehicles is provided in ISO 15638-7.

Within the *TARV* environment, *regulated applications* (4.35) are developed by *jurisdictions* (4.24) and deployed by *application service* (4.2) providers to 'Host Management Centres' (*HMC*). Within the *architecture* (4.7) determined in this part of ISO 15638, the *HMC* provides a service gateway that supervises the secure provision of software and services for *TARVs*. *HMCs* manage the provisioning of applications to any authorised and subscribed *user* (4.45) via its client system. After it is properly provisioned and installed on the client system it can enact the application. Mechanisms for flexible software deployment and management are provided by JAVA®/OSGi® (open services gateway initiative), and the overall *framework* (4.20) and *architecture* is therefore already well proven in use in other domains, such as mobile telephony.

The *Architecture* (4.7) and context for *TARV* above the communication layer supports:

- an end-to-end *framework* (4.20) and run-time environment in which road management and transport safety applications can be developed, deployed and provisioned;

- an open end-to-end *framework* supporting *CALM* vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I) and infrastructure-to-infrastructure (I2I) information and transaction exchange and support of applications such as tachograph auditing, electronic driver diaries, road use charging etc.;
- a secure run-time environment including authentication of data, authorization of users, intrusion protection;
- to supply client side run time environments for both rich JAVA® clients as well as embedded native clients;
- To develop generic open source interface for the cooperative exchange of data between vehicles and infrastructure (V2I);
- To support the operation of both *regulated applications* (4.35) for *TARVs* and commercial and safety applications for *TARVs*;
- To enable *TARVs* to cooperate with other classes of vehicles in the provision of *cooperative ITS* (4.14) services and applications.

ROAM supports this *framework* (4.20) by providing:

- exchange and updating of service application components at any domain in the *TARV architecture* (4.7);
- product / vendor independence by unified or adaptable middleware components;
- common design of structural elements (e.g. data exchange formats, protocol *specifications* (4.41), *API* (application program interface)'s and run-time environment);
- common design of secure communication in distributed systems including authentication of *user* (4.45) data and authorization of users (jointly within ISO 15638-4 specifications);
- re-usability of components by generalised access of resources.

While the requirements for regulated commercial freight vehicles are very specific to the domain of a *jurisdiction* (4.24), and will vary from one *jurisdiction* to another, equipped *TARVs*, in a world of *cooperative ITS* (4.14) systems, do not operate in isolation, and the on-board platform designed to support *TARV regulated application services* (4.36) will also provide other commercial services to *TARVs* and interoperate and support general cooperative safety services for all classes of vehicles.

12.3.2 ROAM OSGi® JAVA® environment

Figure 14 illustrates *TARV* service provision with *ROAM* Identified.

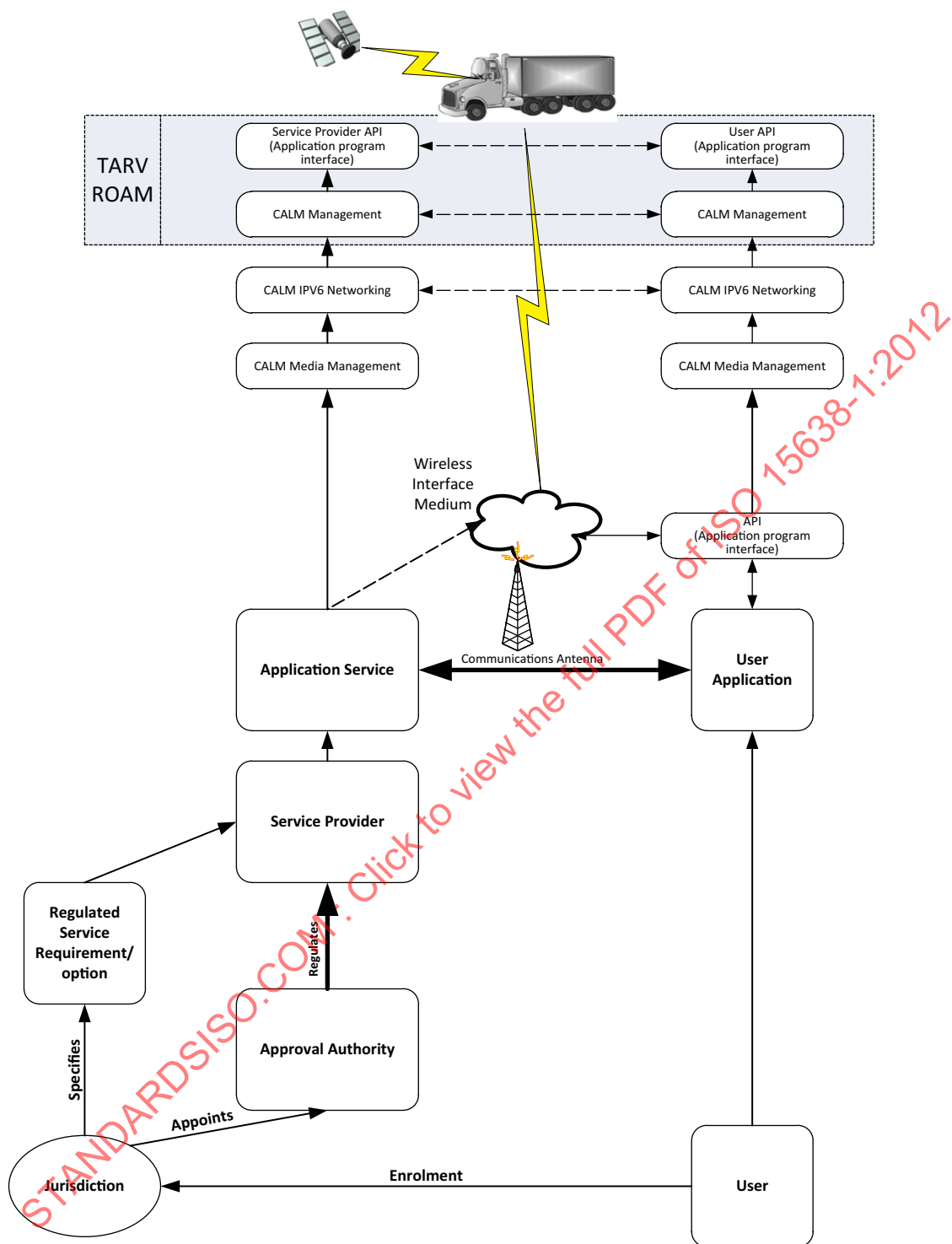


**Figure 14 — TARV service provision with ROAM identified
(Figure 3 modified)**

Please also consult Figure 12 above which illustrates the ITS-station reference *architecture* (4.7).

ROAM provides that, an OSGi® JAVA® type environment is created (much as on smart phones) on top of the communication layers provided by CALM (The use of a similar approach to smart phones both builds on the experience of that sector, and enables commonly available components and software to be used, thus minimising the risks and costs associated with this methodology).

This involves several 'layers'.

12.3.3 ROAM application layers

On the top layer are any applications running on the on-board platform. For *TARV* these will of course be the *TARV application services* (4.2). However, it is logical that for efficiency, the on-board platform will be available to also carry out other tasks, both non regulated services for *TARVs*, and other services such as safety services and information services. These services may be in-vehicle, but may and increasingly will, be cooperative with other vehicles in a *V2V* and *V2I* environment. The ability for the on-board platform to multi-task these 'app's is enabled by the *architecture* (4.7) and is made more efficient by the internal sharing of relevant data. These applications are task specific, and may well commonly use some of the data concept values created for *TARV* and available in the data pantry for other applications, and vice versa, data created for other purposes may be used to support *TARV application services*. This layer is unsurprisingly called the 'Applications' layer.

12.3.4 ROAM 'App' library and data pantry

A layer below these applications is the provision of data for the data pantry. This data provisioning is not generated by a single application, but by a number of small task specific 'Facilities apps'- which are generally small *JAVA®* 'applets', organised as software bundles, that generally busy themselves keeping the data pantry provisioned with up to date data. This data provisioning is envisaged to be carried out by the *Facilities apps*, each of which will service the updating of individual data elements in the '*basic vehicle data* (4.10)' concept, and for the '*core application data* (4.17)' concept where a *jurisdiction* (4.24) has specified or provided an 'app' to do this.

A key feature of this 'layering' is the principal that a particular layer can only communicate with the adjacent layer immediately above or below it or to its side. The communication infrastructure is therefore hidden from the application by the middleware, and the 'app's are separated from the resultant data.

It is crucial also that the data pantry contains just end data. The data pantry is accessible to an *app*, so long as it has authorisation, but the software *app* that generated the data is not available to the *app* online. A *jurisdiction* (4.24) wishing access to an *app* in a commercial vehicle of another country can obtain this only via the home *jurisdiction* of the vehicle. The internet enables this to be a virtually instantaneous provision.

12.3.5 Providing Apps 'on the fly'

In this way a *jurisdiction* (4.24) can provide an *app* to a vehicle entering its territory, that provides an up to date means to determine and 'pull' the data that it has declared that it requires as 'core data'. Once it has provided that *app* it may subsequently replace it with a newer version, but it may not tinker with it on-board the vehicle.

The 'app' will then seek to access the data pantry to obtain the data that it requires and supply that to the application service provider, who will provide it to the *jurisdiction*, having verified their authenticity.

In order to manage these processes a 'Host Management Centre' (*HMC*) is required both at the *jurisdiction* (4.24) ITS-Station and in the vehicle *IVS* (4.23).

The *OSGi® framework* (4.20) provides a comprehensive set of functions to provide and deploy software bundles. This allows the addition, change or removal of applications software or facility software during the runtime of the system. It is proposed that ISO 15638-5 will implement *JAVA®/OSGi®* in such a way, that these runtime system changes can be enacted remotely through the "Host Management Centre" (*HMC*) and the corresponding *HMC* on the associated hosts, without compromising the principles described above. *OSGi®* is open specification (4.40) and so will not involve any IPR payments or licenses.

12.3.6 ROAM execution infrastructure

ROAM offers an environment which provides the means to set up the interfaces by providing standard functionalities. From a *ROAM* perspective the data is regarded as an arbitrary array of bytes. *ROAM* will provide a non-normative application manager that will provide a basic default *HMI* to the end user (4.45). It is based on *JAVA® AWT* and is completely skin-able (the function can operate dressed in the guise/style most suited to the application).

The *TARV-ROAM regulated application* (4.35) execution infrastructure is based on the *OSGi® framework* (4.20). Since the current *OSGi® specification* (4.40) only provides a binding to the *JAVA®* platform, the execution infrastructure is implemented by a set of standard *JAVA® APIs*, as well as a "JAVA® Virtual Machine" (JVM), which runs on top of the operating system.

Please note that the dependency from the execution infrastructure on the underlying operating system constitutes a formal interface between the middleware and the communication infrastructure, where the communication infrastructure is the provider of the *TARV* host hardware, including the operating system.

The lifecycle layer provides a lifecycle *API* to bundles. This *API* provides a runtime model for bundles and is used for lifecycle management in *TARV*. The lifecycle *API* defines how bundles are started and stopped as well as how bundles are installed, updated and uninstalled. Additionally, it provides a comprehensive event *API* to allow a management bundle to control the operations of the service platform.

The service layer provides a dynamic, concise and consistent programming model for *JAVA®* bundle developers, simplifying the development and deployment of service bundles by de-coupling the service's *specification* (4.40) (*JAVA®* interface) from its implementations. This model allows bundle developers to bind to services only using their interface specifications. The selection of a specific implementation, optimized for a specific need or from a specific vendor, can thus be deferred to run-time.

A consistent programming model helps bundle developers cope with scalability issues in many different dimensions. This is critical because the *framework* (4.20) is intended to run on a variety of devices whose differing hardware characteristics may affect many aspects of a service implementation. Consistent interfaces ensure that the software components can be mixed and matched and still result in stable systems.

The *framework* (4.20) allows bundles to select an available implementation at run-time through the *framework* (4.20) service registry. Bundles register new services, receive notifications about the state of services, or look up existing services to adapt to the current capabilities of the device. This aspect of the *framework* (4.20) makes an installed bundle extensible after deployment: new bundles can be installed for additional features or existing bundles can be modified and updated without requiring the system to be restarted.

12.3.7 TARV-ROAM actors and IVS component decomposition

Figure 15 provides a component decomposition of the *TARV IVS* (4.23).

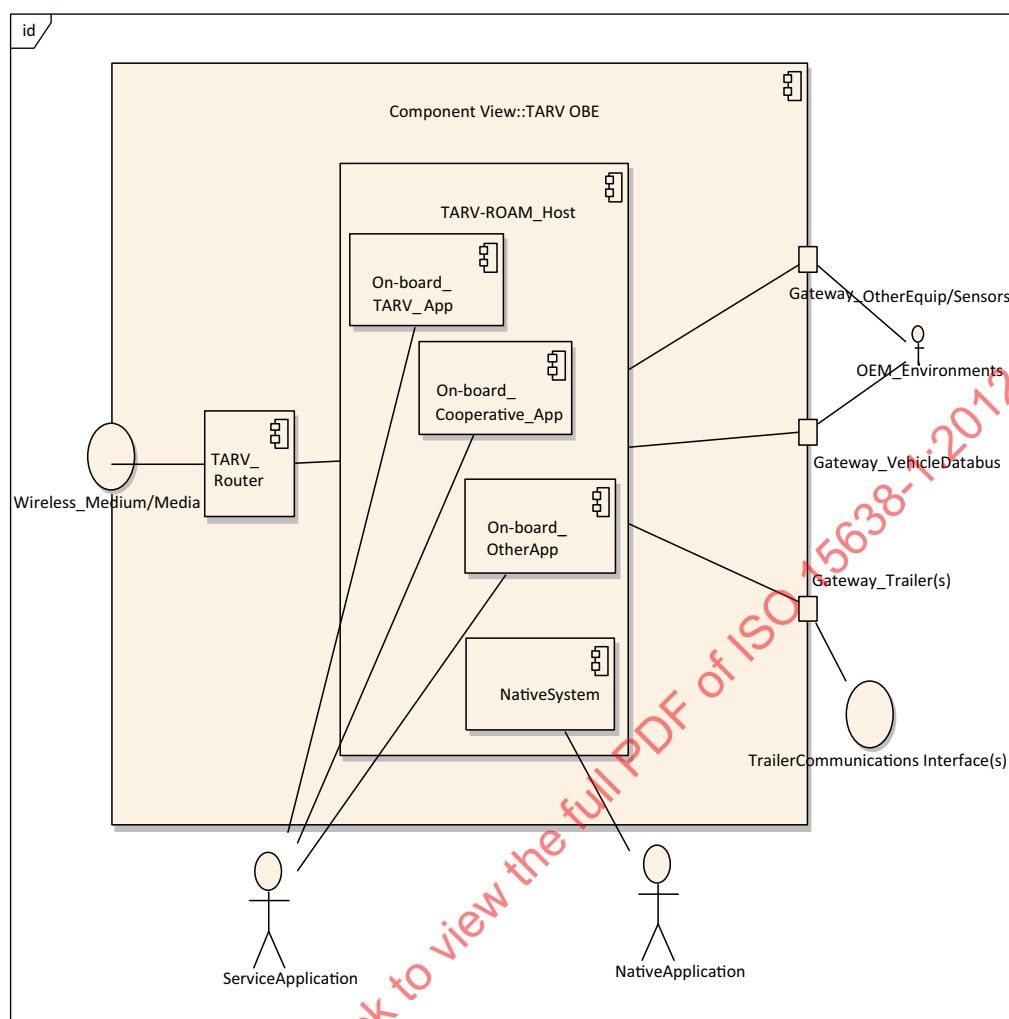


Figure 15 — TARV IVS component decomposition

The components of Figure 4, and *TARV-ROAM* actors are defined as:

12.3.7.1 OEM_Environment

Existing infrastructure which may exchange data (based on a strict authorization model) with a *TARV* on-board system. (examples : *CAN bus* (4.11), trailer identification/communications interface, sensors etc.).

12.3.7.2 TARV-ROAM application service

Application services (4.2) consist of software-components that the *TARV-ROAM_serviceplatform* is capable of dynamically loading, activating, deactivating, updating, and unloading. An *application service* (4.2) is transacted by software running on a *TARV-ROAM_serviceplatform* on a specific *TARV host*. An *application service* (4.2) is the actual instantiation of a service and delivering this service to the *user* (4.45) or to another *application service* using this software. It may consist of more than one component, each providing part of the functionality of the overall service.

12.3.7.3 TARV-ROAM certification authority (digital)

(not shown explicitly in Fig 15)

An organization which issues digital certificates for use by other parties. (specifically in the context of communications security).

12.3.7.4 TARV-ROAM approval authority (regulatory)

(not shown explicitly in Fig 15)

An organisation (usually independent) which conducts *approval* (4.4) and ongoing *audit* (4.8) for TARV 'Service providers' (4.40).

12.3.7.5 TARV-ROAM Communication subsystem/Wireless_Medium/Media

A TARV subsystem mainly comprising CALM wireless routers with the necessary air interfaces, including software for operation and management of these routers including the TARV_Router function.

12.3.7.6 TARV_gateway

Part of the TARV_unit that provides an interface between other extant technology and the TARV environment.

12.3.7.7 TARV-ROAM_host_platform

A TARV-ROAM_hostplatform is part of the TARV unit that hosts one or more TARV-ROAM_serviceplatforms. It can be equipped with very specialized hardware suited for a specific application domain. The TARV-ROAM_host_platform is the software on the TARV-ROAM_host to manage the execution life-cycle of application services (4.2).

12.3.7.8 TARV-ROAM_host

(not shown explicitly in Fig 15)

A TARV-ROAM_host is the actor that manages the TARV-ROAM_host_platform usually the TARV-ROAM HMC (Host Management Centre) operator

12.3.7.9 TARV-ROAM HMC (Host Management Centre) operator

(not shown explicitly in Fig 15)

An organization being responsible to run an OSGi® 'Host Management Centre'.

12.3.7.10 TARV-ROAM jurisdiction

(not shown explicitly in Fig 15)

A *jurisdiction* (4.24) is government, road or traffic authority which owns the 'Regulatory applications' (4.35). A *jurisdiction* often administers via use of a *regulator* (4.25) and or a jurisdiction approval authority (which is not to be confused with a digital approval authority).

12.3.7.11 TARV-ROAM native application

A piece of software running on the native system and that needs to interact with the TARV world or that exposes services to the TARV world.

12.3.7.12 TARV-ROAM native system

A specific part next to the ROAM environment, that is not under the direct control of ROAM itself. The native environment can be present in separate hardware or it is the native operating system where the ROAM environment is running on top. TARV-ROAM shall support service applications on the native system. For example, linkage to the TARV-ROAM environment and remote management shall be supported.

12.3.7.13 TARV-ROAM roadside system

(not shown explicitly in Fig 15)

A specialization of an *OEM_Environment*, often installed at roads which may contain a *TARV* unit (roadside ITS-station) and which may provide access to roadside sensor data and roadside actuators. Sometimes also used as a synonym for 'Roadside Equipment' (*RSE*) or 'Roadside Infrastructure' (*RSI*). In instantiation it may not actually be located at the roadside (for example where the wireless communications medium is satellite communications, GSM, UMTS or *LTE* etc.).

12.3.7.14 TARV_router

Part of the *TARV_unit* that is responsible for connecting *TARV-ROAM_hosts* to the *CALM_wireless_network*. A *TARV_router* can function as a mobile router, access router or border router.

12.3.7.15 TARV-ROAM service centre

(not shown explicitly in Fig 15)

A specialization of an *OEM_environment*, which forms the back-end infrastructure that an *application service* (4.2) provider constructs, deploys, and operates additionally to the application service. It frequently comprises remote server(s) and applications supporting and communicating with the *application service* on the *TARV* host.

Typical specializations of a service centre can be a traffic management centre, a 'Public Service Access Point' (PSAP) etc.

12.3.7.16 TARV-ROAM service centre operator

(not shown explicitly in Fig 15)

An organization which runs a service centre.

12.3.7.17 TARV-ROAM service provider

An organization responsible for the creation and delivery of *application services* (4.2) to service centres and users.

12.3.7.18 TARV_unit/ TARV_IVS

A *TARV_unit* is an *IVS* (4.23) consisting of one or more *TARV-ROAM_hosts*, *TARV_routers* and/or *TARV_gateways*. A *TARV_unit* is 'always on' connected to the *TARV_network*. Thus they can communicate amongst each other, or with service centres or its control centre.

12.3.7.19 TARV user

(not shown explicitly in Fig 15)

An actor using *TARV* services. Usually the vehicle operator. This stakeholder identifies who uses the system. The driver is a subclass of *user* (4.45) in some systems

12.3.7.20 TARV-ROAM vehicle

(not shown explicitly in Fig 15)

In the context of *TARV*, is a regulated commercial vehicle. In *TARV-ROAM* system terms it is a specialization of an *OEM_Environment*, which may contain a *TARV* unit and which may provide access to vehicle and sensor data.

12.4 OSGi® (open services gateway initiative)

12.4.1 OSGi® framework

TARV-ROAM utilises an OSGi® (open services gateway initiative) binding. This binding enables the interoperability of applications across different client systems and *jurisdictions* (4.24) by choosing JAVA® as a common programming language, OSGi® (open services gateway initiative) as a common *framework* (4.20), and defining a set of *TARV APIs* for the functionalities shared between the application services in the *TARV* client system (as defined below).

OSGi® specifications define a dynamic component system for JAVA®. These specifications enable a development model where applications are (dynamically) composed of many different (reusable) components. The OSGi® specifications enable components to hide their implementations from other components while communicating through 'OSGi® Services (4.33)', which are objects that are specifically shared between components. This surprisingly simple model has far reaching effects for almost any aspect of the software development process.

OSGi® offers a dynamically assembled collaborative software environment of reduced complexity in almost all aspects of development. Code is easier to write and test, reuse is increased, build systems become significantly simpler, deployment is more manageable, bugs are detected early, and the runtime provides an enormous insight into what is running.

EXAMPLE A home server that is capable of managing lights and appliances. A component could allow you to turn on and off the light over a web page. Another component could allow you to control the appliances via a mobile text message. OSGi® enables these other functions to be independently added without requiring that the developers had intricate knowledge of each other.

12.4.2 Layering

The OSGi® has a layered model that is depicted in Figure 16.

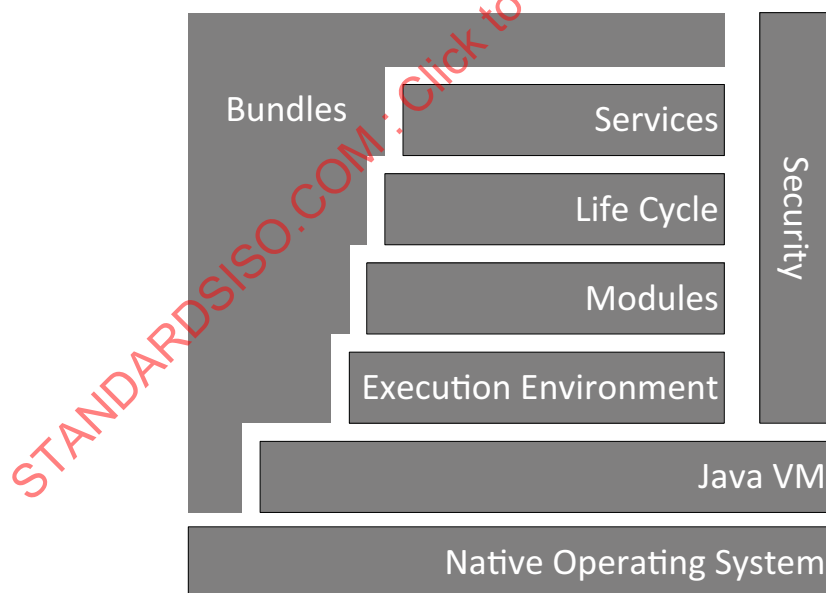


Figure 16 — OSGi® layered model
(Source: replicated from OSGi®)

OSGi® specific terms are used as follows:

— *OSGi® Bundles* (4.28) - Bundles are the OSGi® components made by the developers.

- *OSGi® Services (4.33)* - The services layer connects bundles in a dynamic way by offering a publish-find-bind model for plain old JAVA® objects.
- *OSGi® Life-cycle (4.30)* - The *API* to install, start, stop, update, and uninstall bundles.
- *OSGi® Modules (4.31)* - The layer that defines how a bundle can import and export code.
- *OSGi® Security (4.32)* - The layer that handles the security aspects.
- *OSGi® Execution environment (4.29)* - Defines what methods and classes are available in a specific platform.

These concepts are more extensively explained in the following sections.

12.4.3 Modules

OSGi® is modular. Modularity keeps units (bundles) local and not sharing. In JAVA® terms, a bundle is a *JAR* file. However, where in standard JAVA® everything in a *JAR* is completely visible to all other *JARs*, OSGi® hides everything in that *JAR* unless explicitly exported. A bundle that wants to use another *JAR* must explicitly import the parts it needs. By default, there is no sharing. However the *OSGi® Services (4.33)* model is about bundles that collaborate.

12.4.4 JAR files

The Java Archive (*JAR*) file format enables the system to bundle multiple files to be bundles into a single archive file. Typically a *JAR* file contains the class files and auxiliary resources associated with applets and applications.

The *JAR* file format has the following attributes:

- Security: Digitally signed. Authorised users can optionally provide software security privileges otherwise precluded.
- Decreased download time: applet class files and associated resources can be downloaded to a browser in a single HTTP transaction without the need for opening a new connection for each file
- File compression.
- Packaging for extensions: a means by which functionality can be added to the JAVA® core platform, and the *JAR* file format defines the packaging for extensions.
- Package Sealing: Packages stored in *JAR* files can be optionally sealed so that the package can enforce version consistency.
- Package Versioning: A *JAR* file can hold data about the files it contains, such as vendor and version information.
- Portability: The mechanism for handling *JAR* files is a standard part of the JAVA® platform core *API*.

12.4.5 OSGi® services

OSGi® utilises a '*service registry*'. A bundle can create an object and register it with the OSGi® service registry under one or more interfaces. Other bundles can go to the registry and list all objects that are registered under a specific interfaces or class.

EXAMPLE A bundle provides an implementation of the *DocumentBuilder*. When it gets started, it creates an instance of its *DocumentBuilderFactoryImpl* class and registers it with the registry under the *DocumentBuilderFactory* class. A bundle that needs a *DocumentBuilderFactory* can go to the registry and ask for all

available services that extend the `DocumentBuilderFactory` class. Or a bundle can wait for a specific service to appear and then get a call back.

An OSGi® bundle can therefore *register* a service, it can *get* a service, and it can *listen* for a service to appear or disappear. Any number of bundles can register the same service type, and any number of bundles can get the same service. This is depicted in Figure 17.

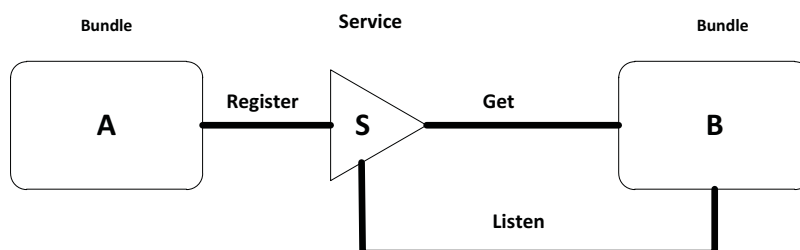


Figure 17 — OSGi® service registration process
(Source: OSGi®)

Each service registration has a set of standard and custom properties that distinguish it from others. An expressive filter language is available to select only particular services of interest. Properties can be used to find the proper service or can play other roles at the application level.

OSGi® *Services* (4.33) are dynamic. This means that a bundle can decide to withdraw its service from the registry while other bundles are still using this service. Bundles using such a service must then ensure that they no longer use the service object and drop any references. Services bundles can thus be added and installed and uninstalled on the fly while the other bundles can adapt.

These service dynamics solve the problem of initialization. OSGi® applications do not require a specific start ordering in their bundles.

The service registry also enables many specialized *APIs* to be modelled with the service registry. Not only does this simplify the overall application, it also means that standard tools can be used to debug. Though the service registry accepts any object as a service, optimally, in order to achieve reuse, registering these objects under (standard) interfaces to decouple the implementer from the client code.

OSGi® Alliance publishes 'Compendium' specifications. These specifications define a large number of standard services, from a 'Log Service' to a 'Measurement and State specification' thus obviating the need to define such *apps*.

12.4.6 OSGi® deployment

Bundles are deployed on an OSGi® *framework* (4.20), the bundle runtime environment. This is not a container like JAVA® application servers but is a *collaborative environment*. Bundles run in the same virtual machine and can share code. The *framework* (4.20) uses the explicit imports and exports to connect the bundles so they do not have to concern themselves with class loading. The management of the *framework* (4.20) is standardized. A simple *API* allows bundles to install, start, stop, and update other bundles, as well as enumerating the bundles and their service usage. This *API* is well proven to control OSGi® *frameworks*.

12.4.7 OSGi® implementations

The OSGi® *specification* (4.40) process requires a reference implementation for each *specification*. At the time of developing this part of ISO 15638 there are at least four open source implementations of the *framework* (4.20) and very many implementations of the OSGi® *Services* (4.33) provided by the open software sector.

The OSGi® (open services gateway initiative) service platform specifications [Core, *Cmpn* (4.13)] define an open standard for a *framework* (4.20) of services that include software installation, application lifecycle

management, dynamic code sharing between applications, service lookup, security, resource management, and functions necessary for remote administration of the gateway.

In the *framework (4.20)* services are swapped in and out, are dynamically updated, and communicate in a structured and dependable way with each other. The *framework (4.20)* provides a rich and structured development platform for component-based software architectures, and takes advantage of JAVA®'s ability to download code from the network. (refer to www.osgi.org)

12.4.8 OSGi® high level composite architecture

Figure 3 above showed the functionality of the OSGi® *framework (4.20)* is divided in the following layers [core]:

- Security layer;
- Module layer;
- Life cycle layer;
- Service layer
- Actual services.

12.4.8.1 Security

For further detail of the 'security layer' see ISO 15638-2.

12.4.8.2 Module layer

The 'module layer' defines a modularization model for JAVA®. It addresses some of the shortcomings of JAVA®'s deployment model. The modularization layer has strict rules for sharing JAVA® packages between bundles or hiding packages from other bundles. The module layer can be used without the life cycle and service layer.

NOTE The life cycle layer provides an *API* to manage the bundles in the module layer, while the service layer provides a communication model for the bundles.

12.4.8.3 Life cycle layer

The life cycle layer provides a life cycle *API* to bundles. This *API* provides a runtime model for bundles. It defines how bundles are started and stopped as well as how bundles are installed, updated and uninstalled. Additionally, it provides a comprehensive event *API* to allow a management bundle to control the operations of the service platform. The life cycle layer requires the module layer but the security layer is optional.

12.4.8.4 Service layer

The service layer provides a dynamic, concise and consistent programming model for JAVA® bundle developers, simplifying the development and deployment of service bundles by de-coupling the service's *specification (4.40)* (JAVA® interface) from its implementations. This model allows bundle developers to bind to services only using their interface specifications. The selection of a specific implementation, optimized for a specific need or from a specific vendor, can thus be deferred to run-time.

NOTE This assists bundle to developers to cope with scalability issues in many different dimensions – critical because the *Framework (4.20)* is intended to run on a variety of devices whose differing hardware characteristics may affect many aspects of a service implementation. Consistent interfaces insure that the software components can be mixed and matched and still result in stable systems.

12.4.8.5 Framework service registry

The *framework* (4.20) allows bundles to select an available implementation at run-time through the *Framework* service registry. Bundles register new services, receive notifications about the state of services, or look up existing services to adapt to the current capabilities of the device. This aspect of the *framework* makes an installed bundle extensible after deployment: new bundles can be installed for added features or existing bundles can be modified and updated without requiring the system to be restarted.

12.4.8.6 OSGi® interactions

The interactions of the layers are depicted in Figure 18.

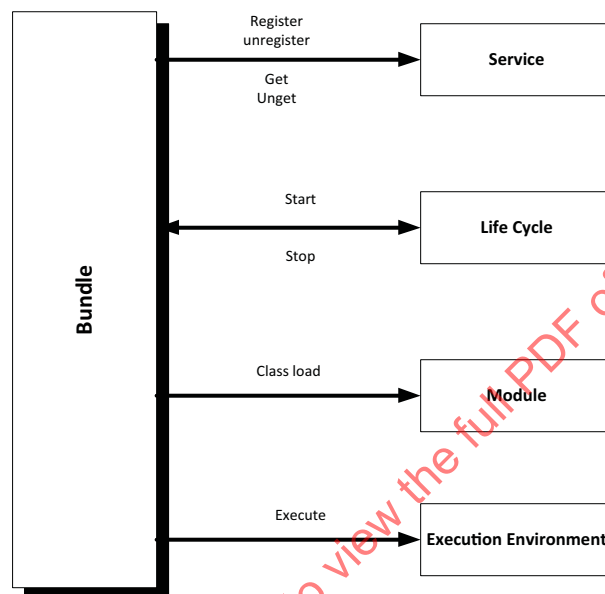


Figure 18 — Interactions of OSGi® framework layers

12.4.8.7 OSGi® services

An OSGi® service is a self-contained component, accessible via a defined service interface. Services are essential elements of the OSGi® *framework* (4.20). A service consists of at least two parts:

- one or several interfaces;
- a class that implements the interface(s).

The interface represents the functionality of the service (i.e. what it can do). Other services that use a particular service exploit the interface, not on the particular service implementation.

A single interface can have multiple implementations provided by different vendors. For example, the logging service can store information either locally or remotely.

After a service is published, other services can use it to accomplish their tasks. In order to use a particular service, other services look up the particular service in the 'Framework Service Registry' with a search filter, obtain a 'service reference' and the 'service object' corresponding to that reference, use the service, and then release it. This is called 'dynamic dependency among services'.

12.4.8.8 OSGi® bundles

To be available to the *framework* (4.20), a service implementation must be packaged.

Service implementations are packaged into 'bundles'. Bundles are essential components of the OSGi® framework (4.20).

Technically, a bundle is a *JAR* file that:

- contains the resources that implement zero or more services. These resources may be class files, as well as other data such as HTML files, help files, icons, native code (e.g. dynamic link library- dll), etc.
- contains a manifest file describing the contents of the *JAR* file and providing information about the bundle. This file uses headers to specify parameters that the framework (4.20) needs in order to correctly install and activate a bundle.
- contains a special class in the bundle to act as bundle activator. This class acts as a starting execution point of the bundle. It declares two methods, start and stop, which are invoked by the framework (4.20) during the state transitions of the bundle.
- can contain optional documentation in the OSGi®-OPT directory of the *JAR* file or one of its subdirectories. This information is not vital to the operation of the bundle.

During its lifecycle a bundle is in one of 6 states (see Figure 19):

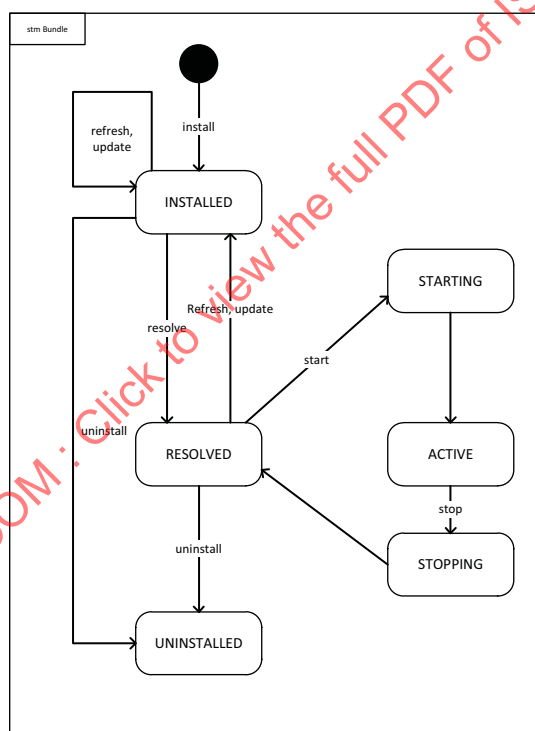


Figure 19 — OSGi® Bundle lifecycle state diagram
(source: OSGi®)

12.4.8.8.1 Installed

The bundle has been successfully installed.

12.4.8.8.2 Resolved

All JAVA® classes and/or native code that the bundle needs are available. This state indicates that the bundle is either ready to be started or has stopped.

12.4.8.8.3 Starting

The bundle is being started, and the `bundleactivator.start` method has been called and has not yet returned.

12.4.8.8.4 Stopping

The bundle is being stopped, and the `bundleactivator.stop` method has been called and has not yet returned.

12.4.8.8.5 Active

The bundle has successfully started and is running.

12.4.8.8.6 Uninstalled

The bundle has been uninstalled. It cannot move into another state.

12.4.9 Importing and exporting packages

Bundles in the *OSGi® framework (4.20)* share JAVA® packages. Each bundle is free to export any of the packages it contains.

A bundle declares the resources it offers to provide to other bundles using `Export-Package` manifest headers, and declares the resources it needs using `Import-Package` manifest headers.

The `Import-Package` manifest header allows a bundle to request access to packages that have been exported by other bundles in the *OSGi®* environment.

The fully qualified package name must be declared in the bundle's `Import-Package` manifest header for all packages bundle needs, except for package names beginning with: JAVA®.

When a bundle has imported a package from another bundle, this is called a 'static dependency' between the two bundles. A 'static dependency' has to be resolved by the *framework (4.20)* (i.e. a bundle exporting the needed package has to be available) before the depending bundle can be started.

12.4.10 OSGi® system services

OSGi® system services provide horizontal functions that are necessary in virtually every system. The 'Log Service', 'Configuration Admin Service', 'Device Access Service', 'User Admin Service', 'IO Connector Service' and 'Preferences Service' are examples of *OSGi®* system services.

12.4.10.1 Log service

The logging of information, warnings, debug information or errors is handled through the log service. It receives log entries and then dispatches these entries to other bundles that subscribed to this information.

12.4.10.2 Configuration admin service

This service provides a flexible and dynamic model to set and get configuration information.

12.4.10.3 Device access service

'Device Access' is the OSGi® mechanism to match a driver to a new device and automatically download a bundle implementing this driver. This is used for plug and play scenarios.

12.4.10.4 User admin service

This service uses a database with *user* (4.45) information (private and public) for authentication and authorization purposes.

12.4.10.5 IO connector service

The 'IO Connector Service' implements the CDC/ CLDC javax. microedition.io package as a service. This service allows bundles to provide new and alternative protocol schemes.

12.4.10.6 Preferences service

This service provides access to hierarchical database of properties, similar to the 'Windows Registry' or the 'JAVA® Preferences' class.

12.4.10.7 Component runtime

The dynamic nature of services -- they can come and go at any time -- makes writing software harder. The 'Component Runtime Specification' can simplify handling these dynamic aspects by providing an XML based declaration of the dependencies.

12.4.10.7.1 Deployment admin

The primary deployment format for OSGi® is the bundle, which is a JAR/ZIP file. The 'Deployment Admin' provides a secondary format: the deployment package. 'Deployment Packages' can combine bundles with arbitrary resources into a single deliverable that can be installed and uninstalled. A comprehensive model of resource processors allows user code to extend the resource types.

12.4.10.8 Event admin

Many OSGi® events have specific typed interfaces, making it hard to receive and filter events generically. The 'Event Admin' provides such a generic, topic-based event mechanism. The *specification* (4.40) includes mapping for all existing *framework* (4.20) and service events.

12.4.10.9 Application admin

The OSGi® bundle model is different from the typical desktop or mobile phone application model that relies on starting and stopping applications. The 'Application Admin' prescribes such a traditional application model and its required management infrastructure.

12.4.11 Further information on OSGi®

Further information on OSGi® can be obtained from www.osgi.org

12.5 TARV-ROAM layered architecture and the role of OSGi®

12.5.1 TARV layered architecture

Figure 20 provides a *TARV-ROAM layered architecture* (4.7) and the role of OSGi®.

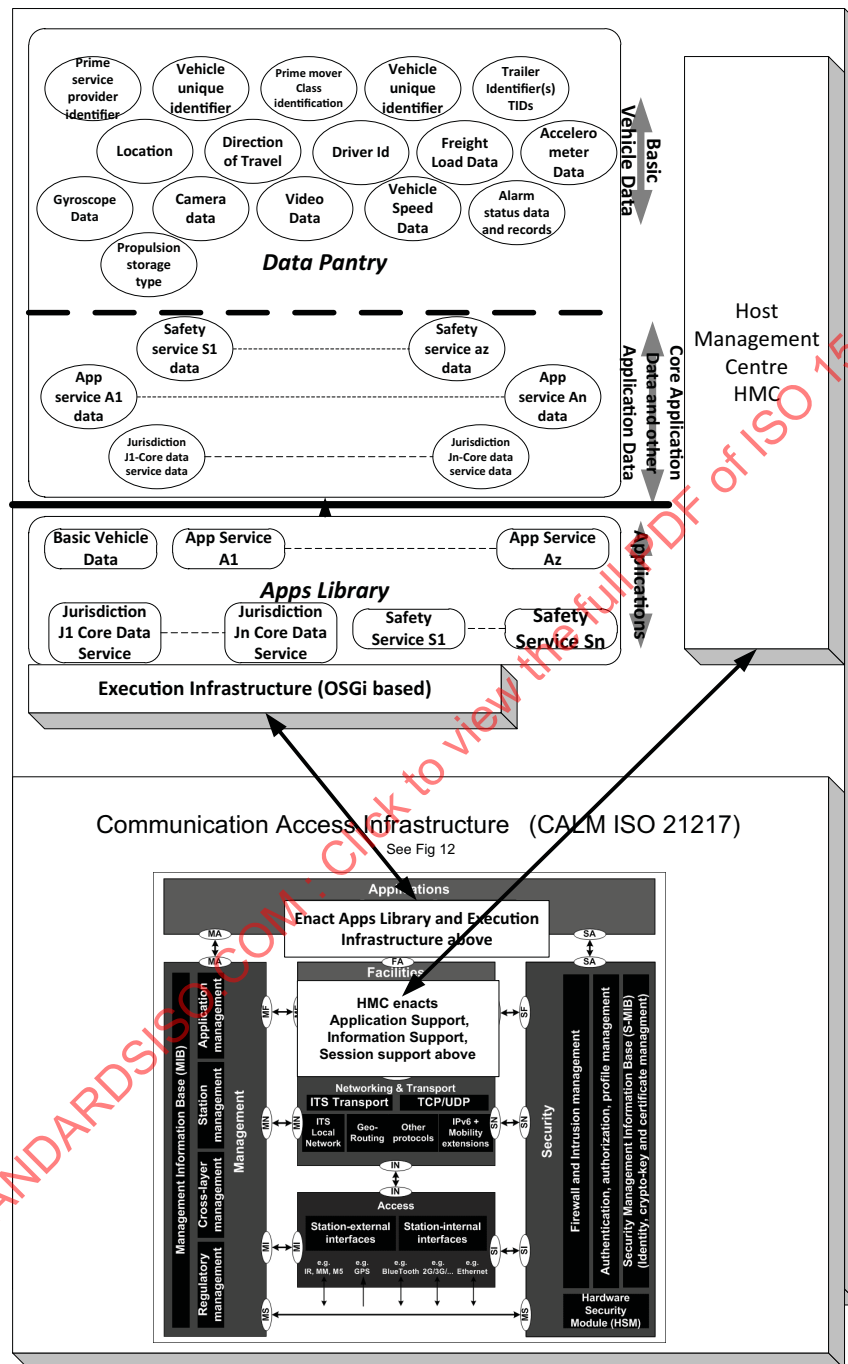


Figure 20 — TARV-ROAM Layered Architecture and the role of OSGi®

12.5.2 High level elaboration of TARV-ROAM communication architecture

Figure 21 provides a high level *UML* (4.42) elaboration of the *TARV-ROAM* communication architecture (4.7).

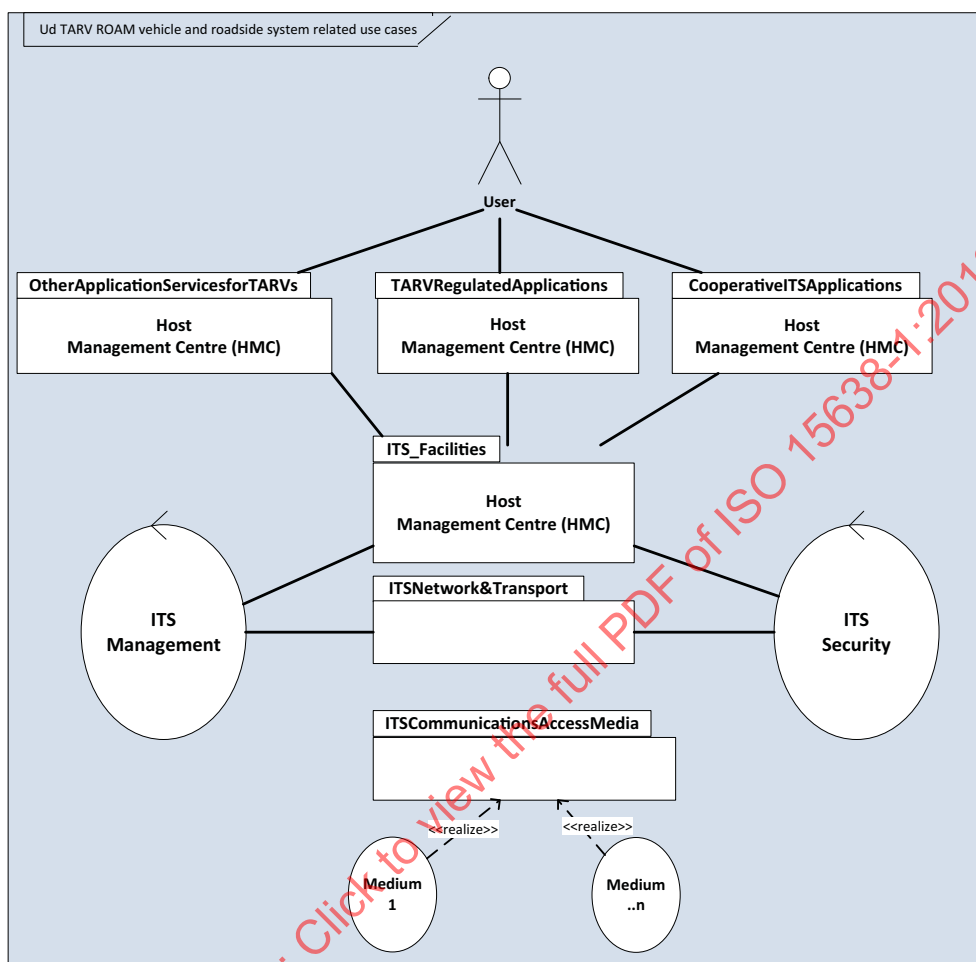


Figure 21 — UML representation of TARV-ROAM communication diagram

Within *TARV* the media access shall be handled by *CALM* (See ISO 15638-2). The facilities layer shall be accessed via *ROAM* and the applications themselves are run in the *OSGi®* (open services gateway initiative. See 12.3 & 12.4) *TARV-ROAM* environment. Deployment options shall be in accordance with ETSI TS 102 665 which defines various deployment options using this abstracted stack, and the interested reader is directed to those documentations for further information.

Intermittent connectivity is inherent in radio-based systems. Applications shall NOT assume that because data has been passed to the communications subsystem that it has been received at the far end of the link. This part of ISO 15638 is designed to use acknowledgements as far as possible. However, buffering and other techniques may be required, and such techniques are left to product design and are not specified in this part of ISO 15638.

Application support is defined by ETSI ITS as existing in the facility layer and these facilities will generally be provided by *ROAM* and/or third party components. Within this layer will also reside generic facilities such as the 'Local Dynamic Map' (*LDM*), 'Human/Machine Interface (*HMI*) Support', 'Discovery', 'Services Management', 'Priority Management', 'Message Queuing' and 'SOA (service oriented architecture)' protocol support.

12.5.3 TARV-ROAM service platform components

Figure 22 shows a *UML* (4.42) view of *TARV-ROAM* service platform components.

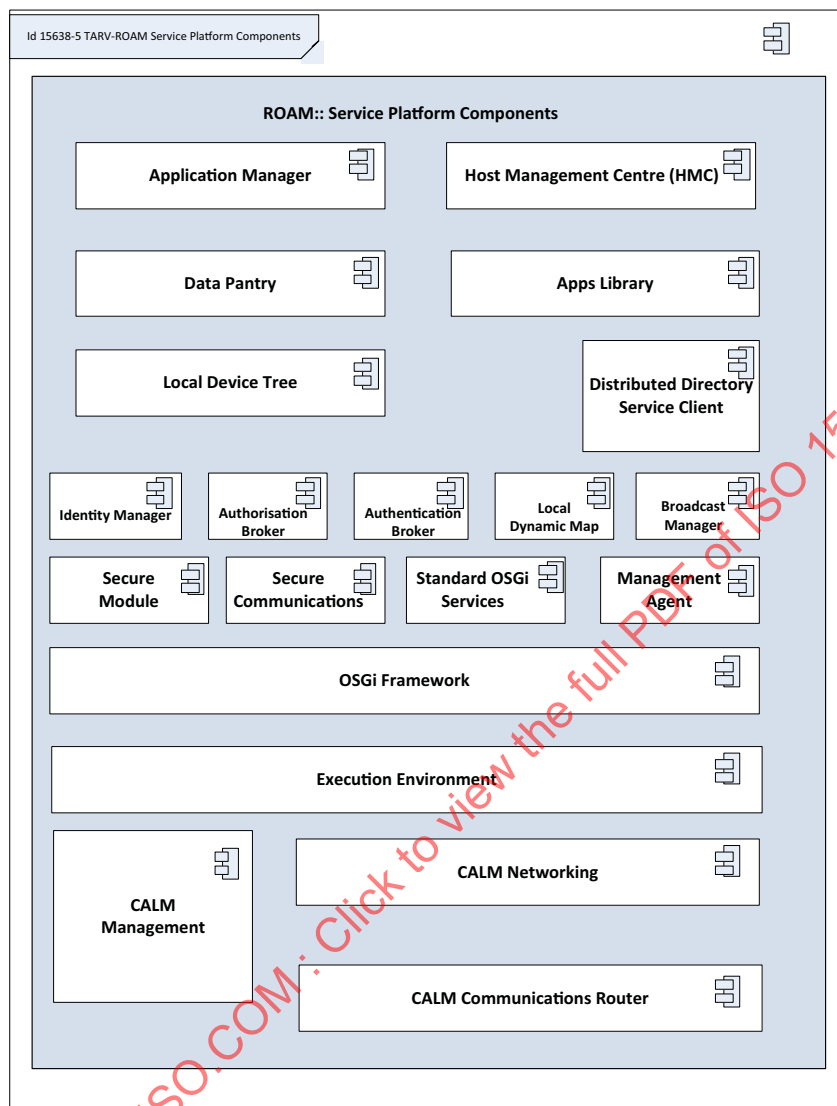


Figure 22 — TARV-ROAM service platform components

The components of Figure 22 are defined as:

12.5.3.1 OSGi® application manager

Part of the *OSGi®_serviceplatform* that provides a basic *HMI* to a *TARV user* (4.45).

12.5.3.2 OSGi® authentication broker

Part of the *OSGi®_serviceplatform* providing access to distributed authentication services. The distributed authentication service has to facilitate the use of single sign-on.

12.5.3.3 OSGi® authorization broker

Part of the *OSGi®_serviceplatform* providing access to distributed authorization services.

12.5.3.4 OSGi® data distribution service (DDS) client

Part of the *OSGi®_serviceplatform* responsible for providing access to the distributed directory service functionality.

12.5.3.5 OSGi® execution environment

Part of the *OSGi®_serviceplatform* responsible for providing a complete execution environment for *OSGi®* applications.

12.5.3.6 OSGi® framework

Part of the *OSGi®_serviceplatform*, responsible for providing class loading facilities, life cycle management and for maintaining a (local) 'service registry'. The *OSGi®_framework* is represented as the system bundle.

12.5.3.7 OSGi® identity manager

Part of the *OSGi®_serviceplatform* responsible for storing and leasing pseudonyms to an application service.

12.5.3.8 OSGi® local data tree

Part of the *OSGi®_serviceplatform* responsible for providing access to local device sensors and actuators through a tree structure.

12.5.3.9 OSGi® local dynamic map

Part of the *OSGi®_serviceplatform* responsible for providing access to the *OSGi®_localdynamicmap* (LDM). The LDM is a concept representing the traffic situation on the road network in the vicinity of the *TARV IVS* (4.23) (to be further defined later in C-ITS standards).

12.5.3.10 OSGi® management agent

Part of the *OSGi®_serviceplatform* that provides support for remote management. In certain cases, the management agent can be part of the *OSGi®_framework*.

12.5.3.11 OSGi® secure communication

Part of the *OSGi®_serviceplatform* responsible for providing secure communication services.

12.5.3.12 OSGi® secure module

Part of the *OSGi®_serviceplatform* used for *tamper* (4.41) evident operations, e.g. on cryptographic keys or sensitive data. It can provide the cryptographic functionalities required for secret and public key operations.

12.5.3.13 OSGi® standard services

Part of the *OSGi®_serviceplatform* providing the standard *OSGi® framework*, system, protocol and other services.

12.6 Host management centre (HMC)

The 'Host Management Centre' (HMC) is an extension to standard *OSGi®*. The HMC is the central point for *TARV-ROAM* management of *TARV* applications executing on the *TARV-ROAM* host. HMC enables remote management of vehicle applications by a trusted party. It is compatible with the development of an HMC concept by the project CVIS and is based on *OSGi®* MEG. The provisioning protocol is based on the OMA-DM protocol (Open Mobile Alliance – Device Management). See Figure 23 below.

The *HMC* provides two fundamental services:

- Service deployment - makes a service available in a *TARV-ROAM* system via a host management centre.
- Service provisioning - handles life cycle management of *OSGi*® based applications using remote management mechanisms, i.e. install (download), start, stop, update etc.

Limited support of native management is provided.

'Service Centres' provide applications to the *HMC*; the *HMC* is responsible for verification of the software and deployment on to the vehicle platform. Due to the potential amount of programs, this verification is automated.

EXAMPLE Vehicle applications may need additional services from the internet. For example, a traffic information application has a local component on the vehicle that informs the driver that updated traffic conditions can be downloaded; these additional services are also provided by a service centre.

New applications are required to pass verification by the *HMC* before they can be deployed. The vehicle platform enforces service or *user* (4.45) specific policies that cannot be enforced at verification time (e.g. inter application communication).

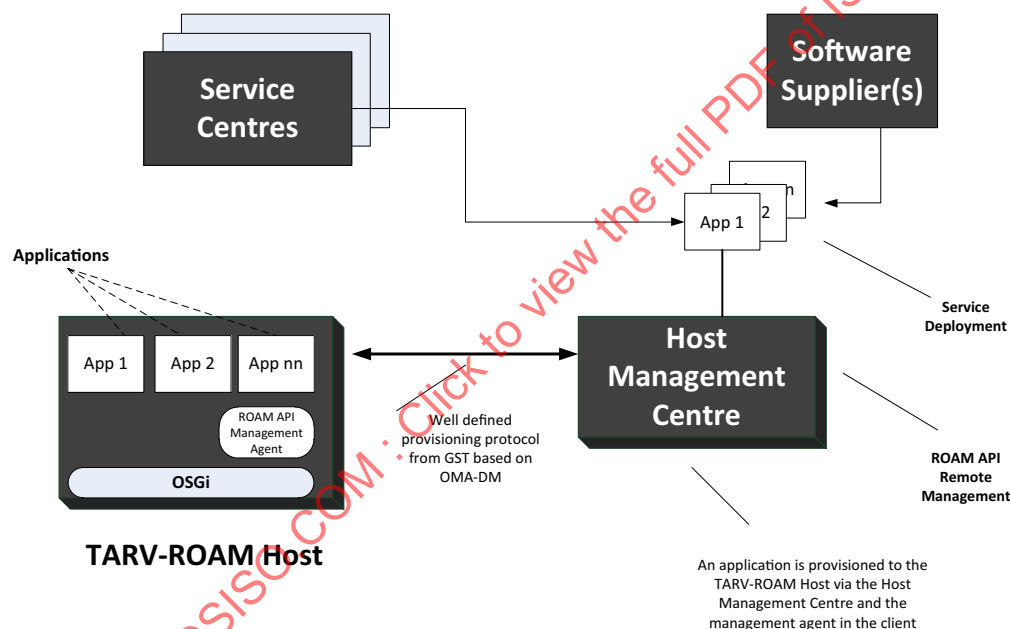


Figure 23 — HMC role in deployment and provisioning

The server-side *HMC* can be installed on an arbitrary server running UNIX® and JAVA®. The *HMC* also requires the PostgreSQL database to be installed (available as open source from <http://www.postgresql.org>).

NOTE PostgreSQL is a powerful, open source object-relational database system. It has more than 15 years of active development and a proven *architecture* (4.7). It runs on all major operating systems, including LINUX®, UNIX® (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows. It is fully ACID compliant, has full support for foreign keys, joins, views, triggers, and stored procedures (in multiple languages). It includes most SQL:2008 data types, including INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, and TIMESTAMP. It also supports storage of binary large objects, including pictures, sounds, or video. It has native programming interfaces for C/C++, JAVA®, .Net, Perl, Python, Ruby, Tcl, ODBC, among others.

An enterprise class database, PostgreSQL boasts sophisticated features such as multi-version concurrency control (MVCC), point in time recovery, tablespaces, asynchronous replication, nested transactions (savepoints), online/hot backups, a sophisticated query planner/optimizer, and write ahead logging for fault tolerance. It supports international character sets, multibyte character encodings, Unicode, and it is locale-

aware for sorting, case-sensitivity, and formatting. It is highly scalable both in the quantity of data it can manage and in the number of concurrent users it can accommodate.

Limit	Value
Maximum Database Size	Unlimited
Maximum Table Size	32 TB
Maximum Row Size	1.6 TB
Maximum Field Size	1 GB
Maximum Rows per Table	Unlimited
Maximum Columns per Table	250 - 1600 depending on column types
Maximum Indexes per Table	Unlimited

12.7 Local data tree (LDT)

12.7.1 General framework and architecture

The local data tree (LDT) provides a uniform view of the data associated with any vehicle and provides what is called 'Basic vehicle data (4.10)' (BVD) throughout the ISO 15638 suite of standards deliverables.

Two local data trees are defined separately within the ISO 15638 suite of standards deliverables. One (TARV LDT) in respect of TARV data that is specific to the application, the other (C-ITS LDT) in respect of general vehicle data that may be used in a wider range of applications across all classes of vehicle, for example in cooperative safety systems. Because of the cooperative nature of the C-ITS (4.14) LDT, this second class is deliberately identical to the CVIS project FOAM 'local device tree' (CVIS project 'local device tree' = 15638 TARV C-ITS 'local data tree'), and therefore compatible/interoperable with any applications/standards which become based upon it. The first tree is specific to TARV and commercial vehicle applications.

The behaviour of the management and update of the two local data trees are deliberately identical, although access rights are different.

Architecturally, the two 'trees' could be considered to be two 'branches' of a single class of data objects called 'tree'. However, in order to ease migration and simplify the introduction and use of cooperative systems for all classes of vehicles, without requiring modified provisioning, update and access for these systems if they are to work in TARVs, they are treated separately within ISO 15638, and therefore described as two instances of a class called 'local data tree'. There is some limited data duplication as a result, but it enables the C-ITS (4.14) LDT to be defined by reference and there is no great volume of data duplication as a consequence. This will allow the easy integration and take up of cooperative ITS (4.14) systems as they are developed, without their need for a special variant to need to be developed to be able to work in TARVs, or for the TARV specifications and LDT needing to be revised as the C-ITS (4.14) LDT becomes modified. At the same time, the two tree approach allows the development of TARV specific instantiations, many of which are likely to precede the instantiation of cooperative ITS (4.14) systems for all classes of vehicles (which in many cases is likely to be a much more protracted progress), without slowing the progress of instantiation of these TARV applications. It also allows the different timing for data provisioning/update without introducing further complexity.

The local data trees are organized in a tree structure. The means of provisioning both local data trees follows the same architecture (4.7) and will just use different 'apps' placed in the on-board 'app' library for their update.

The C-ITS (4.14) vehicle local data tree (called a local device tree within CVIS) is extensible.

Local data trees allow:

- To manage the device database (discover, expand or delete database elements) locally.
- To have different access rights for the TARV LDT and C-ITS (4.14) LDT, but in a manner that is easy to implement.

- To have different provisioning timings, and if necessary criteria, between the *TARV LDT* and *C-ITS (4.14) LDT*, while remaining within the same *architecture (4.7)* and interoperabilities.
- Optional: To manage or access the device database from a remote server by using the *OMA device management specification (4.40)*.

Some use cases identified for a local data tree *API* are shown in Figure 24.

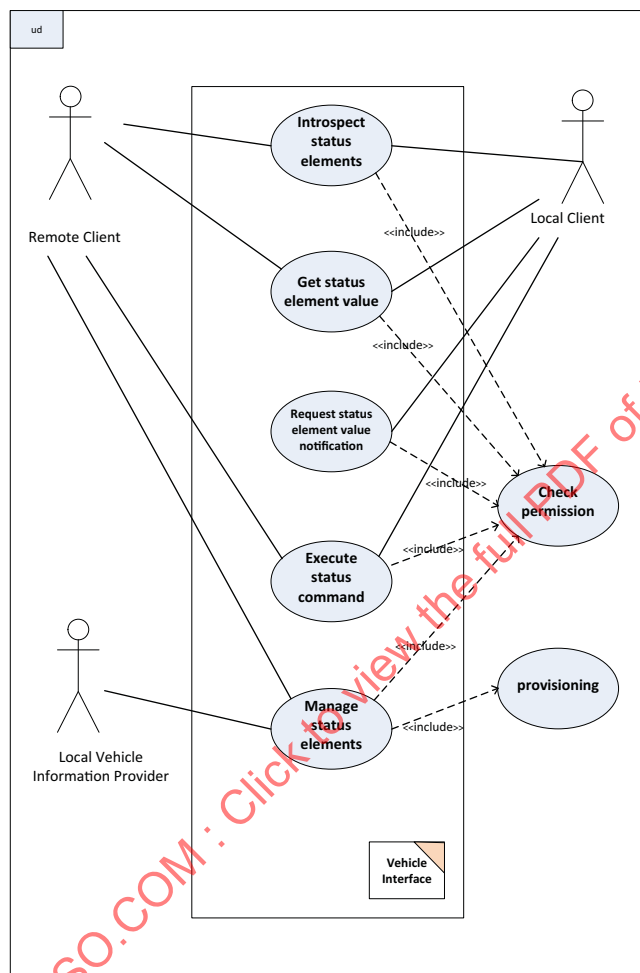


Figure 24 — Data/device tree use cases
(source: adapted from CVIS)

In obtaining data for a data tree, it is significant, that, in addition to data calculated within *TARV*, data will be required from the vehicle, and the coding schema of status information generated by electronic control units and sensors is likely to be proprietary. Therefore it will probably not be possible to harmonize all this across vehicle manufacturers. However, it is believed that, if a common set of status elements with status information of a defined type, can be specified at an application level, proprietary systems will be prepared in many cases to provide the data in that format, which will enable a vehicle configuration independent access for *TARV*, safety related and other applications. This is expected to be a greater problem for provisioning the *C-ITS (4.14)* local data tree than the *TARV* local data tree.

Data, even where agreed and interoperable, can also appear or disappear dynamically, e.g. a *GPS (4.22) /GNSS (4.21)* -sensor could be plugged in or unplugged from the client system. The interface has to provide a mechanism to handle this dynamic behaviour.

It shall also be a basic assumption that the same information will not be available in all vehicles. This implies that a discovery mechanism shall be supported which enables the client application to look up the available status items.

The set of status elements can be accessed remotely, e.g. from a backend server by using the OMA device management *specification* (4.40).

Figure 25 provides a view of the component *architecture* (4.7) of a DMT using the OSGi® OMA device management specification.

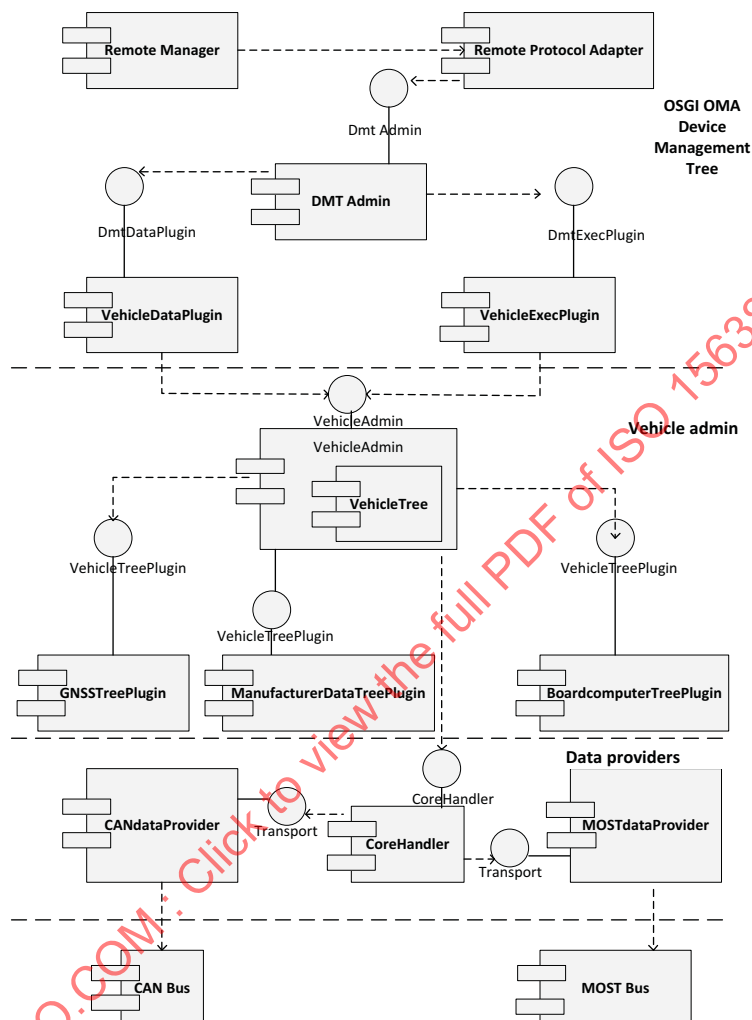


Figure 25 — OSGi® OMA device management tree
(Source :OSGi®)

See www.OSGi.org for details of how this can be achieved in greater detail.

This allows entities like a *TARV-ROAM* host management centre or an *application service* (4.2) provider to access status element information.

12.7.2 LDT actors

There are three principle actors involved with a local data tree:

- Local client
 - A software component running on the client system that wants to access device status elements. The local client could represent a front-end component of an application service.

- Local device information provider
 - Provides software components that contains information about the structure of the status element database and the mapping of the device data to this structure.
- Remote client
 - A software component running on the backend server that wants to access or manage device status elements.

12.7.3 LDT admin layer

The local data tree admin layer organizes the sensor data represented by status elements in a tree structure and provides the appropriate meta information.

The *LDT_Admin* stores access rights functionally. Information concerning access rights is provided by different '*LDT* plug-ins' (see below), responsible for the appropriate sub-nodes. *API* clients may execute only the methods allowed for their *UserID* in the access list. The exact rights management is dependent on the *LDT API* implementation.

If a client wishes to access status element in the tree, it shall first provide its *UserID* (string), and the path to the node it needs to access. It is also possible for clients to provide any path above the one they need to access, and then using the *API* methods to access the sub nodes, reach that required.

The data mapping (raw-data to tree-node) and tree structure creation functionality is distributed between one or many so-called '*LDT* plug-ins'. Each *LDT* plug-in is responsible for the structure and the mapping of the status elements of its sub tree nodes.

If a client wishes to extend or change the tree structure and perform the data mapping, it shall first register a tree plug-in interface in the *OSGi*® service registry. At this point the *LDT* admin is notified and creates the appropriate sub node for the plug-in in the data tree. The *LDT* plug-in receives then the newly created sub node (its root node) and creates its own structure.

12.7.4 LDT admin interface

The *LDT_Admin* interface is the entry point for applications accessing device data from the tree. Using the *LDT_Admin*, an application may request the tree node, providing its path. As in a typical tree-like data structure, using the *API*, nodes can be accessed by calling the *getChildNode* method.

12.7.5 LDT plug-ins

LDT plug-ins are responsible for organizing the status elements into a tree structure. Each plug-in contains the necessary mapping information for its sub tree structure and the status element names, which it maps on this structure. *LDT_Admin* is responsible to provide each plug-in with its root node instance.

The root node *uniform resource identifier (URI)* (4.43) is provided by the plug-in. After receiving the sub tree root node, the plug-in can extend the tree according the specific mapping logic it provides.

12.7.6 Data providers layer

The data providers layer offers a "flat" data container, which handles references to all of the status elements.

Data providers store no information about matching their data into a structure. The status element value at this level is accessed using a *cell_Id*.

In order to access data, clients have to access the appropriate *cell_Id* data container, and the layer provides them with a so-called "Cell-object", which encapsulates the status element value.

The layer provides also mechanisms for sending messages for changing status element values using the underlying device bus.

12.7.7 Corehandler

The `corehandler` module acts as a broker, which forwards the data between the `CellProvider` modules and the clients requesting data.

12.7.8 CellProvider

The `CellProvider` module provides the access to the lowest layers of the device sensors interface. Modules implementing the `CellProvider` interface are responsible for translating the specific device data coming from a device bus into an abstract bus independent format (the so-called “cells”) and vice-versa (a cell object to a specific bus format).

12.7.9 LDTNode

The `LDTNode` represents a single node in the data tree. Internal device nodes have properties (meta-data) associated with it; leaf nodes are representing status elements containing a status element value. It is possible to traverse the whole tree, using the root node and the `getChildrenNodes()` method. `LDT_Admin` is responsible to provide the *API* clients, requesting a node with a `LDTNode`.

12.7.10 Access control

Different clients have different access rights for the node.

This can be achieved by different proxy instances to the same `LDTNode` for different users. Thus, the *user* (4.45) holds reference to a proxy, and there are different proxies for the different users, all pointing to the same `LDTNode`.

The configuration of each proxy for the different users or *user* (4.45) types is achieved using *OSGi®* `UserAdmin`. The `LDTAdmin` defines the users and roles, and the relations between them, and the plug-ins the mapping between the roles and the concrete permissions at node level for the different nodes they provide. The role of the proxy is to make a permission check (using the `UserAdmin` for example) before executing a method of the `LDTNode`. The relation Proxy - `LDTNode` is 0 or more to 1. (See *OSGi®* `UserAdmin` for details www.osgi.org).

12.7.11 LDTAdmin

API clients use this interface in order to request access to a certain node with a certain *uniform resource identifier* (URI) (4.43) in the data tree. Based on the `userID`, the `LDTAdmin` implementation is responsible to provide clients with different implementations of the `LDTNode` interface, i.e. for `userIDs` with different access rights there must be different method access. This is achieved by wrapping each `LDTNode` implementation with another class, also implementing the tree interface, and support different instances of the wrapper for the different users of the *API*.

12.7.12 LDTNodeValueListener

A node leaf contains a value of the status element in the form of a cell object, which encapsulates the raw value. In the event that a client wishes to update the value of a status element, it shall first register as a listener and receive the updated values.

12.7.13 LDTPlugin

Modules containing information about the structure of a sub tree and the mapping between the structure and the status elements must implement the `LDTPlugin` interface and register it as an *OSGi®* service. Thus they provide the `LDTAdmin` service with information about the absolute path to their root, and will receive their

root node as a *LDTNode* instance. After the *LDTAdmin* creates the tree plug-in root node, and passes it to the plug-in, the plug-in has the opportunity to create its tree structure upon its own root node.

12.7.14 Device node command

If an *API* client wants to execute a method (for example set value) on a certain node, it shall provide it with a *DeviceNodeCommand*, containing the name of the methods and the parameters. Invoking a *DeviceNodeCommand* (in case such is supported) which normally causes the data providers layer to establish communication to the appropriate device node, accessible over a device bus.

12.7.15 Core handler

Using the OSGi® data providers entry point interface (OSGi® service), clients may register listeners, request and send cells.

12.7.16 CellProvider

Each status elements provider shall register a *CellProvider* interface in order to provide it cells to the *LDTAdmin*, and to those clients, which execute the *getCell()* method of the *LDTNode* interface. The *CellProvider* module provides the access to the lowest layers of the device sensors interface. Modules implementing the *CellProvider* interface are responsible for translating the specific device data coming from a device bus into an abstract bus independent format (the “cells”) and vice-versa (a cell object to a specific bus format).

The interface provides information about the identification(s) of the cell, for adding and removing a listener, requesting and creating a cell.

12.7.17 Cell listener

Provides a listener interface for receiving cell updates.

With the method “*usesCellAfterNotify*” the listener has the possibility to provide the information that it will not use the cell instance after the *notify* method returns, and thus the appropriate *CellProvider* module can reuse the instance of this cell. This option shall be used only in order to save instances generation in cases of very frequent update of the status, and only when the listener implementation really does not use the cell instance after returning from the *handleCell()* method in order to avoid conflicts.

NOTE The reason for allowing this technique is the CVIS project experience with reading some data provided by a CAN device bus, which is updated intensively. Creating a new object for each status update turns the virtual machine garbage collector active permanent to almost 100%, which slows down the performance of the whole virtual machine. Reusing instances in the same scenario speeds up the performance significantly.

12.7.18 Cell

A cell is an abstract basic class for the different data containers.

12.7.19 Data cell

This is probably the most used data container in the *LDT API*. It contains the physical value of the status data.

12.7.20 Complex cell

These are device status elements, which contain more than one value.

An example for such a status element is the *GNSS* (4.21) position. The position contains longitude, latitude, and height. A position measurement (current position) could also contain current heading and current speed.

An update for change in the position must be effected simultaneously, and notification to clients shall be of the whole set of data, not individual elements. This is identified as complex data and the element representing this data according to the current *architecture* (4.7) is the `ComplexCell`. The `ComplexCell` is a container for data cells, i.e. each of the elements part of the complex data, is represented by a cell. Cells in the complex data cell are accessed by their name or by their index.

12.7.21 String descriptor

Represents string status value. Such a status value could be the VIN, or registration number of a vehicle.

12.7.22 Measurement

A measurement object can be requested for each status value, represented as a data cell. Measurement objects are part of the *OSGi® specification* (4.40).

EXAMPLE A measurement object is used to maintain the joining of value, error, unit and time-stamp. The value and error are represented as doubles and time is measured in milliseconds since midnight, January 1, 1970 *UTC*.

Mathematical methods shall be provided that correctly calculate taking the error into account. A runtime error will occur when two measurements are used in an incompatible way e.g., when a speed (m/s) is added to a distance (m). The measurement class correctly tracks changes in unit during multiplication and division, always coercing the result to the simplest form. S.

Errors in the measurement class are absolute errors. Actual values are required to fall in the range value +/- error 95% .

Measurement objects are optional and not used instead of the `DataCell`, but in addition to improve performance.

12.7.23 StateDataCell

Represents a discrete status value. The *API* allows an *OSGi®* state object to be created from the current `StateDataCell` value.

12.7.24 State

State object can be requested for each `StateDataCell` value. State objects are optional, and are not used instead, but additional to the `StateDataCells` for performance reasons. State objects are part of the *OSGi® specification* (4.40).

12.7.25 Remote method cell

Used for sending a message, or in the case that the device has the necessary intelligence, also for executing methods remotely. Because the message format is dependent on the device bus, it is impossible for the *API* clients to request the `RemoteMethodCell`. In order for a client to send a message to a device unit it shall use the data providers layer to create the cell, and then only to set the method parameters.

12.8 TARV supported LDTs

This version of TARV supports two *LDT* essential data concepts

- 1) TARV LDT;
- 2) C-ITS LDT.

The *TARV LDT* essential data is focussed to the specific requirements for regulated commercial freight vehicles.

The *C-ITS (4.14)* (co-operative vehicle systems) *LDT* essential data is focussed to the requirements for cooperative vehicle systems for all classes of vehicle, and is safety application centric.

Figure 26 provides a representation of the *C-ITS (4.14)* local data tree.

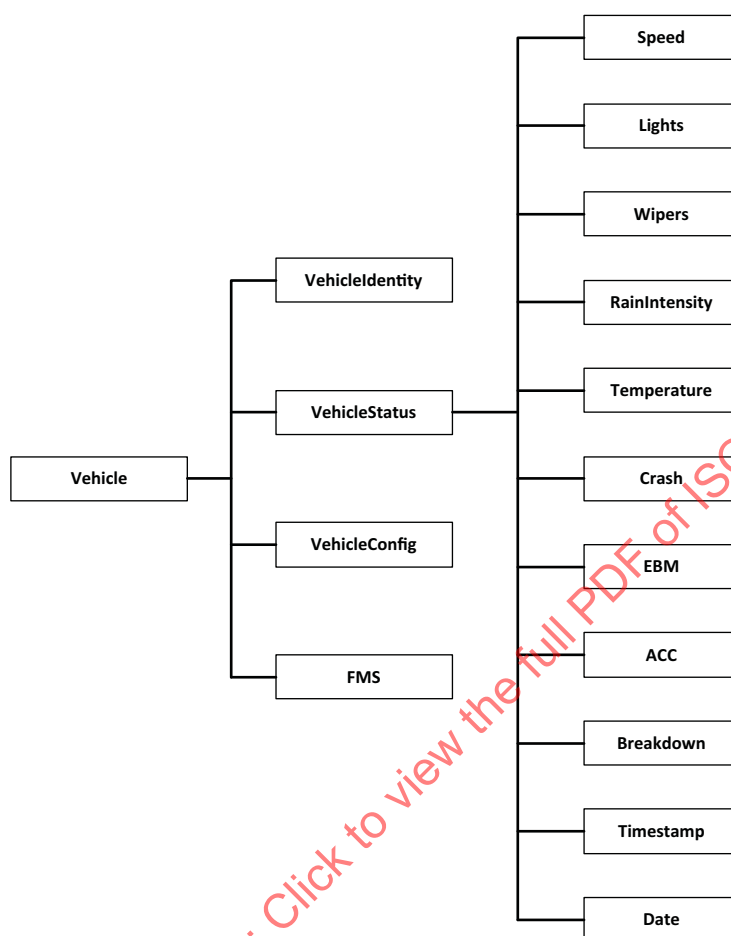


Figure 26 — C-ITS local data tree

The content of the *C-ITS (4.14)* local data tree may be revised by subsequent International Standards for cooperative vehicle systems. This can be easily achieved by installing a revised 'app' in the on-board 'app' library. The example used above is that used for project CVIS.

Figure 27 shows the *TARV LDT*.

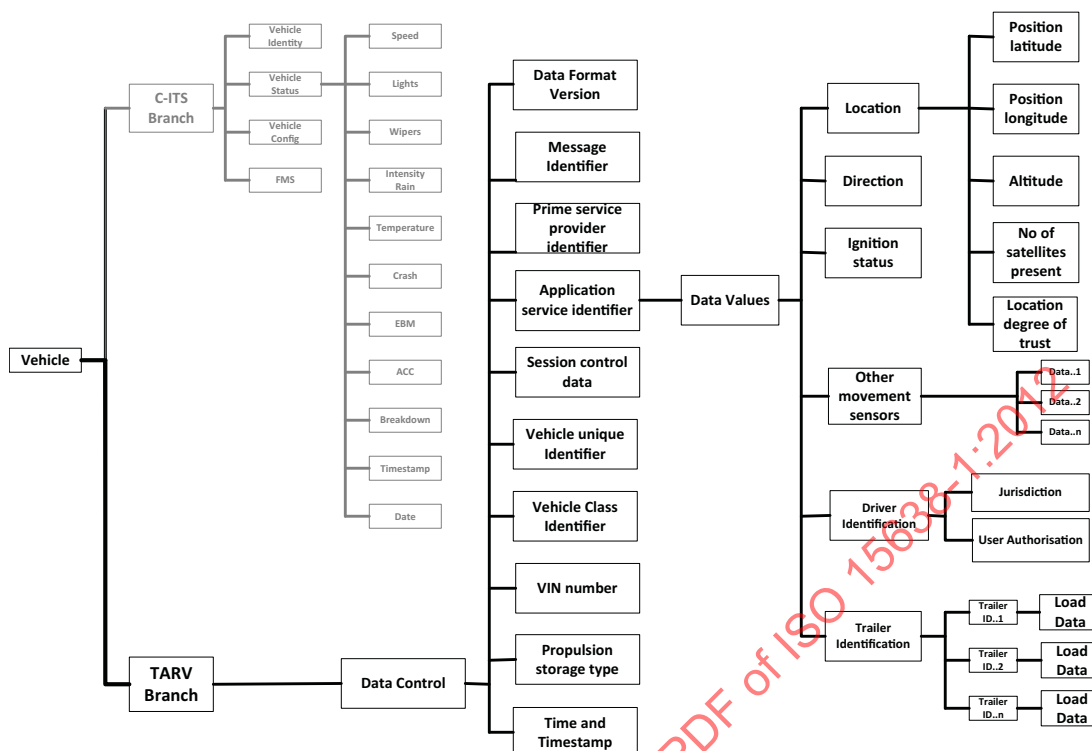


Figure 27 —TARV-ROAM local data tree

12.9 Distributed directory service (DDS)

The 'distributed directory service (DDS)' provides a discovery function for *TARV* and *C-ITS* (4.14) systems. This provides 'Yellow Pages' capability to the system.

The *DDS* allows applications to search for other applications based on predetermined criteria.

EXAMPLE It may seek to identify what 'apps' are available from roadside or other vehicles in the specific geographic area. In the case of *TARV*, this facility can be used to find the core data requirements of the local *jurisdiction* (4.24) for its own vehicle class. It could be used to instigate management procedures for dangerous or notified freight passage, or instigate automated customs clearance etc.

Detailed operation of the distributed directory service shall be by reference to a (yet to be published) cooperative vehicle system standard for the distributed directory service.

12.10 Typical use case examples

Figures 28 to 31 provide some typical use case examples.

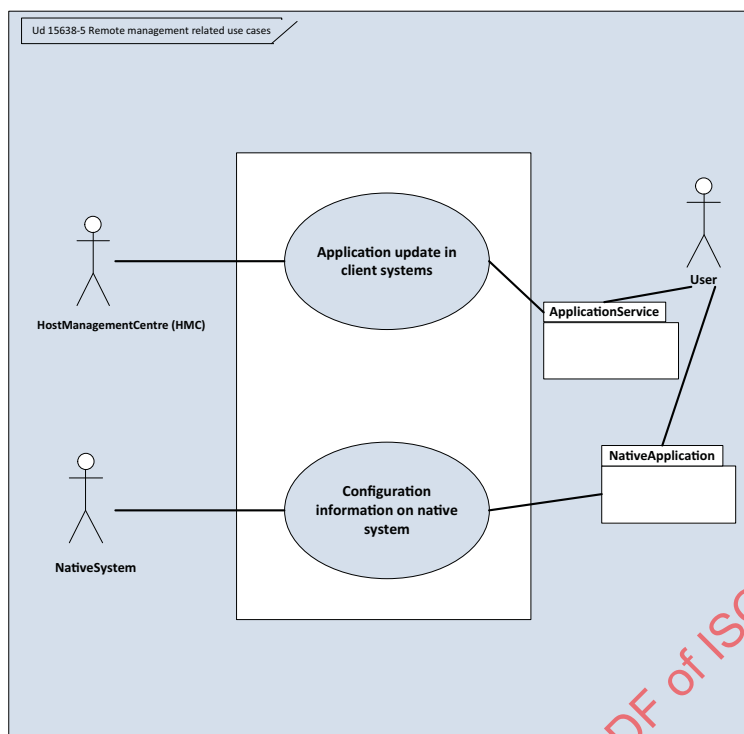


Figure 28 — TARV - ROAM remote management related use cases

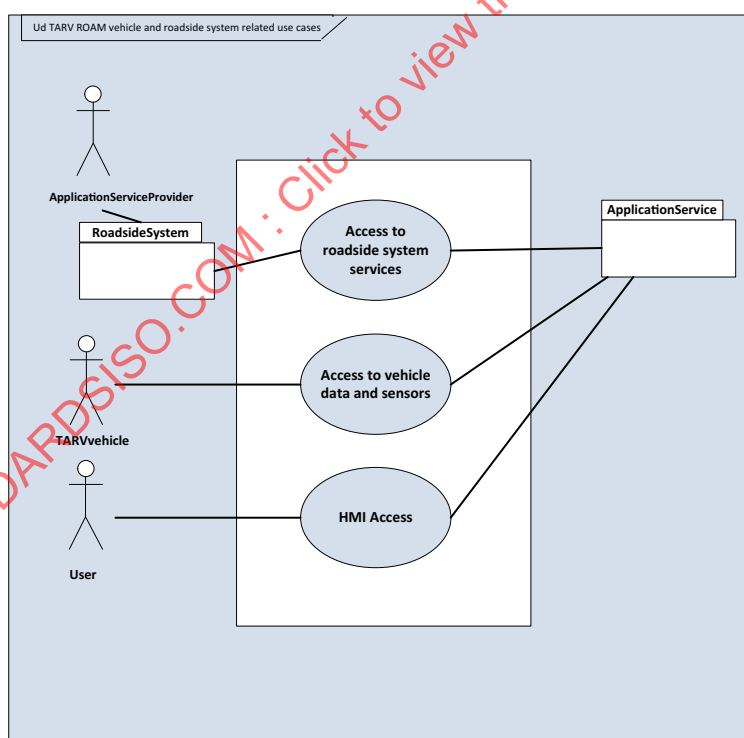


Figure 29 — TARV - ROAM vehicle and roadside system related use cases

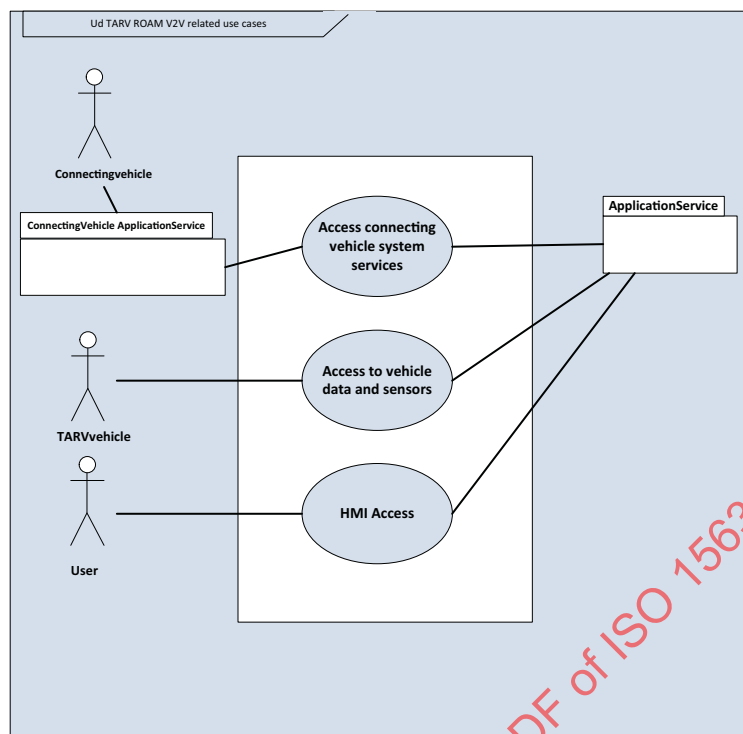


Figure 30 — TARV – ROAM V2V system related use cases

Although high level and *architecture* (4.7) aspects of security are included in *ROAM*, detailed security provisions are specified in ISO 15638-4. The high level security aspects can be envisaged as shown in Figure 31.

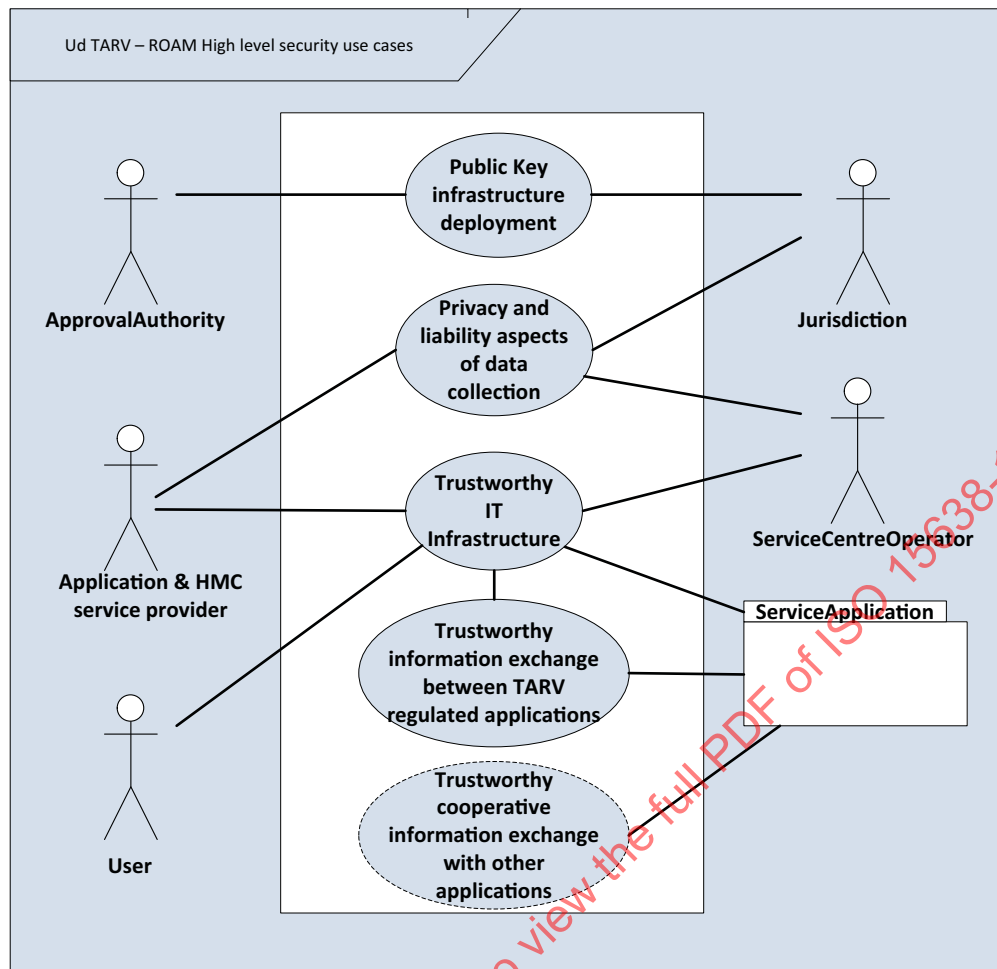


Figure 31 — TARV - ROAM High level security level use cases

13 Privacy issues

13.1 General issues of privacy

Issues of privacy are less sensitive in the case of *TARV* than for non commercial applications. It is recommended that users address issues of privacy in their employment contracts with drivers, making explicit what driver related data (both direct and indirect) will be collected and used and the circumstances in which that data will be used, and declaring any third parties which may have access to such data; and that the drivers assent to this in the contract is obtained. The details of such provisions are outside the scope of this part of ISO 15638.

13.2 Personal privacy

Personal privacy protection shall be protected in accordance with the local data privacy regulations applicable within any *jurisdiction* (4.24) and shall respect International privacy regulations, which can be found in ISO TR 12859 (Data privacy in ITS service provision).

Jurisdictions (4.24) may impose regulation and *audit* (4.8) to protect personal privacy. Such measures are not defined within this part of ISO 15638. For the international requirements for personal privacy protection in *ITS* systems the reader is directed to ISO TR 12859 (Data privacy in ITS service provision).

13.3 Commercial privacy

Jurisdictions (4.24) shall keep private and shall not make public, nor available to any other enterprise or organisation, any data collected via *TARV*, except in the fulfilment of local regulation within the *jurisdiction*.

The general provisions for the protection of privacy within *cooperative ITS* (4.14) systems shall be provided, and is provided, via ISO 15638-2.

13.4 Communications privacy

Interoperability with other *cooperative ITS* (4.14) services is enabled within this part of ISO 15638, the general provisions for the protection of privacy within *cooperative ITS* systems shall be provided, and is provided via ISO 15638-2.

ISO 15638-2 provides:

- location privacy (outside of *TARV* service provision);
- anonymity against unauthorized parties;
- Preservation of resolvable pseudonymity for security reasons;
- accountability and non-repudiation.

Additionally, the privacy of commercially sensitive data transmitted over wireless networks shall be protected, and is protected via the provisions of ISO 15638-2. (see ISO 15638-2)

13.5 TARV-ROAM privacy

In respect of *TARV-ROAM*, at the time of developing this part of ISO 15638 there are no International Standards than can enable this part of ISO 15638 to provide a *specification* (4.40) simply by reference to a referenced International Standard. Until such an International Standard for *cooperative ITS* (4.14) systems is published, the following privacy aspects shall be taken into consideration in any *TARV* instantiation.

- | | |
|-------------------|--|
| ROAM-PRIVACY-0001 | Privacy and liability: the <i>TARV</i> infrastructure shall contain flexible operation data capability to allow for monitoring and strict partitioning. |
| ROAM-PRIVACY-0002 | PKI deployment: <i>TARV</i> deployment enablers shall define the <i>conditions</i> (4.16) for PKI deployment. |
| ROAM-PRIVACY-0003 | <i>User</i> (4.76) authentication: the <i>TARV</i> client system shall provide end-user authentication capability. |
| ROAM-PRIVACY-0004 | Single sign-on (SSO): the <i>TARV</i> infrastructure shall provide SSO. Hence a single authentication of an end-user (4.45) to the client system is sufficient to use subscribed services offered by (in general different) <i>service providers</i> (4.40). |
| ROAM-PRIVACY-0005 | Circle of trust: the <i>TARV</i> infrastructure shall ensure that different business stakeholders can share data only in a trusted manner. |
| ROAM-PRIVACY-0006 | Scalable trusted communication: the <i>TARV</i> infrastructure shall support scalable authenticated communication. |
| ROAM-PRIVACY-0007 | V2V / V2I authentication: real-time constraint for authentication and communication shall be met. |

ROAM-PRIVACY-0008	Trusted execution: the <i>TARV</i> client system shall ensure separation of executions assets (i.e. an external application cannot access other application data) including protection of certificates.
ROAM-PRIVACY-0009	Resources management: The <i>TARV</i> client system shall ensure that no application can use more resources (e.g. CPU, memory, ...) than declared ahead.
ROAM-PRIVACY-0010	Resources declaration: <i>TARV</i> applications must declare resources needed in advance of any specific transaction.
ROAM-PRIVACY-0011	Trusted update: the <i>TARV</i> client system shall ensure that only trusted components are downloaded, and that applications are guaranteed some level of execution (to prevent from denial of service).
ROAM-PRIVACY-0012	Trusted policy management: the <i>TARV</i> infrastructure shall ensure that executing platforms support features to (1) decide policies, (2) enforce policies, and (3) enforce policies in a protected way.
ROAM-PRIVACY-0013	Assurance of integrity: <i>ROAM</i> shall provide a mechanism to assure the integrity of communicated messages between <i>TARV</i> nodes.
ROAM-PRIVACY-0014	Assurance of confidentiality: <i>ROAM</i> shall provide a mechanism to assure the confidentiality of communicated messages between <i>TARV</i> nodes.
ROAM-PRIVACY-0015	Broadcast support: <i>ROAM</i> shall provide a mechanism to distribute information through a broadcast medium in such a manner that it can only be read/deciphered by authorised clients.
ROAM-PRIVACY-0016	Application management: <i>ROAM</i> shall intermediate between the different <i>TARV</i> applications that request access to the <i>HMI</i> in a way that road safety is not compromised.
ROAM-PRIVACY-0017	Communication with external applications The vehicle interface shall guarantee secure communications with external applications introducing appropriate mechanisms (firewall).
ROAM-PRIVACY-0018	RE-Identification: <i>ROAM</i> shall provide a mechanism to identify communication partners for a certain time interval without violating privacy requirements. This shall support to individually communicate between the same partners during the time interval without revealing more details about its identities.
ROAM-PRIVACY-0019	Vehicle Identification: <i>ROAM</i> shall provide a mechanism for applications to use identifications which safeguard personal privacy needs on one hand side but allow data sharing for applications on basis of a general profile (e.g. vehicle type, driver category, temporary unique <i>ID</i> for a given maximum duration).

14 Quality of service requirements

This part of ISO 15638 contains no specific requirements concerning quality of service. Such aspects will be determined by a *jurisdiction* (4.24) as part of its *specification* (4.40) for any particular *regulated application service* (4.36).

15 Test requirements

There are no test requirements within this part of ISO 15638.

16 Marking, labelling and packaging

This part of ISO 15638 has no specific requirements for marking labelling or packaging. However, where the privacy of an individual may potentially or actually be compromised by any instantiation based on the ISO 15638 family of standards, the contracting parties shall make such risk explicitly known to the implementing *jurisdiction* (4.24) and shall abide by the privacy laws and regulations of the implementing *jurisdiction* and shall mark up or label any contracts specifically and explicitly drawing attention to any loss of privacy and precautions taken to protect privacy. Attention is drawn to ISO TR 12859 in this respect. See also Clause 13.

17 Declaration of patents and intellectual property

This part of ISO 15638 contains no known patents or intellectual property other than that which is implicit in the media standards referenced in ISO 15638-2 or open specifications as recognised by the display of '©'. While the *CALM* standards themselves are free of patents and intellectual property, *CALM* in many cases relies on the use of public networks and IPR exists in many of the public network media standards. The reader is referred to those standards for the implication of any patents and intellectual property.

Application services (4.2) specified within ISO 15638-6 and ISO 15638-7 contain no direct patents nor intellectual property other than the copyright of ISO. However, national, regional or local instantiations of any the applications services defined in ISO 15638-6 and ISO 15638-7, or of the generic vehicle information defined in ISO 15638-5, the security requirements contained in ISO 15638-4, or the requirements of ISO 15638-3, may have additional requirements which may have patent or intellectual property implications. The reader is referred to the regulation regime of the *jurisdiction* (4.24) and its regulations for instantiation in this respect.

Annex A (Informative)

International examples of regulated services

A.1 General

The following examples are examples of implementations, tests and trials of *TARV* technology or technology with *TARV* potential, currently implemented, or under test and trials around the world, prior to ISO 15638.

NOTE The following examples are presented as submitted to us by the relevant agencies, as examples, in the format they consider most appropriate, or the format of the legislation they provide. Beyond the high level decimal numbering of the title, the form and layout of this Annex is not therefore consistent with normal ISO layout formatting, but is reproduced in the form that it was provided as an example.

A.2 Example Japan

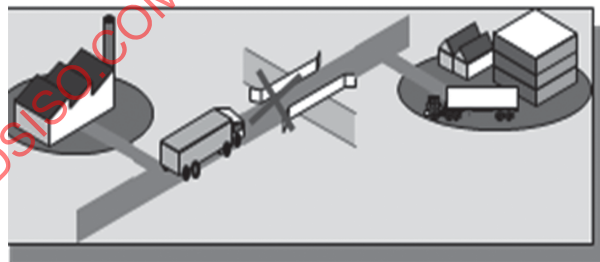
A.2.1 An overview concerning the legislation of road traffic permits for over-standard vehicles

A.2.1.1 General

Roads have been built for enabling certain standardized vehicles to go through safe and smoothly.

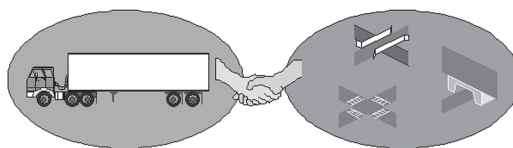
Therefore, Over-standard vehicles are unable to go through roads, because any potential negative impact from them will be an obstacle for traffic or damage of road structures.

(The Road Law Article 47, Clause 1, 2, The Vehicle Restriction Government Ordinance Article 3, Clause 1)



However, over-standard vehicles should be compelled to have any possibility to go through roads in certain instances, for social and economic activities in practical terms.

On this occasion, these vehicles need to accord with road structures, because roads have a crucial role to support such social and economic activities and also are fundamental infrastructures to be protected.



For this reason, over-standard vehicles are given permits to go through specified roads in the restricted case that road administrators accept them to be compelled to do so.

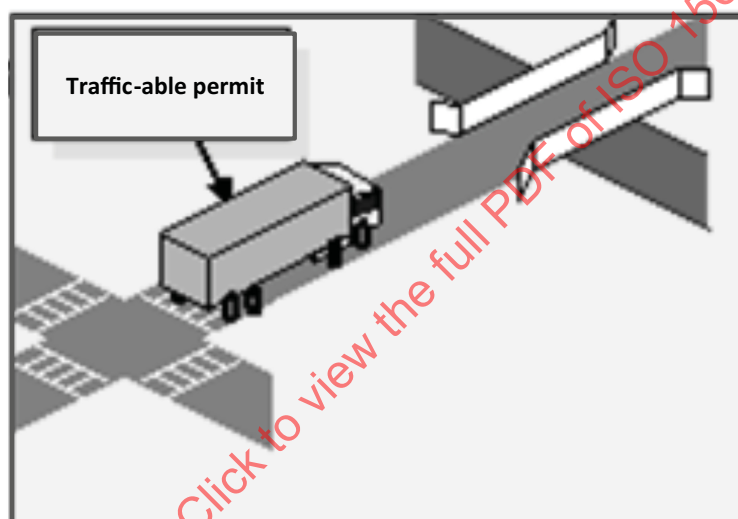
In order to issue permits, the specifications of the vehicle and the load on board have to be checked, then necessary conditions to maintain road structures or prevent traffic danger should be attached to the permit.

That is the legislation of road trafficable permits for over-standard vehicles.

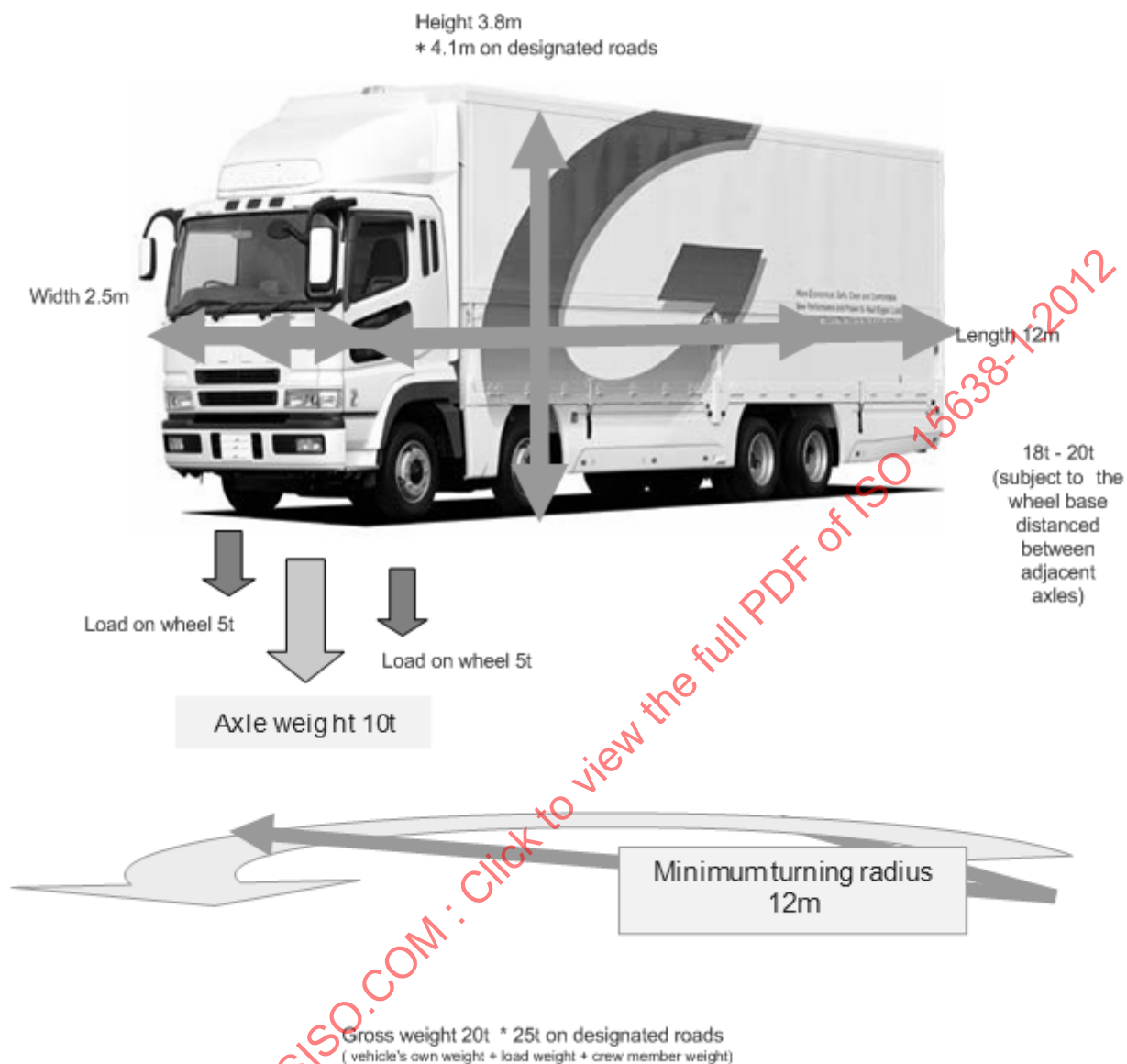
(The Road Law Article 47 part 2).

NOTE On the occasion to permit over-standard vehicles to go through bridges or overpasses, any specified conditions for reducing impact from the over-standard vehicle or eliminating other vehicles, have to be attached to such permit, so that total gross weight on the bridge should be brought as close to the designed weight of such bridge as possible.

For example, the condition enforces the over-standard vehicle to go on a crawl, prohibits from going through together with other vehicle(s) or deploys the escort vehicle(s) to lead such vehicle. Or, obliges the over-standard vehicle to go through at night, because of less traffic amount than daytime, by designating the trafficable time schedule described in the permit.

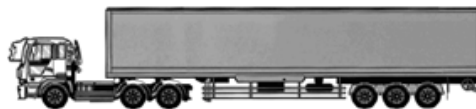


A.2.1.2 Specified limitations on size and weight of vehicles, based on The Vehicle Restriction Government Ordinance Article 3, Clause 1



Typical over-standard vehicles

【Van Type】



【Tank Type】



【Car Carrying Type】



【Ocean freight container Transport Type】



【Heavy goods Hauling Type】



【Truck-mounted crane Type】



A.2.1.3 The online application system of road trafficable permits for over-standard vehicles

For the application to permit over-standard vehicles to go through specified roads, based on The Road Law Article 47 part 2, the particular computer system enables *applicants* (4.1) to make applications, submit applications and receive digital permits via internet with monitoring and operating PCs at their offices or home.

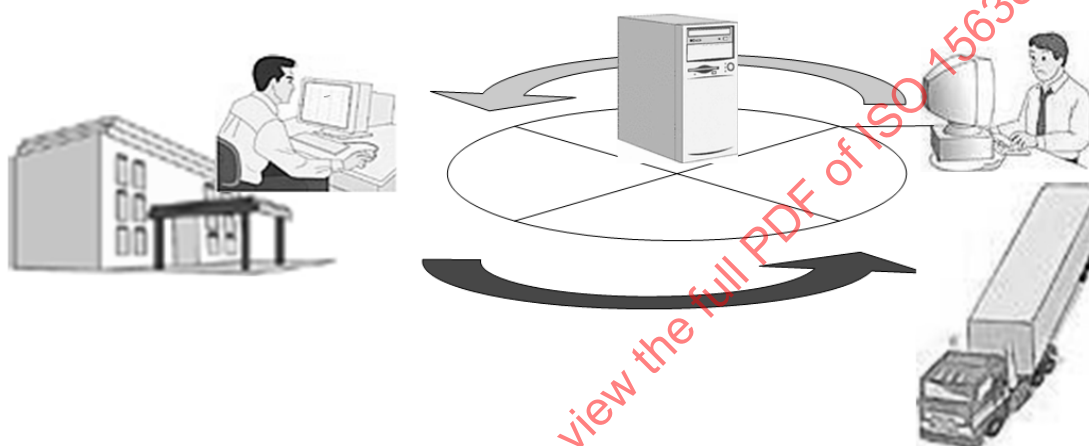
That is the online application system of road trafficable permits for over-standard vehicles.

Road Administrator

- examine contents of applications
- issue permits

Applicant

- make & submit applications
- receive digital permits



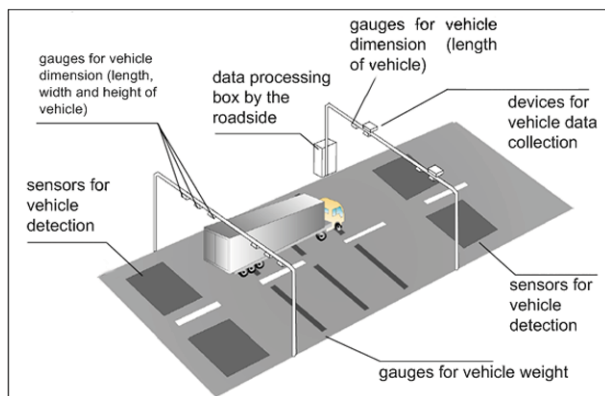
A.2.1.4 Monitoring vehicles for detection of non-compliance

By the implementation of monitoring vehicles, compliance consciousness has been increased, on the other hand, there exists run with non-compliance yet, for the reason of grudging requiring time to receive the permit or inability to comply with designated conditions.

Automatically-Gauging System Unit (Weigh In Motion)

The Weigh In Motion Unit (set up in 39 points along National Governance Roads domestically)

The Units have been implemented fully since Oct. 1, 2008.



The unit collects data from vehicles such as the weight of the vehicle in running condition.

Then non-existence of the permit, non-compliance of the vehicle or other will be detected.

The scene as a non-compliance vehicle detected



* detected over 12t of axle weight

Regulatory Enforcement in the field

Road Administrative Officers measure weight and dimension of the vehicle



Non-compliance suspicious vehicles are directed to the base placed by the roadside for regulatory and supervision.

And there should be implemented remediation directive or instructive caution to the assured non-compliance vehicle.

A.3 Example Australia Electronic work diaries

New South Wales Government and Transport Certification Australia

Operational pilot of

Electronic Work Diaries

Legislation in a number of Australian states currently requires that every commercial driver must keep a written or electronic work diary (EWD), recording hours of work and rest, when travelling more than 100km (200km for Queensland) from base. The operational pilot will help determine if EWDs are an effective improvement on the written work diary – not just for heavy vehicle drivers but all parties involved in Australian heavy vehicle transport.

Several commercial electronic record-keeping systems already exist, but EWDs need to collect information in a form that meets regulatory requirements. Using an EWD could also save time in the truck or bus, on the roadside and in the office.

What is the objective of the EWD Operational pilot?

The pilot will contribute to the national process of testing and refining:

- the proposed national draft policy;
- the technical specification for the approval of electronic systems;
- the use of electronic systems for enforcement and business purposes.

The pilot will also help identify better safety, productivity and environmental outcomes.

Decisions have to be made before a system can be rolled out, and of course as with anything new, there are issues and questions which largely revolve around who will perform various roles, some technical and operational elements and the eventual use of EWD data. The pilot will test various systems and operations to resolve these issues so that the EWD policy and specification can be finalised.

What is an Electronic Work Diary (EWD)?

An EWD is an electronic version of the written work diary, used to record a heavy vehicle driver's work and rest hours as required by the Heavy Vehicle Driver Fatigue Legislation in Australia.

An EWD comprises three components:

- a) In-Vehicle Unit (IVU);
- b) Driver Recording Device (DRD);
- c) Record Keeper function.

An EWD may also offer additional commercial and *regulatory* (4.38) features such as rest stop alerts, navigation and dispatching services. This is dependent on what the provider is offering.

The EWD records information required by Heavy Vehicle driver fatigue regulations, using an in-Vehicle unit and a driver Recording device (for example, a removable USB memory stick allocated to each driver). The EWD is an option to a written work diary.

Which governments are involved in this pilot?

All jurisdictions are supportive of the pilot, with transport agencies from New South Wales, Queensland, South Australia, Victoria, Western Australia and the Australian Government participating in the pilot.