ETSI EG 202 798 V1.1.1 (2011-01)

ETSI Guide

Intelligent Transport Systems (ITS);
Testing;
Framework for conformance and interoperability testing

Reference

DEG/ITS-0020022

Keywords

ITS, testing, conformance, interoperability

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This ETSI Guide (EG) has been produced by ETSI Technical Committee Intelligent Transport System (ITS).

The present document assumes that the reader has basic knowledge in testing as presented e.g. in [i.4], [i.7].

Introduction

"Intelligent Transport Systems" (ITS) are systems to support transportation of goods and humans with information and communication technologies in order to efficiently and safely use the transport infrastructure and transport means (cars, trains, planes, ships). With a focus on road transportation, elements of ITS for global applications are standardized in various standardisation organisations, both on an international level at e.g. ISO TC204, and on regional levels, e.g. in Europe at ETSI TC ITS and at CEN TC278 [i.13], [i.22].

The importance of ITS for the regional and the international market is expressed by the large number of activities of stake-holders, within regional research projects, industry initiatives and regional and international standardisation.

In Europe, the urgent need for ITS standards and the related test standards is expressed by the new mandate M/453 of the Commission of the European Union [i.36].

Mandate M/453 [i.36] also is given in the context of international harmonisation, as expressed by the EU-US joint declaration of intent on research cooperation in cooperative systems [i.35].

ETSI is prepared to take over a leading role in this process towards harmonized ITS. A major effort is on conformance and interoperability testing, which is being prepared by the present document on "ITS testing framework". The protocol conformance and interoperability testing framework is essential for a systematic and consistent approach towards testing of globally applicable ITS communications equipment.

1 Scope

The scope of the present document is to support ITS projects on the development of test specifications for ITS base standards from ETSI, ISO, CEN and other "Standard Developing Organisations" (SDOs) by providing:

- An ITS testing framework for conformance testing.
- An ITS testing framework for interoperability testing.

The testing framework proposed in the present document provides guidance for development of conformance and interoperability test strategies, test systems and the resulting test specifications for ITS.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

Not applicable.

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI ETR 266: "Methods for Testing and Specification (MTS); Test Purpose style guide".

[i.2] ETSI ETS 300 406: "Methods for Testing and Specification (MTS); Protocol and profile conformance testing specifications; Standardization methodology".

[i.3] ETSI EG 201 058: "Methods for Testing and Specification (MTS); Implementation Conformance Statement (ICS) proforma style guide".

[i.4] ETSI ES 201 873 (all parts): "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3".

[i.5] ETSI EG 202 237: "Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT); Generic approach to interoperability testing".

[i.6] ETSI EG 202 810: "Methods for Testing and Specification (MTS); Automated Interoperability Testing; Methodology and Framework".

[i.7] ISO/IEC 9646 (all parts): "Information technology - Open Systems Interconnection - Conformance testing methodology and framework".

[i.8] ISO 10746 (all parts): "Information technology - Open Distributed Processing - Reference model".

[i.9] ISO 21210: "Intelligent transport systems - Communications Access for land Mobiles (CALM) - Non-IP networking".

[i.10] ISO 21213: "Intelligent transport systems - Communications Access for Land Mobiles (CALM) - 3 G".

[i.11] ISO 21214: "Intelligent transport systems - Communications Access for Land Mobiles (CALM) - IR".

[i.12] ISO 21215: "Intelligent transport systems - Communications Access for Land Mobiles (CALM) - M5".

[i.13] ISO 21217: "Intelligent transport systems - Communications Access for Land Mobiles (CALM) - Architecture".

[i.14] ISO 21218: "Intelligent transport systems - Communications Access for Land Mobiles (CALM) - Medium Service Access Points".

[i.15] ISO 24102: "Intelligent Transport Systems - Communications Access for Land Mobiles (CALM) - ITS station management".

[i.16] ISO 29281: "Intelligent transport systems - Communications Access for Land Mobiles (CALM) - Non-IP networking".

[i.17] ETSI TS 102 636: "Intelligent Transportation System (ITS); Vehicular communications; GeoNetworking".

[i.18] ETSI TS 102 637 (all parts): "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; multiple parts".

[i.19] ETSI TS 102 687: "Intelligent Transport Systems (ITS); Transmitter Power Control Mechanism for Intelligent Transport Systems operating in the 5 GHz range".

[i.20] ETSI TS 102 724: "Intelligent Transport Systems (ITS); Harmonized Channel Specifications for Intelligent Transport Systems operating in the 5 GHz frequency band".

[i.21] ETSI TS 102 860: "Intelligent Transport Systems (ITS); Classification and management of applications".

[i.22] ETSI EN 302 665: "Intelligent Transport Systems (ITS); Communications Architecture".

[i.23] ETSI TS 102 760-2: "Intelligent Transport Systems (ITS); Test specifications for Intelligent Transport Systems; Communications Access for Land Mobiles (CALM); Medium Service Access Points (ISO 21218); Part 2: Test Suite Structure and Test Purposes (TSS&TP)".

[i.24] ETSI TS 102 797-2: "Intelligent Transport Systems (ITS); Road Transport and Traffic Telematics (RTTT); Test specifications for Intelligent Transport Systems, Communications Access for Land Mobiles (CALM), Interface Manager (ISO 24102); Part 2: Test Suite Structure and Test Purposes (TSS & TP)".

[i.25] ETSI TS 102 859-2: "Intelligent Transport Systems (ITS); Testing; Conformance test specifications for Transmission of IP packets over GeoNetworking; Part 2: Test Suite Structure and Test Purposes (TSS&TP)".

[i.26] ETSI TS 102 868-2: "Intelligent Transport System (ITS); Testing; Conformance test specification for Co-operative Awareness Messages (CAM); Part 2: Test Suite Structure and Test Purposes (TSS&TP)".

[i.27] ETSI TS 102 869-2: "Intelligent Transport System (ITS); Testing; Conformance test specification for Decentralized environmental Notification Messages (DNM); Part 2: Test Suite Structure and Test Purposes (TSS&TP)".

[i.28] ETSI TS 102 870-2: "Intelligent Transport Systems (ITS); Testing; Conformance test specifications for GeoNetworking Basic Transport Protocol (BTP); Part 2: Test Suite Structure and Test Purposes (TSS&TP)".

[i.29] ETSI TS 102 871-2: "Intelligent Transport Systems (ITS); Testing; Conformance test specifications for GeoNetworking ITS-G5; Part 2: Test Suite Structure and Test Purposes (TSS&TP)".

[i.30] ETSI TS 102 981-2: "Intelligent Transport Systems; Test specifications for Intelligent Transport Systems; Communications Access for Land Mobiles (CALM); IP networking (ISO 21210); Part 2: Test Suite Structure and Test Purposes (TSS&TP)".

[i.31] ETSI TS 102 982-2: "Intelligent Transport Systems; Test specifications for Intelligent Transport Systems; Communications Access for Land Mobiles (CALM); IR (ISO 21214); Part 2: Test Suite Structure and Test Purposes (TSS&TP)".

[i.32] ETSI TS 102 983-2: "Intelligent Transport Systems; Test specifications for Intelligent Transport Systems; Communications Access for Land Mobiles (CALM); M5 (ISO 21215); Part 2: Test Suite Structure and Test Purposes (TSS&TP)".

[i.33] ETSI TS 102 984-2: "Intelligent Transport Systems; Test specifications for Intelligent Transport Systems; Communications Access for Land Mobiles (CALM); Architecture (ISO 21217); Part 2: Test Suite Structure and Test Purposes (TSS&TP)".

[i.34] ETSI TS 102 985-2: "Intelligent Transport Systems; Test specifications for Intelligent Transport Systems; Communications Access for Land Mobiles (CALM); Non-IP networking (ISO 29281); Part 2: Test Suite Structure and Test Purposes (TSS&TP)".

[i.35] EU-US Joint Declaration of Intent on Research Cooperation in Cooperative Systems, Washington, D.C., EC/DGINFSO and USDOT/RITA, November 2009.

NOTE: Available at: http://ec.europa.eu/information_society/activities/esafety/doc/library/us/joint_decl_on_coop_systems.pdf.

[i.36] Standardisation mandate addressed to CEN, CENELEC and ETSI in the field of information and communication technologies to support the interoperability of co-operative systems for intelligent transport in the european community, M/453, October 2009.

NOTE: Available at: http://ec.europa.eu/enterprise/sectors/ict/files/standardisation_mandate_en.pdf.

[i.37] ISO 16445: "Intelligent Transport Systems - Communications access for land mobiles (CALM) - Handover mechanisms".

[i.38] IEEE 802.11: "Telecommunications and information exchange between systems - Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications".

[i.39] ISO 16444: "Intelligent Transport Systems - Communications access for land mobiles (CALM) - Geo-routing".

[i.40] ISO 16440: "Intelligent Transport Systems - Communications access for land mobiles (CALM) - WAVE".

[i.41] ISO 16460: "Wave Intergration".

[i.42] ISO IS 24978: "Intelligent transport systems - ITS Safety and emergency messages using any available wireless media - Data registry procedures".

[i.43] ISO IS 15662: "Intelligent transport systems - Wide area communication - Protocol management information".

[i.44] ISO IS 24101 (all parts): "Intelligent transport systems - Communications access for land mobiles (CALM) - Application management".

[i.45] ISO IS 24103: "Intelligent transport systems - Communications access for land mobiles (CALM) - Media adapted interface layer (MAIL)".

[i.46] ISO IS 25112: "Intelligent transport systems - Communications access for land mobiles (CALM) - Mobile wireless broadband using IEEE 802.16".

[i.47] ISO IS 29282: "Intelligent transport systems - Communications access for land mobiles (CALM) - Applications using satellite networks".

[i.48] ISO IS 21216: "Intelligent transport systems - Wireless communications - CALM using millimetre communications".

[i.49] ISO IS 21212: "Intelligent transport systems - Communications access for land mobiles (CALM) - 2G Cellular systems".

[i.50] ISO IS 11776: "Light gauge metal containers - Non-round open-top cans - Cans defined by their nominal capacities".

[i.51] ISO IS 25111: "Intelligent transport systems - Communications access for land mobiles (CALM) - General requirements for using public networks".

[i.52] ISO IS 21213: "Intelligent transport systems - Communications access for land mobiles (CALM) - 3G Cellular systems".

[i.53] ISO IS 29283: "ITS CALM Mobile Wireless Broadband applications using Communications in accordance with IEEE 802.20".

[i.54] ISO IS 25113: "Intelligent transport systems - Communications access for land mobiles (CALM) - Mobile wireless broadband using HC-SDMA".

[i.55] ISO IS 18183: "Public broadcast reception".

[i.56] IEEE 1609.5: "Standard for Wireless Access in Vehicular Environments (WAVE) - Communication Manager".

[i.57] IEEE 1609.1: "Standard for Dedicated Short Range Communications (DSRC) Resource Manager".

[i.58] IEEE 1609.0: "Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) - Architecture".

[i.59] IEEE 1609.3: "IEEE Draft Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services".

[i.60] IEEE 1609.4: "IEEE Draft Standard for Wireless Access in Vehicular Environments (WAVE) - Multi-Channel Operation".

[i.61] IEEE 1609.2: "Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages".

[i.62] ETSI TS 102 731: "Intelligent Transport Systems (ITS); Security; Security Services and Architecture".

[i.63] ETSI TS 102 894: "Intelligent Transport System (ITS); Users & Applications requirements; Facility layer structure, functional requirements and specifications".

[i.64] ETSI EN 302 895: "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM) Specification".

[i.65] ETSI TS 102 723: "Intelligent Transport Systems; OSI cross-layer topics; Part 11: Interface between network and transport layers and facilities layer".

[i.66] ETSI TR 102 893: "Intelligent Transport Systems (ITS); Security; Threat, Vulnerability and Risk Analysis (TVRA)".

[i.67] ETSI ES 202 910: "Intelligent Transport Systems (ITS); Security; Identity Management and Identity Protection in ITS".

[i.68] ETSI ES 202 663: "Intelligent Transport Systems (ITS); European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band".

[i.69] ISO IS 13181 (all parts): "CALM Security".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in [i.7], [i.8], [i.13], [i.22] and the following apply:

conformance testing: process for testing that an implementation is compliant with a protocol standard, which is realized by test systems simulating the protocol with test scripts executed against the implementation under test

Interoperability testing: process for testing that devices can inter-operate, which is realized by connecting devices from different vendors and operating them, either manually or automatically, according to scenarios based on a protocol standard

Testing framework: document providing guidance and examples necessary for the development and implementation of a test specification

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in [i.7], [i.8], [i.13], [i.22] and the following apply:

RP	Reference Point
SDO	Standard Developing Organisation

4 ITS foundations

4.1 Motivation for and structure of the document

In order to carry out thorough protocol conformance and interoperability testing, complex test beds would be needed. In practice, it looks much more feasible to perform protocol conformance and interoperability tests in a simulated environment. A protocol conformance and interoperability testing framework for ITS thus is essential for a systematic and consistent approach towards testing of globally applicable ITS communications equipment.

The ITS testing framework presented in the present document is a set of guidelines for conformance testing and interoperability testing. It is based on the common architecture for communications in ITS from ETSI and ISO. Although it is referring explicitly to a consistent set of ITS base standards from ETSI and ISO, in no way it is restricted to this set of ITS base standards. Further on, it is enabling in a sense, that a manufacturer can implement and test those functionalities, considered to be important for market access.

The remaining part of clause 4 gives a short tutorial on communications in ITS. For more details existing base ITS standards from ETSI and ISO are listed.

Clause 5 provides an introduction to the ITS testing framework.

Clause 6 describes the approach for conformance testing.

Clause 7 describes the approach for interoperability testing.

Clause 8 provides more detailed guidance for the development of formal ITS test specifications and test suites.

Annex A provides information on the generic approach for interoperability testing.

4.2 ITS environment

The ITS architecture is presented in [i.13], [i.22] using different views. Main views presented are the top level networking view and the ITS station reference architecture view.

The top level networking view [i.13], [i.22] is presented in figure 1. A more implementation-oriented presentation is given in [i.17]. It is presumed that communication within a single network does not meet all the requirements of all ITS applications. Instead combinations of networks are envisioned, in which multiple ITS access and networking technologies are applied.

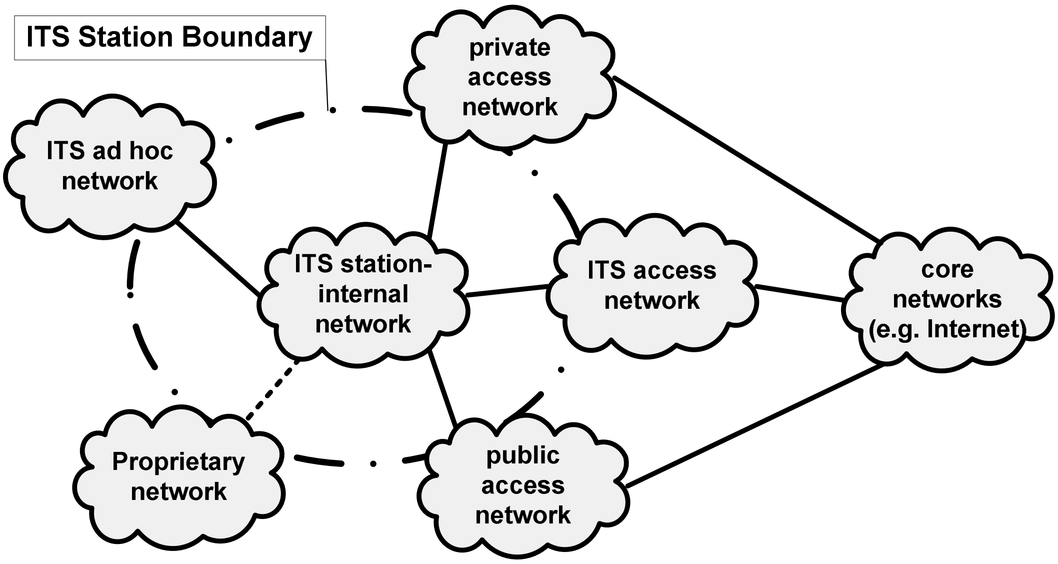

Figure 1: Top level networking view of ITS

The networks are grouped into

- ITS domain:
    - ITS station-internal network;
    - ITS station-external network:
        - ITS ad-hoc network;
        - ITS access network.
- Generic domain:
    - all other networks.

ITS are composed of four different ITS sub-systems, i.e. vehicular sub-system (cars, trucks, emergency vehicles, etc.), road-side sub-systems, central sub-systems and personal (portable) sub-systems as specified in [i.22], [i.13] presented in figure 2.

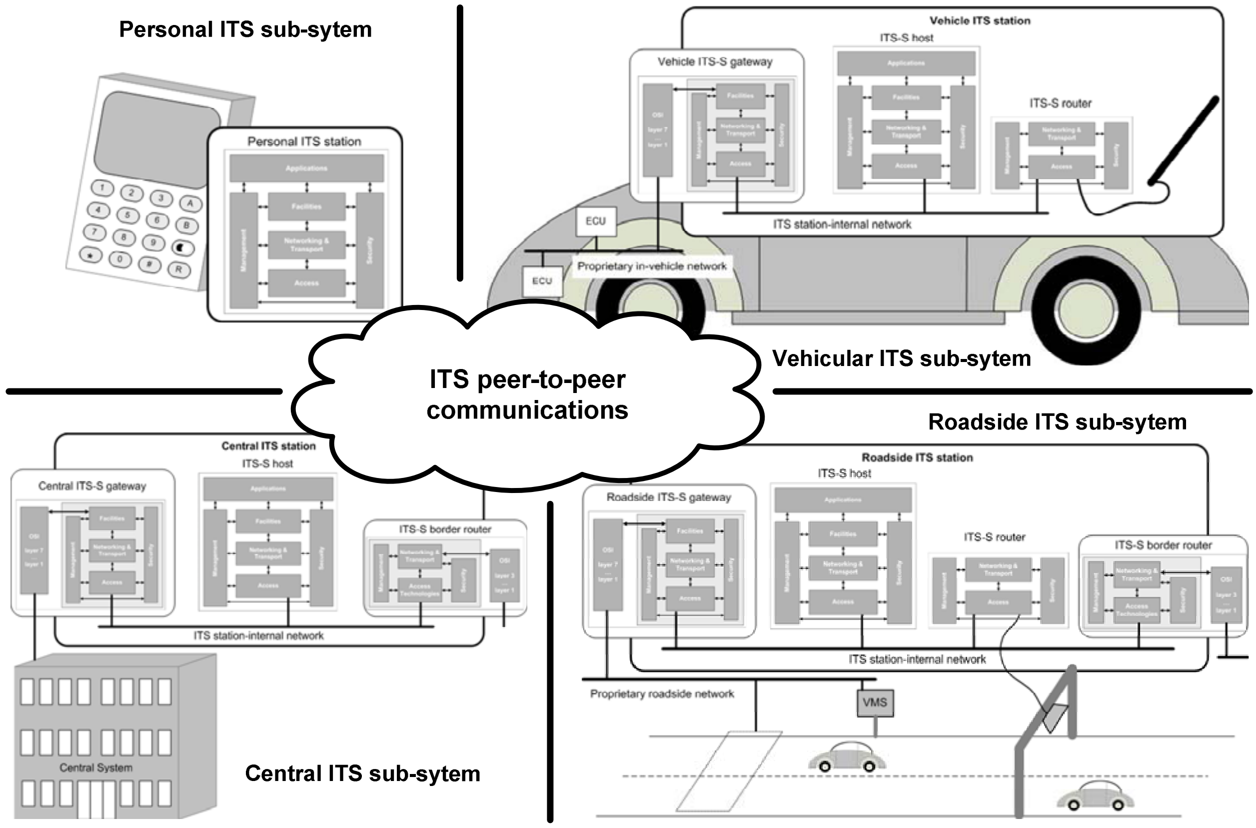

Figure 2: ITS sub-systems [i.13], [i.22]

Each ITS sub-system contains an ITS station with functionality as specified in the ITS station reference architecture presented in figure 3.

NOTE: The term "ITS station" (ITS-S) indicates a set of possible functionalities rather than a physical unit.

ITS stations act as peer stations in communication networks applying different access technologies, different networking & transport protocols, different facility protocols and the related security and management functionalities. The communication media usually are wireless. The various layers and entities in the ITS station reference architecture are interconnected via interfaces MI, MN, MF, MA, SI, SN, SF, SA, MS, IN, NF, FA [i.22], [i.13], see figure 3.

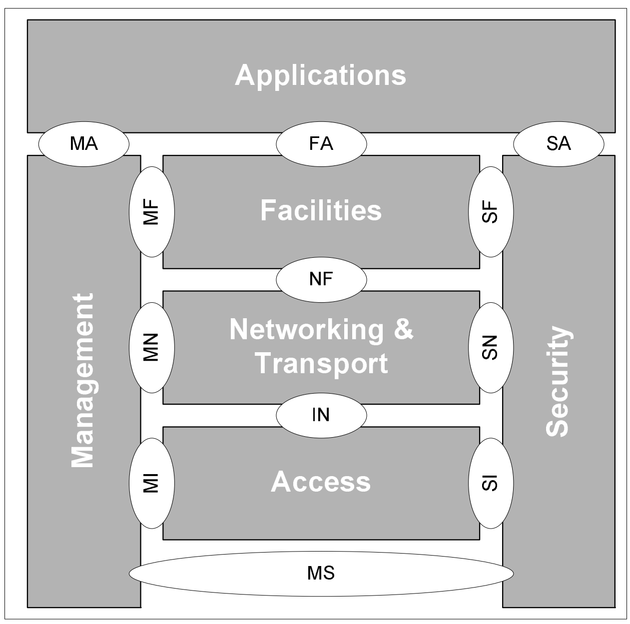

Figure 3: ITS station reference architecture [i.13], [i.22]

Accessibility of these interfaces may be dependent on the implementation.

Table 1 presents the layers and entities of the ITS station reference architecture being present in ITS-functional components.

Table 1: Summary of ITS layers and entities in ITS-S functional components

Main objects (ITS sub-systems)	ITS-S functional components	Layers and entities					
		Applications	Facilities (FL)	Networking & Transport (NL)	Access (AL)	Management (ME)	Security (SE)
Vehicle Roadside Central Personal	ITS-S host	X	X	X	X	X	X
	ITS-S router			X	X	X	X
	ITS-S gateway		X	X	X	X	X
	ITS-S border router			X	X	X	X

4.3 ITS base standards

Figure 4 shows a snap-shot of base standard development in several standardisation organisations as of April 2010.

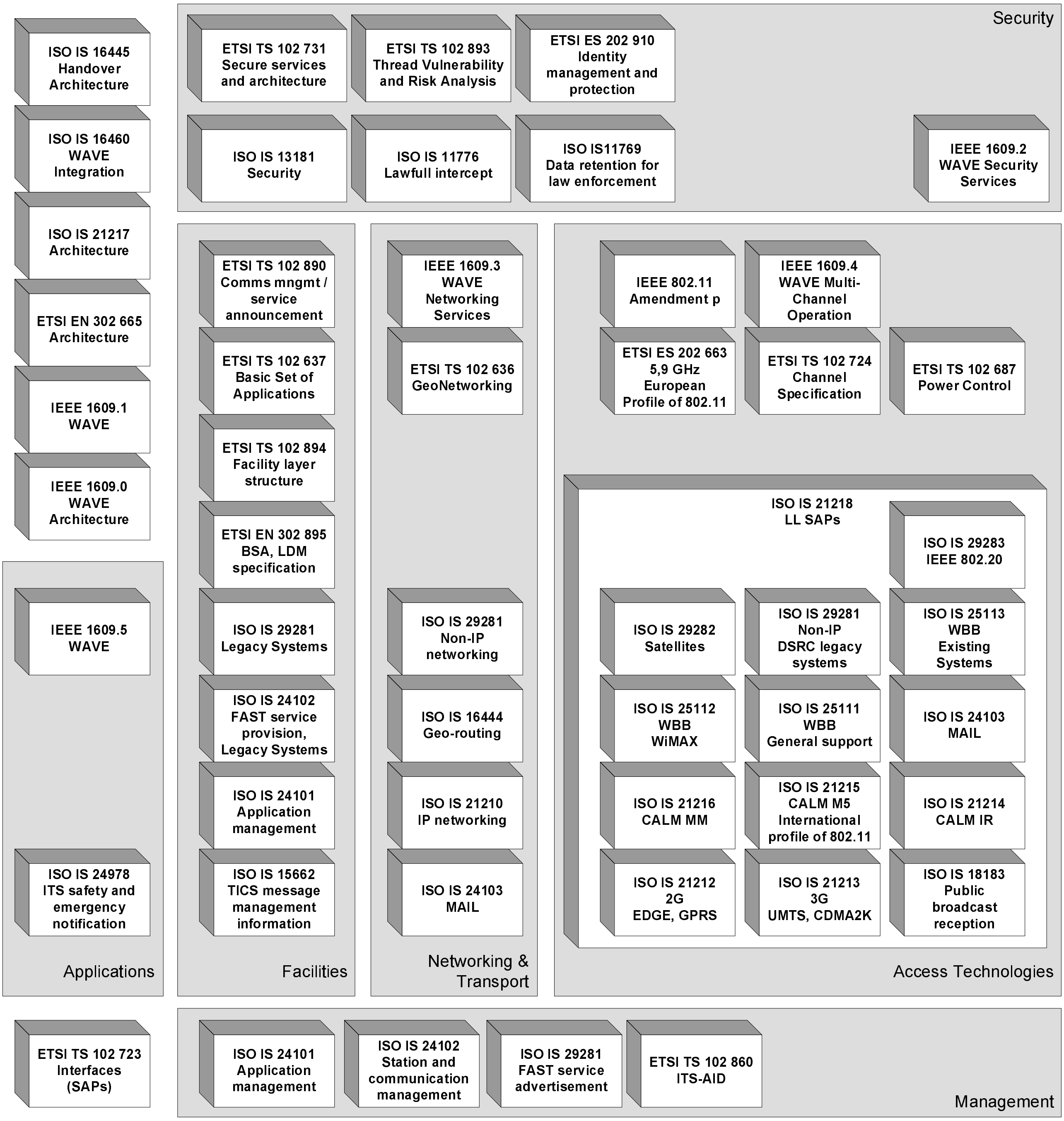

Figure 4: ITS base standard development

Standards displayed in figure 4, which may be subject for testing according to this ITS testing framework, are e.g.:

Access layer protocols:

- Channel specification [i.20] and power control [i.19] for ITS-G5;
- CALM media IR [i.11] and M5 [i.12];
- CALM medium service access points [i.14].

Networking & transport layer protocols:

- Basic transport protocol (for GeoNetworking) [i.17];
- GeoNetworking (media-independent functionality and media-dependent functionality ITS-G5) [i.17];
- IPv6 over GeoNetworking [i.17];
- CALM FAST networking and transport protocol [i.16];
- IPv6 in the context of ITS [i.9];
- CALM Geo-Routing (new work item at ISO TC204 WG16), referring to the basic GeoNetworking protocols developed at ETSI.

Facilities layer protocols:

- CAM [i.18];
- DENM [i.18];
- CALM legacy system support [i.15], [i.16] e.g. CEN DSRC for electronic fee collection.

Management, security and cross-layer protocols:

- CALM FAST service advertisement [i.15], [i.16];
- "Inter ITS-SCU" management communication [i.15];
- ITS station and communication management [i.15];
- ITS architecture elements [i.13].

4.4 ITS test standards

Examples of test standards for ITS base standards being under development/being finalized at ETSI and ISO are presented in figure 5.

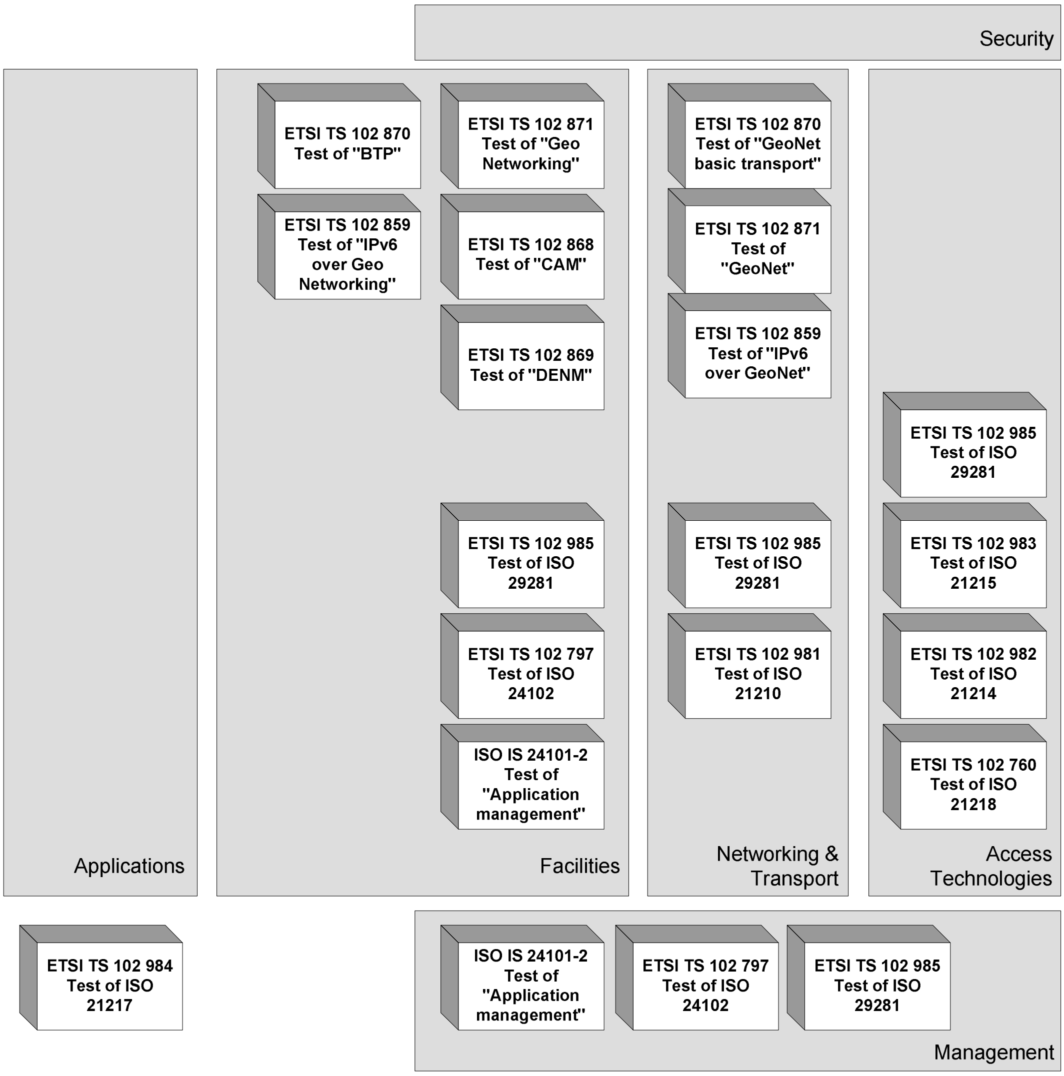

Figure 5: Activities on testing ITS base standards

Some test work items identified at ETSI at time of writing the present document are [i.23] to [i.34].

5 Introduction to the ITS testing framework EG

The present document provides:

- Guidance to identify the candidate implementations under test (IUT) for conformance testing in clause 6.1 and the equipment under test (EUTs) for interoperability in clause 7.1, i.e. answering the question "what is to be tested".
- Guidance to define the applicable test procedures in clauses 6 and 7, i.e. answering the question "how is to be tested".
- Guidance to develop the resulting test specifications and deliverables (for instance: TSS&TP, TP proforma, TTCN-3 test suite and documentation) in clause 8.

Figure 6 illustrates the ITS testing framework and the interactions with ITS base standards and ITS test specifications. The ITS testing framework is based on concepts defined in ISO 9646 [i.7], TTCN-3 [i.4] and EG 202 237 [i.5].

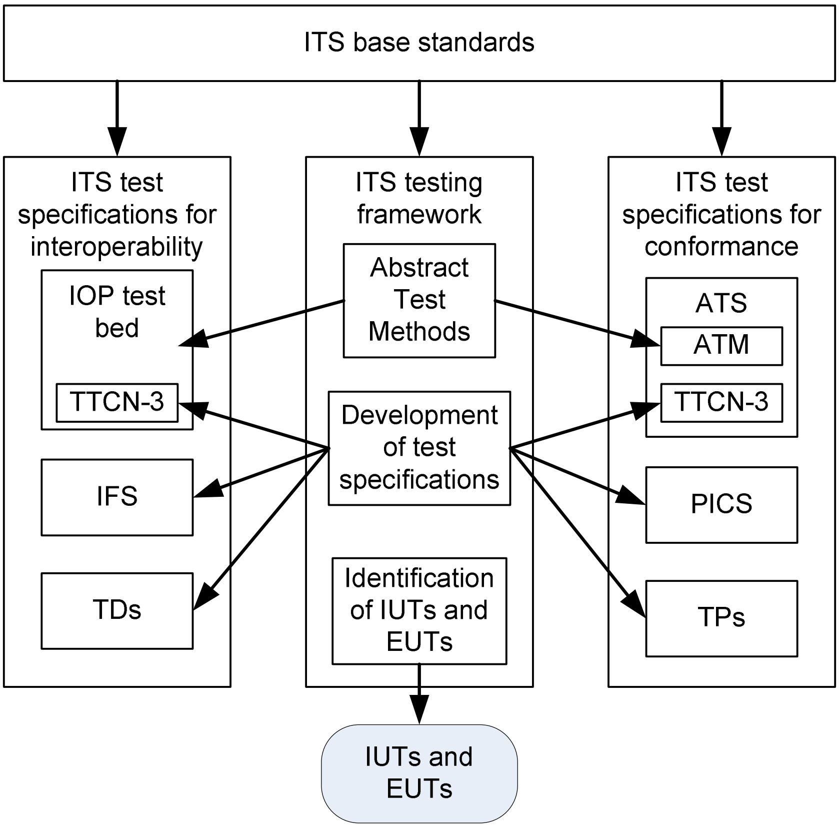

Figure 6: ITS testing framework interactions

As far as possible, specific ITS protocol base standards are identified which may be subject to conformance and/or interoperability testing according to this testing framework. However this ITS testing framework in no way restricts its applicability to these protocols. This ITS testing framework also in support of any other ITS protocol base standard.

5.1 Conformance testing

Clause 6 provides guidelines for conformance testing, which includes:

- Identification of candidate "Implementations Under Test" (IUT);
- Identification of reference points;
- Identification of the ATM:
    - abstract protocol tester;
    - functional TTCN-3 test architecture.

5.2 Interoperability testing

Clause 7 provides guidelines for interoperability testing, which includes:

- Identification of candidate EUTs.
- Identification of test scenarios.
- Definition of test bed architecture.

- Identification of test bed interfaces.

5.3 Steps for development of test specifications

Clause 8 provides guidance to write test specifications and to develop TTCN-3 test suites:

- Developing "Implementation Conformance Statements" (ICS) or "Interoperable Functions Statement" (IFS) from base standards, if not already provided as part of the base standard.
- Developing "Test Suite Structure and Test Purposes" (TSS&TP) from ICS and base standards.
- Developing "Test Descriptions" (TDs) from base standards.
- Developing ITS TTCN-3 test suite, e.g. naming conventions, code documentation, test case structure.

6 Conformance testing

The following clauses are providing conformance testing methodologies to be applied in ITS.

6.1 Candidate IUTs

The "Implementation Under Test" (IUT) is a protocol implementation considered as an object for testing. This means that the test process will focus on verifying the compliance of this protocol implementation (IUT) with requirements set up in the related base standard. An IUT normally is implemented in a "System Under Test" (SUT). For testing, an SUT is connected to a test system over at least a single interface specified in the related ITS base standard. Such an interface is identified as "Reference Point" (RP) in the present document.

NOTE: Other interfaces between the test system and the IUT may be used to control the behaviour of the IUT during the test process.

Further details on RPs are presented in clause 6.2.

IUTs normally are protocols located in a layer or an entity of the ITS station reference architecture presented in figure 3. Examples of IUTs and their location in the ITS station reference architecture are illustrated in figure 7. Further details are presented in tables 2 and 3.

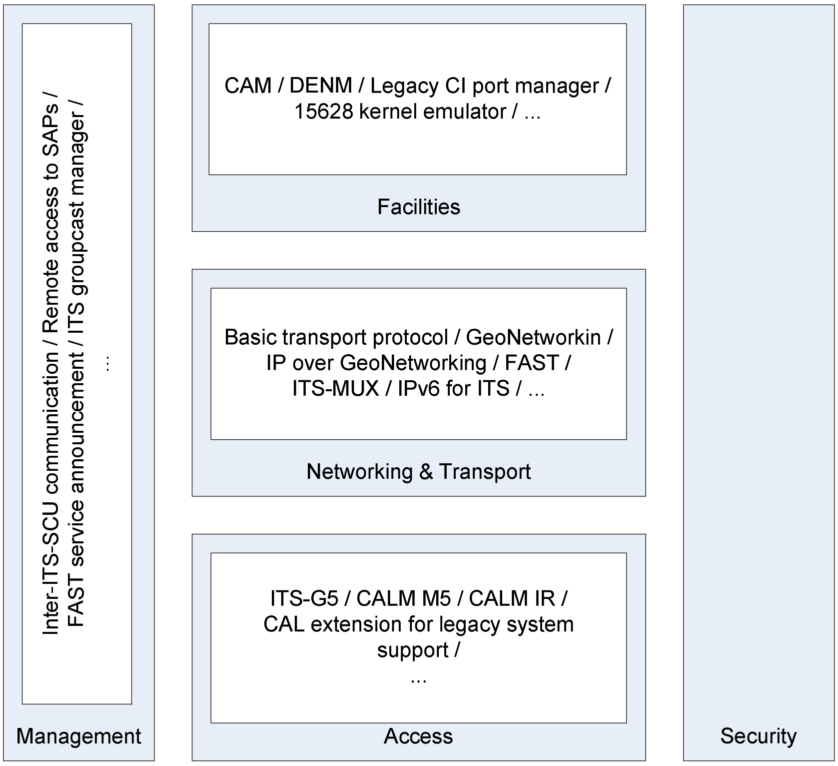

Figure 7: Examples of IUTs in the ITS station reference architecture

Table 2 exemplifies candidates for IUTs for RP-External identified in clause 6.2.

Table 2: Candidates for IUTs for RP-External

Layer/entity	IUT (protocols)	ITS specific	Notes
Apps	ITS-S application messages	YES	Not considered for conformance testing. To be used for interoperability testing.
FL	CAM [i.18]	YES	
	DENM [i.18]	YES	
	Legacy CI port manager [i.16]	YES	
	15628 kernel emulator [i.16]	YES	
NL	Basic transport protocol for GeoNetworking [i.17]	YES	
	FAST networking and transport protocol [i.16]	YES	
	GeoNetworking [i.17]	YES	
	IPv6 for ITS [i.9]	YES	No test of IPv6 functionality from IETF, but only tests on how to use this functionality in the context of ITS.
	IPv6 over GeoNetworking [i.17]	YES	
	ITS-MUX [i.16]	YES	
	Legacy protocols such as TCP, UDP	NO	
AL	CAL extension for legacy systems [i.16]	YES	
	CALM IR [i.11]	YES	
	CALM M5 (ITS-G5) [i.12]	YES	The basic MAC and PHY is standardized in IEEE 802.11 [i.38] and thus not subject of tests within this ITS framework.
	CALM medium service access points/support of legacy technologies [i.14]	YES	
	Legacy technologies	NO	Covered by other testing frameworks.
ME	FAST service advertisement [i.15]	YES	
	Legacy system support [i.15]	YES	
SE	To be defined	YES	Base standards are under development.

Table 3 exemplifies candidates for IUTs for the RP-Internal-ITS-S identified in clause 6.2.

Table 3: Candidates for IUTs for RP-Internal-ITS-S

Layer/entity	Protocols	ITS specific	Notes
ME	Inter-ITS-SCU PDUs (allows for e.g. remote access to SAPs) [i.15]	YES	
	LAN technology protocols, wired or wireless, e.g. TCP/IP, Ethernet, Bluetooth	NO	Does not require specific testing for ITS.
NL	Local forwarding of FAST packets [i.16]	Yes	

Tables 2 and 3 need to be amended in the following cases:

- A new protocol is identified as a new potential IUT and interacts in one of the RPs.
- A new ITS-S functional component or ITS sub-system is defined. This should be included as a SUT as long as it interacts in one of the RPs.
- If a new RP is identified, the same analysis is done for incorporating the corresponding SUT and IUT.

6.2 Reference points

This clause illustrates candidate reference points (RPs) where test systems can be connected in order to test conformance of ITS protocols (IUTs) with ITS base standards.

RPs listed in tables 4 and 5 are classified in two groups:

- **External networking RPs** (RP-External) are based on the ITS sub-systems presented in figure 2 and the related external networks presented in figure 1.

Table 4: External networking reference points RP-External

RP Identifier	RP Name	ITS sub-system	ITS sub-system	Network
RP-ITS-Ext-1	R_A	Vehicle	Vehicle	ITS ad-hoc
RP-ITS-Ext-2	R_B	Vehicle	Roadside	ITS ad-hoc
RP-ITS-Ext-3	R_C	Personal	Personal	ITS ad-hoc
RP-ITS-Ext-4	R_D	Roadside	Personal	ITS ad-hoc
RP-ITS-Ext-5	R_E	Vehicle	Personal	ITS ad-hoc
RP-ITS-Ext-6	R_F	Roadside	Roadside	ITS ad-hoc
RP-ITS-Ext-7	R_I	Vehicle	Roadside	ITS ad-hoc
RP-ITS-Ext-8	R_J	Roadside	Personal	ITS ad-hoc
RP-ITS-Ext-9	R_K	Vehicle	Roadside	Public
RP-ITS-Ext-10	R_L	Vehicle	Central	Public
RP-ITS-Ext-11	R_M	Roadside	Personal	Public
RP-ITS-Ext-12	R_N	Personal	Central	Public
RP-ITS-Ext-13	R_O	Vehicle	Personal	Public
RP-ITS-Ext-14	R_P	Roadside	Central	Public
RP-ITS-Ext-15	R_S	Roadside	Central	Private
RP-ITS-Ext-16	R_Z	Central	Central	Public/Private

- **Internal network RPs** (RP-Internal-ITS-S) are based on the ITS-S functional components listed in clause 4.2 and taking into account the interaction via the ITS station internal network presented in figures 1 and 2 between the functional components inside the same sub-system.

Table 5: Internal networking reference points RP-Internal-ITS-S

ITS sub-system	RP Name	ITS-S functional component	ITS-S functional component	ID
Vehicle	R_{V1}	ITS-S host	ITS-S host	**RP-ITS-Int-1**
	R_{V2}	ITS-S host	ITS-S router	**RP-ITS-Int-2**
	R_{V3}	ITS-S host	ITS-S gateway	**RP-ITS-Int-3**
	R_{V4}	ITS-S host	ITS-S border router	**RP-ITS-Int-4**
	R_{V5}	ITS-S router	ITS-S router	**RP-ITS-Int-5**
	R_{V6}	ITS-S router	ITS-S gateway	**RP-ITS-Int-6**
	R_{V7}	ITS-S router	ITS-S border router	**RP-ITS-Int-7**
	R_{V8}	ITS-S gateway	ITS-S gateway	**RP-ITS-Int-8**
	R_{V9}	ITS-S gateway	ITS-S border router	**RP-ITS-Int-9**
Roadside	R_{R1}	ITS-S host	ITS-S host	**RP-ITS-Int-10**
	R_{R2}	ITS-S host	ITS-S router	**RP-ITS-Int-11**
	R_{R3}	ITS-S host	ITS-S gateway	**RP-ITS-Int-12**
	R_{R4}	ITS-S host	ITS-S border router	**RP-ITS-Int-13**
	R_{R5}	ITS-S router	ITS-S router	**RP-ITS-Int-14**
	R_{R6}	ITS-S router	ITS-S gateway	**RP-ITS-Int-15**
	R_{R7}	ITS-S router	ITS-S border router	**RP-ITS-Int-16**
	R_{R8}	ITS-S gateway	ITS-S gateway	**RP-ITS-Int-17**
	R_{R9}	ITS-S gateway	ITS-S border router	**RP-ITS-Int-18**
Central	R_{C1}	ITS-S host	ITS-S host	**RP-ITS-Int-19**
	R_{C2}	ITS-S host	ITS-S router	**RP-ITS-Int-20**
	R_{C3}	ITS-S host	ITS-S gateway	**RP-ITS-Int-21**
	R_{C4}	ITS-S host	ITS-S border router	**RP-ITS-Int-22**
	R_{C5}	ITS-S router	ITS-S router	**RP-ITS-Int-23**
	R_{C6}	ITS-S router	ITS-S gateway	**RP-ITS-Int-24**
	R_{C7}	ITS-S router	ITS-S border router	**RP-ITS-Int-25**
	R_{C8}	ITS-S gateway	ITS-S gateway	**RP-ITS-Int-26**
	R_{C9}	ITS-S gateway	ITS-S border router	**RP-ITS-Int-27**

NOTE: Table 5 shows all possible RPs for the vehicle sub-system, roadside sub-system and the central sub-system. The presentation applies also in principle to the personal sub-system - although not shown in the table, as the distinction is only an illustrative distinction according to the context of usage of an ITS station.

Access to the proprietary networks (proprietary in-vehicle network, proprietary roadside network and proprietary central network shown in figure 2) always goes via a gateway and thus is not considered in table 5.

Important candidates of the external networking RPs are those related to communications with a vehicular ITS sub-system.

All internal networking RPs are potential candidates.

6.3 Identification of abstract test method

6.3.1 Abstract protocol tester

An abstract protocol tester presented in figure 8 is a process providing the test behaviour for testing an IUT. Thus it will emulate a peer IUT of the same layer/the same entity. This type of test architecture provides a situation of communication which is equivalent to real operation between real ITS devices. The ITS test system will simulate valid and invalid protocol behaviour, and will analyse the reaction of the IUT. Then the test verdict, e.g. pass or fail, will depend on the result of this analysis. Thus this type of test architecture enables to focus the test objective on the IUT behaviour only.

In order to access an IUT, the corresponding abstract protocol tester needs to use lower layers to establish a proper connection to the system under test (SUT) over a physical link (Lower layers link).

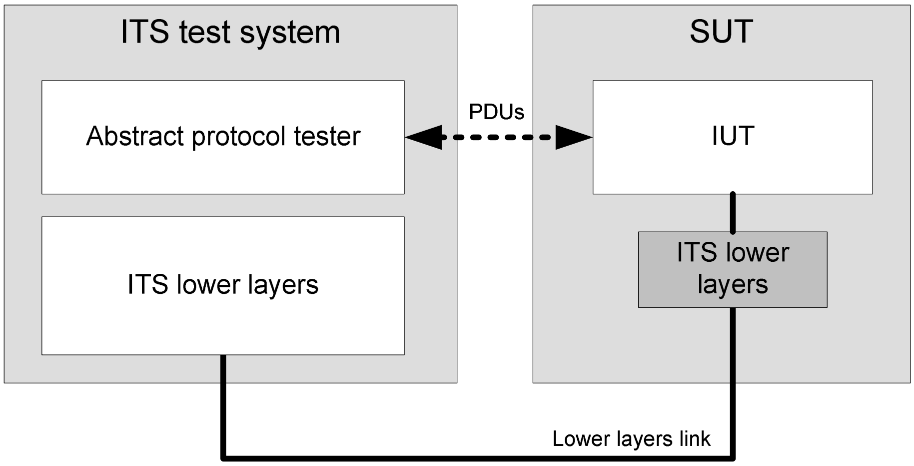

Figure 8: Generic abstract protocol tester

The "Protocol Data Units" (PDUs) are the messages exchanged between the IUT and the abstract protocol tester as specified in the base standard of the IUT. These PDUs are used to trigger the IUT and to analyse the reaction from the IUT on a trigger. Comparison of the result of the analysis with the requirements specified in the base standard allows to assign the test verdict.

Further control actions on the IUT may be necessary from inside the SUT, for instance to simulate a primitive from the upper layer or the management/security entity. Further details on such control actions are provided by means of an upper tester presented in clause 6.3.2.

The above "Abstract Test Method" (ATM) is well defined in ISO 9646-1 [i.7] and supports a wide range of approaches for testing including the TTCN-3 abstract test language [i.4].

For instance, to test the CAM facility, the abstract protocol tester will emulate the CAM functionality. In order to test the CAM facility, the CAM abstract protocol tester will use e.g. the BTP and GeoNetworking protocol in the networking & transport layer and the ITS-G5 access technology in the access layer.

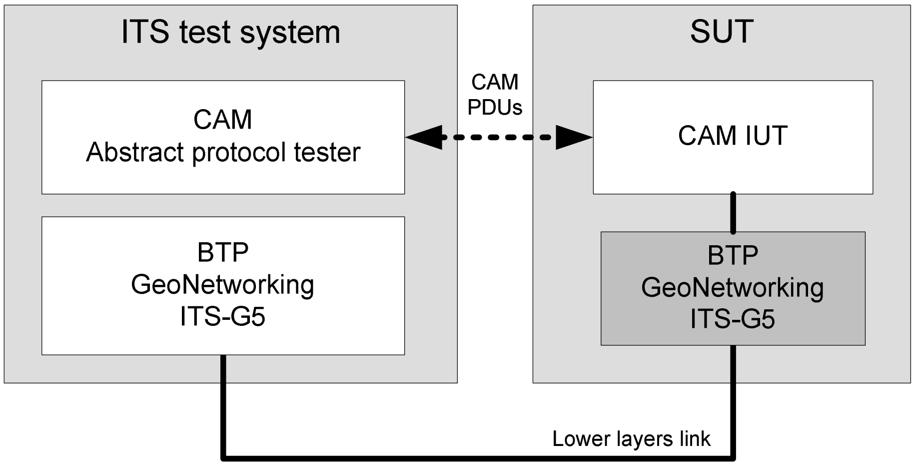

Figure 9: Abstract protocol tester for CAM

A current snap-shot of protocols to be tested (IUT) is shown in table 6. This table indicates which lower layer protocols (may) belong to which IUT in order to build the proper ITS test system.

Table 6: Mapping between protocols (IUTs) and lower layer protocols for RP-External and RP-Internal-ITS-S

Protocol to be tested (IUT)	Protocols of lower layers	IUT base standards
CAM	BTP, GeoNetworking, ITS-G5	TS 102 637-2 [i.18]
DENM	BTP, GeoNetworking, ITS-G5	TS 102 637-3 [i.18]
Basic transport protocol	GeoNetworking, ITS-G5	TS 102 636-5-1 [i.17]
GeoNetworking	ITS-G5	TS 102 636-4-1 and TS 102 636-4-2 [i.17]
IPv6 over GeoNetworking	ITS-G5	TS 102 636-6-1 [i.17]
DLL for IR access technology	IR PHY	ISO 21214 [i.11]
DLL for M5 access technology	M5 PHY	ISO 21215 [i.12]
CALM FAST networking & transport protocol	Any ad-hoc access technology	ISO 29281 [i.16]
FAST service announcement	CALM FAST networking & transport protocol [i.16], Any ad-hoc access technology	Facility located in the management entity, specified in ISO 24102 [i.15]
15628 kernel emulator	CALM FAST networking & transport protocol [i.16] Any ad-hoc access technology	Facility specified in ISO 29281 [i.16]
15628 legacy CI port manager	CALM FAST networking & transport protocol [i.16] 15628 Legacy CI [i.16] with extended "Communication Adaptation Layer" (CAL).	Facility specified in ISO 29281 [i.16]
ITS groupcast manager	CALM FAST networking & transport protocol [i.16] Any ad-hoc access technology	ISO 24102 [i.15] and ISO 29281 [i.16]
ITS-MUX	CALM FAST networking & transport protocol [i.16], GeoRouting Any ad-hoc access technology	ISO 29281 [i.16]
Inter-ITS-SCU-communication	Any networking protocol, any LAN technology	ISO 24102 [i.15]
Remote access to SAPs	ITS-SCU-communication [i.15] Any networking protocol, any LAN technology	ISO 24102 [i.15]
CALM FAST networking & transport protocol - ITS-S local forwarding	Any LAN technology	ISO 29281 [i.16]

The framework enables to define abstract protocol testers for other IUTs than CAM following the generic approach presented in figure 8. For instance to test the basic transport protocol (being the IUT), the protocols of the lower layers are GeoNetworking and ITS-G5.

New protocols (IUTs) may be added to table 6.

6.3.2 Functional TTCN-3 test architecture

This clause illustrates how to implement the abstract test architecture presented in clause 6.3.1 in a functional test environment. There are many possibilities to implement this abstract test architecture using different types of programming languages and test devices. This ITS testing framework uses TTCN-3 being a standardized testing methodology including a standardized testing language [i.4], which is fully compliant with the ISO 9646 abstract test methodology [i.7].

Figure 10 presents the functional test architecture for ITS conformance testing, applying the TTCN-3 test system specified in ES 201 873-5 [i.4].

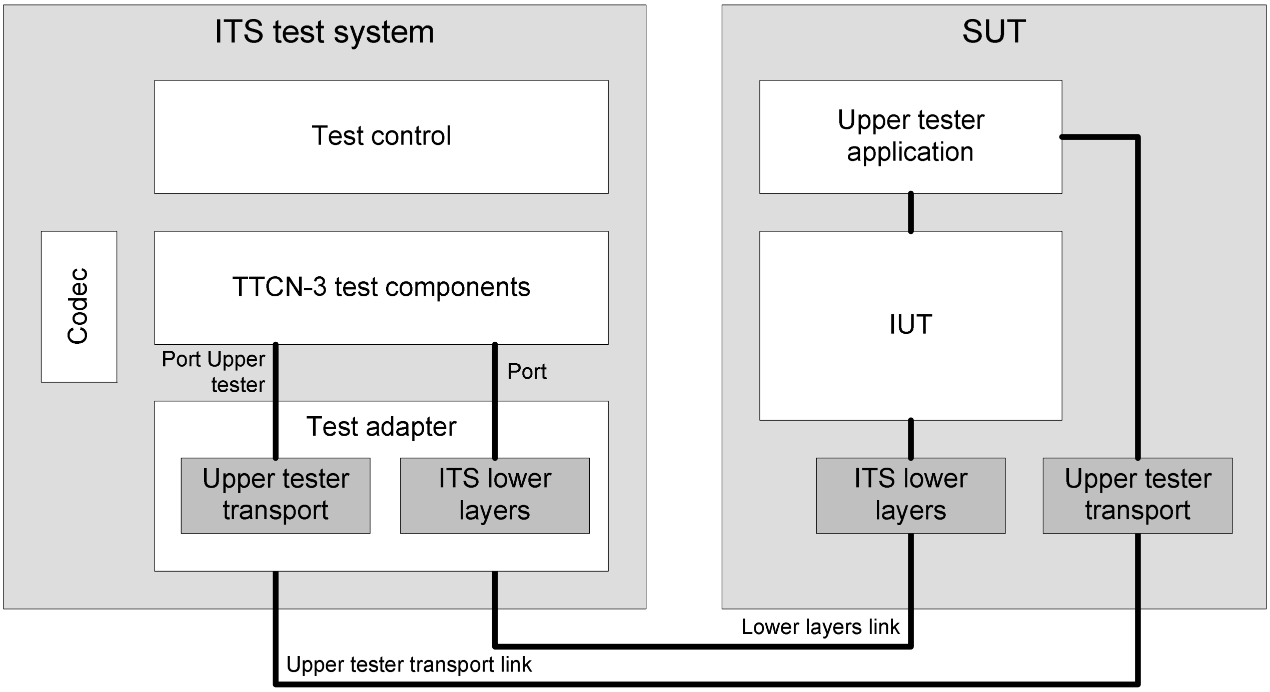

Figure 10: Conformance test system architecture

The "System Under Test" (SUT) contains:

- The "Implementation Under Test" (IUT), i.e. the object of the test.
- The "Upper tester application" enables to simulate sending or receiving service primitives from protocol layers above the IUT or from the management/security entity.
- The "ITS lower layers" enable to establish a proper connection to the system under test (SUT) over a physical link (Lower layers link). The lower layers link is located at a "Reference Point" (RP), see clause 6.2.
- The "Upper tester transport" is a functionality, which enables the test system to communicate with the upper tester application. Then the upper tester can be controlled by a TTCN-3 test component as part of the test process.

The "ITS test system" contains:

- The "TTCN-3 test components" are processes providing the test behaviour. The test behaviour may be provided as one single process or may require several independent processes.
- The "Codec" is a functional part of the test system to encode and decode messages between the TTCN-3 internal data representation and the format required by the related base standard.
- The "Test Control" enables the management of the TTCN-3 test execution (parameter input, logs, test selection, etc.).

- The "Test adapter" (TA) realizes the interface between the TTCN-3 ports using TTCN-3 messages, and the physical interfaces provided by the IUT.

In case Inter-ITS-SCU-communications as specified in [i.15] is implemented, which allows remote access to SAPs in an ITS station, the upper tester functionality may be implemented using the ITS management entity. In this case, the "upper tester transport link" presented in figure 10 will be the local area network used for Inter-ITS-SCU-communications.

7 Interoperability testing

Annex A provides a generic introduction to interoperability testing. The following clauses provide interoperability testing methodologies to be applied in ITS.

7.1 Candidate EUTs

For interoperability testing, only "Equipment Under Test" (EUT) is considered. An EUT is a physical implementation of an ITS station or a physical implementation of a functional sub-set of an ITS station, which interacts with one or several other EUTs via one or more RPs.

7.2 Test scenarios

In ITS, a large number of use cases [i.18] is already identified. In a specific implementation of an ITS station, very likely only a sub-set of these use cases is supported. In order to perform interoperability tests, EUTs supporting the same use cases are required. This classification of interoperability tests is given by test scenarios. A test scenario thus selects a set of use cases and is restricted to a sub-set of the full functionality of such a set.

In other words, EUTs considered for defining the test scenarios are implementations of ITS stations with various roles, e.g. vehicle ITS stations, roadside ITS stations, personal ITS stations, central ITS stations, but sharing a common functionality.

A typical example is a test scenario for vehicle ITS stations and optional roadside ITS stations, where these ITS stations periodically transmit cooperative awareness messages (CAM) [i.26].

7.3 Test bed architecture

A test architecture is an abstract description of logical entities as well as their interfaces and communication links involved in a test.

After analysing the requirements of ITS applications and taking into consideration the test scenarios and the external RPs, the present document suggests that the SUT should be composed of at least three vehicles and two roadside units, being the EUTs, in order to cover completely the test scenarios and the use cases. This is shown in figure 11.

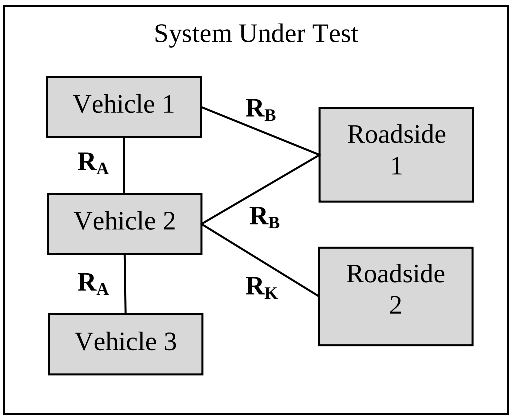

Figure 11: Interoperability ITS SUT

The approach followed to design the test bed architecture is listed below:

- All necessary test scenarios and use cases are covered.
- The design is independent of:
  - How the tests will be executed. The architecture could be implemented to be executed either in a manual way, or in an automated way or using a manual-automated combination.
  - Whether the vehicles (EUTs) are moving or are not. The architecture could be use either in a laboratory environment or in a field trial environment.
- The design provides interfaces for:
  - configuring the test bed and EUTs;
  - stimulating the EUTs;
  - monitoring the behaviour of the EUTs;
  - tracing the execution of the tests.
- The design is able to support live and offline interoperability test execution.

These interfaces may imply the usage of automated interoperability testing and interoperability testing with conformance check as described in annex A.

The ITS functional test bed architecture is illustrated in figure 12.

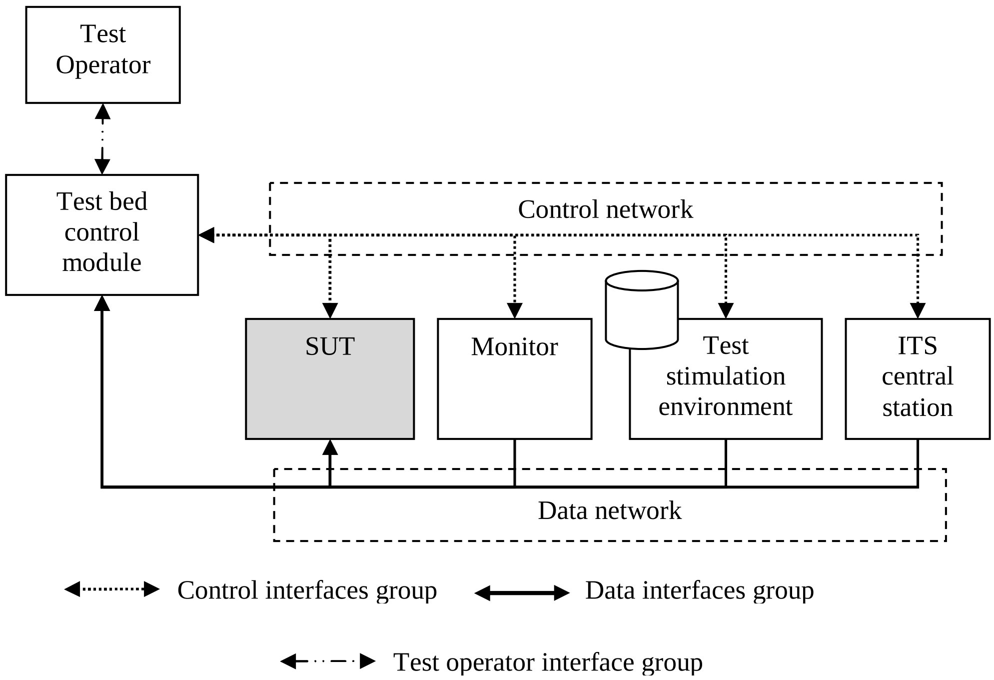

Figure 12: Functional test bed architecture

This test bed is mainly composed of several functional entities:

- **SUT:** It is composed of a set of EUTs (vehicles and roadsides). It is supposed that the EUTs are equipped with all the devices (sensors, etc.) needed to perform the tests.
- **Test bed control module:** This entity manages the whole test bed. It is considered to be the core of the test bed. This module synchronizes, configures, controls and runs the other entities and even the SUT. In addition, this entity gathers all the information generated by each entity in term of traces with the aim of having a global overview of the execution of the tests. Depending of the implementation of the test bed, this module might also assign the test verdicts.
- **Test stimulation environment:** This entity is in charge of stimulating the SUT for a specific test conditions, e.g. geographical area coverage, GPS information, road hazard elements, traffic simulation. This entity may be implemented e.g. by a database storing the information or even by using real devices such as traffic lights.
- **ITS central station:** It provides centralized ITS applications for some use cases.
- **Monitor:** This entity checks and gathers messages on relevant communication links.
- **Networks:** the test bed identifies two types of network depending on the type of information which is going to be carried out. One of the networks is used for carrying out data, and the other one is used for control. In addition, each network identifies a set of interfaces which are described in clause 7.4.

NOTE 1: The definition of the test bed architecture should be done simultaneously with the test description specification.

NOTE 2: For the implementation of an automated test bed, the present document suggests to follow the guidelines indicated in [i.6].

7.4 Test bed interfaces

The test bed classifies the interfaces in three groups:

- **Data:** this group contains the interfaces where data is exchanged. Depending on the type of data being exchanged, the interfaces are classified into three categories:
    - **Stimulating:** this interface carries information generated by the test bed in order to stimulate the EUTs for a specific behaviour.
    - **Monitoring:** this interface carries the protocol message exchanged between the EUTs during the execution of the tests.
    - **Tracing:** this interface carries information about the status of the execution of the EUTs and the test bed entities in order to be able to analyze as much as possible the execution of a test.
- **Control:** this group is used to configure and control the various entities in the test bed, and even the EUTs, by passing necessary parameters.
- **Test Operator:** this group provides the capability of controlling the test bed control module. Through this interface, a test operator would be able to select the test to be executed, to configure the different entities involved in the tests and to analyse the results obtained during the test execution.

Figure 13 illustrates interfaces involved in the test bed.

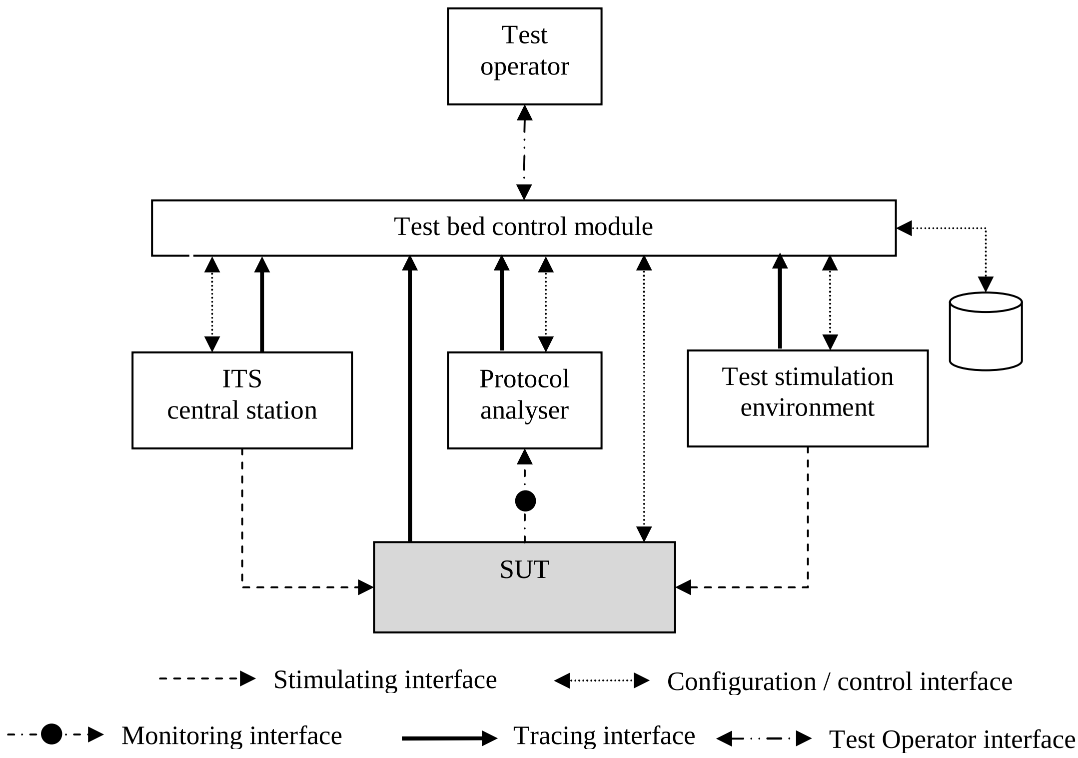

Figure 13: Interfaces of a test bed

8 Development of test specifications and test suites

This part of the present document uses terminology and concepts that may need minimum knowledge of the corresponding methodologies [i.4] and [i.7].

Applying the methodology presented in clauses 6 and 7 will result in test specifications. This clause provides guidance to draft test specifications for conformance testing and for interoperability testing. The following test deliverables usually are provided:

- For conformance testing:
  - ICS proforma;
  - TSS&TPs;
  - TTCN-3 abstract test suites including the corresponding documentation.
- For interoperability testing:
  - IFS proforma;
  - test descriptions.

8.1 Provision of ICS pro-forma

The guidance to produce ICS proforma is provided in EG 201 058 [i.3] and no extra guidance is required for the context of ITS.

8.2 Provision of TSS&TPs

A test purpose is a prose description of a well-defined objective of testing. Applying to conformance testing, it focuses on a single conformance requirement or a set of related conformance requirements from the base standards.

Several types of presentation of the test purposes exist. These presentations are combining text with graphical presentations, mainly tables, and include sometimes message sequence charts. The present document presents a proposed table template to write test purposes with recommendations concerning the wording and the organization of the test purposes.

There are usually numerous test purposes, which need to be organized in structured groups. The organization of the test purposes in groups is named "Test Suite Structure".

The development of the test purposes follows the analysis of the conformance requirements, clearly expressed in the base standards. Furthermore, the analysis of a base standard leads to the identification of different groups of functionalities, which are used to define the first levels of the test suite structure.

The guidance provided in the following clauses is based on two ETSI reference documents produced by the MTS technical committee [i.1] and [i.2].

8.2.1 Test suite structure (TSS)

Defining the test suite structure consists of grouping the test purposes according to different criteria like for instance:

- The functional groups and sub-groups of procedures in the base standard, from which the requirement of the test purpose is derived.
- The category of test applying to the test purposes, for instance:
  - valid behaviour test;
  - invalid behaviour test;

- timer test;
- etc.

Usually the identification of the different functional groups of procedures leads to the definition of the top levels of the TSS. Then further levels at the bottom of the TSS is used to group test purposes belonging to the same type of test.

Table 7 shows an example of a two level TSS used in the TSS&TP for the CAM facility.

Table 7: Example of test suite structure for CAM facility

Root	Group	Sub-Group	Category
CAM	Message generation		Valid behaviour
	ITS profile checking		Valid behaviour
	Information adaptation	Crash Status	Valid behaviour
		Dangerous goods	Valid behaviour
		Confidence station length/width	Valid behaviour
		Door open	Valid behaviour
		Distance To Stop Line	Valid behaviour
		Turn Advice	Valid behaviour
		Curvature Change	Valid behaviour
		Occupancy	Valid behaviour
		Light Bar In Use	Valid behaviour
		Siren In Use	Valid behaviour
		Traffic Light Priority	Valid behaviour
		Schedule Deviation	Valid behaviour
		PT Line Description	Valid behaviour
		Exterior Lights	Valid behaviour
	Position adaptation		Valid behaviour
	Message processing		Valid behaviour

8.2.2 TP pro-forma

A test purpose is an informal description of the expected test behaviour. As such it is written in prose.

In order to increase the readability of the TP, the following two recommendations should be followed:

- Each TP should be presented in a table, containing two main parts:
  - The TP header, which contains the TP identifier, the TP objective and the external references (PICS, and base standard).
  - The behaviour part, which contains the test behaviour description. This part can be optionally divided in the three following parts, in order to increase the readability:
    - the initial conditions;
    - the expected behaviour;
    - the final conditions.
- The prose describing the test behaviour (including initial and final conditions) should follow some rules, as for instance the use of reserved keywords and syntax.

Table 8: TP pro-forma template

TP Id	
Test objective	
Reference	
PICS Selection	
Initial conditions *(optional)*	
Expected behaviour	
Final conditions *(optional)*	

Table 9: Description of the fields of the TP pro-forma

TP Header	
TP ID	The TP ID is a unique identifier. It shall be specified according to the TP naming conventions defined in the above clause.
Test objective	Short description of test purpose objective according to the requirements from the base standard.
Reference	The reference indicates the clauses of the reference standard specifications in which the conformance requirement is expressed.
PICS selection	Reference to the PICS statement involved for selection of the TP. Contains a Boolean expression.
TP Behaviour	
Initial conditions	The initial conditions defines in which initial state the IUT has to be to apply the actual TP. In the corresponding Test Case, when the execution of the initial condition does not succeed, it leads to the assignment of an Inconclusive verdict.
Expected behaviour (TP body)	Definition of the events, which are parts of the TP objective, and the IUT are expected to perform in order to conform to the base specification. In the corresponding Test Case, Pass or Fail verdicts can be assigned there.
Final conditions	Definition of the events that the IUT is expected to perform or shall not perform, according to the base standard and following the correct execution of the actions in the expected behaviour above. In the corresponding Test Case, the execution of the final conditions is evaluated for the assignment of the final verdict.

Defining the initial and final conditions, separately from the expected behaviour, makes the reading of the TP easier and avoid misinterpretations.

The "expected behaviour", which matches the events corresponding to the TP objective, can also be named "TP body", which is similar to the "test case body" in an abstract test suite (ATS).

8.2.3 TP identifier

The TP identifier identifies uniquely the test purposes. In order to ensure the uniqueness of the TP identifier, it follows a naming convention.

The more useful and straightforward naming convention consists of using the test suite structure, to form the first part of the TP identifier. Then the final part consists of a number to identify the TP order within a TP group.

Table 10 shows an example of TP naming convention applying to the TSS described in table 9.

The TP identifier is formed by the abbreviation "TP", followed by abbreviation representing the group of the following TSS levels, ending with a number representing the TP order. Each field of the TP identifier is separated by a "/".

Table 10: Example of TP naming convention for CAM facilities

TP/<root>/<gr>/<sgr>/<x>/<nn> or TP/<root>/<gr>/<x>/<nn>, when no <sgr>		
<root> = root	CAM	
<gr> = group	MSG	Message Generation
	IPC	ITS profile checking
	INA	Information adaptation
	POA	Position adaptation
	MSP	Message Processing
<sgr> = sub- group	CRS	Crash Status
	DAG	Dangerous goods
	CLW	Confidence station length/width
	DOP	Door open
	DSL	Distance To Stop Line
	TAD	Turn Advice
	CUC	Curvature Change
	OCC	Occupancy
	LBU	Light Bar In Use
	SIU	Siren In Use
	TLP	Traffic Light Priority
	SCE	Schedule Deviation
	PLD	PT Line Description
	EXL	Exterior Lights
<x> = type of testing	BV	Valid Behaviour tests
	BI	Invalid Syntax or Behaviour Tests
<nn> = sequential number		01 to 99

A TP identifier, following the TP naming convention of the table could be TP/CAM/INA/DAG/BV/01.

The TP numbering uses two digits for presentation, and starts with 01 rather than with 00. Exceeding 99 TPs per group is not recommended. In such a case, it is rather recommended to create sub-groups, in order to keep clarity in the Test Suite Structure.

8.2.4 Test objective

The test objective clearly indicates which protocol requirement is intended to be tested in the test purpose. This part eases the understanding of the TP behaviour. This also eases the identification of the requirements, which were used as a basis for the test purpose.

It is recommended to limit the length of the test objective to one sentence.

See also the example in table 13.

8.2.5 Reference

In the reference row, the TP writer indicates, in which clauses of the protocol standards, the requirement are expressed. This information is critical, because it justifies the existence and the behaviour of the TP.

The reference row may refer to several clauses. When the clause containing the requirement is big (for instance, more than ½ page), it is recommended to indicate the paragraph of the clause where the requirement was identified.

The reference to the base standard actually is precise enough to enable the TP reader to identify quickly and precisely the requirement.

See also the example in table 13.

8.2.6 PICS selection

The PICS selection row contains a Boolean expression, made of PICS parameters. It is recommended to use PICS acronym, which clearly identify the role of the PICS, instead of using the PICS table and row references.

See also the example in table 15.

A mapping table is included in the TP document to link the PICS acronym with its corresponding reference in the PICS document. See the example table 11.

Table 11: Example of pre-defined keywords for PICS

Mnemonic	PICS item
PIC_BASICVEH	[x] A.2/1
PIC_BASICIRS	[x] A.3/1
PIC_EMERVEH	[x] A.2/2
PIC_PUBTRANSVEH	[x] A.2/3
PIC_DOOROPEN	[x] A.16/28
PIC_LIGHTBARINUSE	[x] A.16/3
PIC_SIRENEINUSE	[x] A.16/4
NOTE: [x] being the bookmark to the PICS standard, in the reference clause of the TP document.	

8.2.7 TP behaviour

First of all, the following global rules apply, when writing the behaviour description:

- The behaviour description is written in an explicit, exhaustive and unambiguous manner.
- The behaviour description only refers to externally observable test events (send/receive PDUs, timer, counters, etc.) or to events or states, which can be directly or indirectly observed externally.
- All test events used in the behaviour description are part of the procedures specified in the protocol standards.
- The wording of the test events in the behaviour description is explicit, so that the ATS writers do not have to interpret the behaviour description.
- All test events in the behaviour description should result as far as possible in one ATS statement (for instance a TTCN statement).

The test behaviour is described in prose. This enables to use different ways to express similar behaviour. But using different expressions to define identical behaviours can lead to some misinterpretation of the test purposes. Also the meaning and the expected order of the test event have a clear and unique meaning for different readers.

Thus, the present document recommends to use pre-defined keywords in order to express clearly and uniquely the test behaviour.

Table 12 shows some recommended pre-defined keywords and their context of usage. The pre-defined keywords are also likely to be used in combination with the "{" "}"delimiters, in order to clearly delimitate their action in the test behaviour description.

Table 12 does not present an exhaustive list, so that additional keywords might be defined as necessary. The definition of additional keywords is included in the corresponding TSS&TP document.

Table 12: List of pre-defined keywords for the behaviour description

Behavioural keywords	
`with`	`with`, together with `"{" "}"` delimiters is used to express the initial conditions, which consist of a set of events, to be executed before starting with the test behaviour corresponding to the test objective. EXAMPLE: `With { the IUT having sent a valid LinkSetup message and ... }`

`ensure that`	`ensure that`, together with "{" "}" delimiters is used to define the place of the expected behaviour (TP body) or the final conditions.
	EXAMPLE:
	`ensure that {`
	`   when { the IUT receives a valid LinkSetup message... }`

`when/then`	`when` combined with `then` enables to define the test behaviour involving a combination of stimuli and response events. The when/then combination is used when the occurrence of an event is triggered by the realization of a previous event.
	EXAMPLE:
	`ensure that {`
	`   when {`
	`      a crash signal is activated }`
	`   then {`
	`      the IUT sends a CAM message containing crashStatus TaggedValue indicating "True"} }`

Event keywords	
`the IUT`	Event in the TP is expressed from the point of view of the IUT. This avoid any misinterpretation.
`receives`	states for an event corresponding to the receipt of a message by the IUT.
`having received`	states for a condition where the IUT has received a message.
`sends`	states for an event corresponding to the sending of a message by the IUT.
`having sent`	states for a condition where the IUT has sent a message.

`from/to`	Indicates the destination or the origin of a message as necessary (interface, ...)
	EXAMPLE:
	`ensure that {`
	`   when { the IUT receives a valid Paging message from the Broadcast port.. }`

`on expiry of`	Indicate the expiry of a timer, being a stimulus for forthcoming event. EXAMPLE: `ensure that { on expiry of the Timer T202, the IUT sends a valid LinkShutdown message...`
`after expiry of`	Used to indicate that an event is expected to occur after the expiry of a timer. EXAMPLE: `ensure that { the IUT sends a valid CAM message after expiry of the minimum timer interval }`
`before expiry of`	Used to indicate that an event is expected to occur before the expiry of a timer. EXAMPLE: `ensure that { the IUT sends a valid CAM message before expiry of the maximum timer interval }`
Event attribute keywords	
`valid`	Indicates that the event sent or received is a valid message according to the protocol standard, thus: • containing all mandatory parameters, with valid field values; • containing required optional fields according to the protocol context, with valid field values.
`invalid`	Indicates that the event sent or received is a invalid message according to the protocol standard. Further details describing the invalid fields of the message is added. EXAMPLE: `With { the IUT having sent an invalid LinkSetup message containing no mandatory LinkCapability parameter... }`
`containing`	Enables to describe the content of a sent or received message
`indicating`	Enables to specify the interpretation of the value allocated to a message parameter. EXAMPLE: `With { the IUT having sent a valid LinkSetup message containing a mandatory LinkCapability parameter indicating "Variable Rate supported"... }`

Logical keywords	
`and`	Used to combine statements of the behaviour description.
`or`	
`not`	

Table 13: TP example for CAM DangerousGoods

TP Id	TP/CAM/INA/DAG/BV/01
Test objective	Checks that CAM message includes DangerousGoods information if they are transported
Reference	TS 102 637-2 [i.18], clauses 7.1 and 7.2
PICS Selection	PICS_BASICVEH or PICS_EMERVEH or PICS_PUBTRANSVEH
Initial conditions	

with {
   the IUT being in the "initial state" and
   the IUT having sent a valid CAM message
      not containing DangerousGoods TaggedValue
}	

Expected behaviour	

ensure that {
   when {
      Dangerous goods are transported
   }
   then {
      the IUT sends a valid CAM message
         containing DangerousGoods TaggedValue indicating value > 0
   }
}	

8.3 Development of TTCN-3 test suite

8.3.1 Global TTCN-3 test architecture

This clause presents the global ITS TTCN-3 test architecture to be used as a basis to develop further TTCN-3 test suites. This information will be used to provide the ATS documentation as part of the TTCN-3 test specification deliverables.

Figure 14 illustrates the global ITS TTCN-3 test architecture by showing elements which can be used in the context of ITS. The components shown in this figure cannot be used simultaneously in a single test suite. Thus this figure shall be interpreted as a catalogue of TTCN-3 components, which may be selected, as applicable, in a test suite.

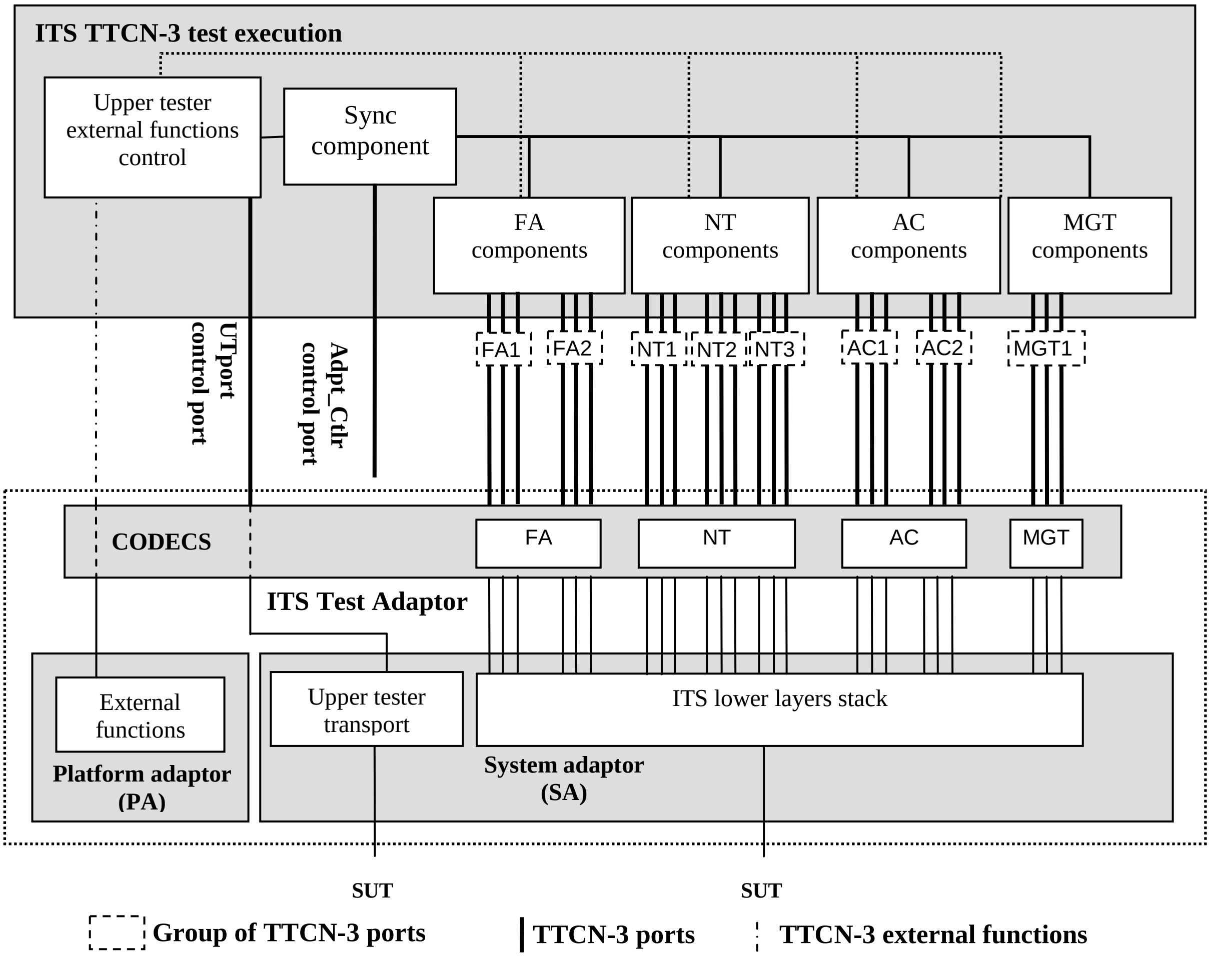

Figure 14: Global ITS test architecture

The following protocol test components are available. Each of these components can be mapped to either a "Main Test Component" (MTC), when only one component shall be executed, or to a "Parallel Test Component" (PTC), if several components shall be used.

- **FA:** This component executes the test cases belonging to facilities layer test groups, e.g. CAM, DENM, legacy CI port manager, 15628 kernel emulator and uses the FA port groups.
- **NT:** This component executes the test cases belonging to networking & transport layer test groups, e.g. BTP, GeoNetworking, IPv6 over GeoNetworking, FAST and future networking & and transport protocols, and uses the NT port groups.
- **AC:** This component executes the test cases belonging to access layer test groups, e.g. MAC M5, MAC IR and future access protocols, and uses the AC port groups.
- **MGT:** This component executes the test cases belonging to management layer test groups, e.g. Inter-ITS-SCU communication and future management protocols, and uses the MGT port groups.

The **Sync** component can be mapped to the MTC when several PTCs are necessary. This component triggers and synchronizes the PTCs in order to orchestrate the protocol behaviour executed by the PTCs. In addition, it starts and terminates the test cases.

The **upper tester external functions control** represents the functions which the protocol test components may use to control the upper tester which is located in the SUT. These functions may use either external functions or dedicated messages and a dedicated port to realise synchronisation. These functions can be easily executed in the protocol test components.

As figure 14 presents a generic ITS test architecture only, it can be modified in order to match further testing requirements.

The "ITS test adapter" (TA) is a part of the test system, which is partially specified in [i.4], but the implementation of the TA, which is vendor-dependent, cannot be considered as fully abstract. According to [i.4], the TA consists of the following elements:

- "System adapter" (SA): It includes the "ITS Lower Layers Stack" for establishing the communication with the IUT as well as the "Upper Tester Transport". The "ITS lower layers stack" is configurable by the Adpt_Ctrl port.
- "Platform adapter" (PA): It includes the implementation of the set of external functions declared in the test suite.
- "CODECS": It encodes and decodes messages between the TTCN-3 internal data representation and the format required by the related base standard. As necessary, it may also be used to encode and decode messages for the upper tester synchronisation.

The global ITS test architecture uses a set of ports. These ports are classified into data ports (e.g. FA1, FA2, NT1, NT2, NT3, AC1, AC2, MGT1) and control ports (e.g. "Adpt_Ctlr" and "Upper tester"):

- Data ports: They are used to send protocol messages to the IUT and to receive protocol messages from the IUT.
- Control Ports: They are used to configure and control the "Test adapter" (TA) by passing necessary parameters to the various entities in the TA. The upper tester port is part of the control ports.

Mapping of IUTs onto proper data ports for sending and receiving different messages is presented in table 14.

Table 14: Mapping between IUTs and port groups

Port Type	Component	Port Group	IUTs
Data	FA	FA1	CAM, DENM
		FA2	CVIS/COOPER Facilities
	NT	NT1	BTP
		NT2	GeoNetworking, IPv6 over GeoNetworking
		NT3	FAST
	AC	AC1	MAC M5
		AC2	MAC IR
	MGT	MGT1	Inter-ITS-SCU communication
Control	N/A	Adpt Ctlr	N/A
	N/A	Upper Tester	N/A

Table 14 only presents a current view of some IUTs. The mapping of further IUTs onto port groups can be included in the table.

8.3.2 Importing ASN.1 definition

The ITS set of standards uses ASN.1 for the definition of the message types. ES 201 873-7 [i.4] specifies the usage of ASN.1 types and values with TTCN-3. Following ES 201 873-1 [i.4] and ES 201 873-7 [i.4], ensure that the usage of ASN.1 definition is fully harmonized with TTCN-3.

The process for using ASN.1 data types and values in TTCN-3 modules consists of importing the existing ASN.1 productions. For this purpose, the TTCN-3 "**`import from`**" statement should be used, in association with the "**`language`**" statement. Several ASN.1 language version are currently available:

- "**`ASN.1:2008`**",
- "**`ASN.1:2002`**",
- "**`ASN.1:1997`**",
- "**`ASN.1:1994`**",
- "**`ASN.1:1988`**".

Before importing ASN.1 modules in TTCN-3, they should fully comply with the corresponding ASN.1 syntax according to the version used in the import statement.

ASN.1 modules to be imported, are provided in the source code library accordingly to the TTCN-3 tool requirements.

When ASN.1 productions are imported in a TTCN-3 module, all ASN.1 data and values, imported through the "**`import from`**" statement are available in equivalent TTCN-3 types and values . Therefore, ASN.1 types have associated TTCN-3 types, which are specified in ES 201 873-7 [i.4], clause 8. The usage of the imported ASN.1 types and values should be according to these associated types.

ASN.1 type and value identifiers are converted with the same identifiers in TTCN-3. Therefore, two transformations may apply:

- Hyphen "-" characters, which are not allowed in TTCN-3 are replaced by underscores "_";
- If TTCN-3 keywords are used as ASN.1 identifiers, then a single underscore "_" is appended at the end of the identifier.

The imported ASN.1 data and values follow the type rules, which apply to their corresponding associated TTCN-3 types.

The conversion of ASN.1 to TTCN-3, performed by the TTCN-3 tools, can follow two possible approaches:

- Implicit: ASN.1 types and data are imported and build an internal representation that is not available to the user. The imported ASN.1 types and data are transparently available in the TTCN-3 modules where they were imported.
- Explicit: ASN.1 modules are converted in TTCN-3 modules, which are visible from the users. While using the import statement with the ASN.1 language, the TTCN-3 tool uses the TTCN-3 modules resulting from the conversion.

These two approaches are compatible.

8.3.3 The TTCN-3 naming conventions

TTCN-3 core language contains several types of elements with different rules of usage. Applying naming conventions aims to enable the identification of the type when using specific identifiers according to the type of element.

For instance, a variable declared in a component has different scoping rules than a local variable declared in a test case. Then identifiers of component variables are different from identifiers of local variables, in order to recognized which type of variable the identifier belongs to.

Furthermore, applying naming conventions maintains the consistency of the TTCN-3 code across the test suites, and thus increase the readability for multiple users and ease the maintenance.

Table 15 shows the generic naming conventions applied in the ETSI test suites.

Table 15: ETSI TTCN-3 generic naming conventions

Language element	Naming convention	Prefix	Example identifier
Module	Use upper-case initial letter	none	IPv6Templates
Group within a module	Use lower-case initial letter	none	messageGroup
Data type	Use upper-case initial letter	none	SetupContents
Message template	Use lower-case initial letter	m_	m_setupInit
Message template with wildcard or matching expression	Use lower-case initial letters	mw_	mw_anyUserReply
Modifying message template	Use lower-case initial letter	md_	md_setupInit
Modifying message template with wildcard or matching expression	Use lower-case initial letters	mdw_	mdw_anyUserReply
Signature template	Use lower-case initial letter	s_	s_callSignature
Port instance	Use lower-case initial letter	none	signallingPort
Test component instance	Use lower-case initial letter	none	userTerminal
Constant	Use lower-case initial letter	c_	c_maxRetransmission
Constant (defined within component type)	Use lower-case initial letter	cc_	cc_minDuration
External constant	Use lower-case initial letter	cx_	cx_macId
Function	Use lower-case initial letter	f_	f_authentication()
External function	Use lower-case initial letter	fx_	fx_calculateLength()
Altstep (incl. Default)	Use lower-case initial letter	a_	a_receiveSetup()
Test case	Use ETSI numbering	TC_	TC_COR_0009_47_ND
Variable (local)	Use lower-case initial letter	v_	v_macId
Variable (defined within a component type)	Use lower-case initial letters	vc_	vc_systemName
Timer (local)	Use lower-case initial letter	t_	t_wait
Timer (defined within a component)	Use lower-case initial letters	tc_	tc_authMin
Module parameters for PICS	Use all upper case letters	PICS_	PICS_DOOROPEN
Module parameters for other parameters	Use all upper case letters	PX_	PX_TESTER_STATION_ID
Formal Parameters	Use lower-case initial letter	p_	p_macId
Enumerated Values	Use lower-case initial letter	e_	e_syncOk

Next to such general naming conventions, table 16 shows specific naming conventions that apply to the ITS TTCN-3 test suite.

Table 16: ITS specific TTCN-3 naming conventions

Language element	Naming convention	Prefix	Example identifier
ITS Module	Use upper-case initial letter	Its"IUTname"_	ItsCam_
Module containing types and values	Use upper-case initial letter	Its"IUTname"_TypesAndValues	ItsCam_TypesAndValues
Module containing Templates	Use upper-case initial letter	Its"IUTname"_Templates	ItsCam _Templates
Module containing test cases	Use upper-case initial letter	Its"IUTname"_TestCases	ItsCam _TestCases
Module containing functions	Use upper-case initial letter	Its"IUTname"_Functions	ItsCam _Functions
Module containing external functions	Use upper-case initial letter	Its"IUTname"_ExternalFunctions	ItsCam_ExternalFunctions
Module containing components, ports and message definitions	Use upper-case initial letter	Its"IUTname"_Interface	ItsCam _Interface
Module containing main component definitions	Use upper-case initial letter	Its"IUTname"_TestSystem	ItsCam _TestSystem
Module containing the control part	Use upper-case initial letter	Its"IUTname"_TestControl	ItsCam _TestControl

Besides these naming conventions, further recommendations are proposed with regard of:

- Structure of data:
  - Types should be defined in alphabetic order within TTCN-3 groups within the same TTCN-3 module.

- All message types referenced in port type definitions and related to the same interface should be defined in the module, where these ports are defined.

- Log Statement:
  - All TTCN-3 log statements must follow the following format:
    - Three asterisk should be used to precede the log text.
    - The TTCN-3 test case/function identifier in which the log statement is defined should follow.
    - One of the categories INFO, WARNING, ERROR, PASS, FAIL, INCONC, TIMEOUT should follow.
    - Free text should follow.
    - And the log text should end with three asterisk.

EXAMPLE: log("*** f_sendMsg: INFO: Message has been sent ***")

  - Any invocation of an external function must be followed by log statement.
  - Each TTCN-3 setverdict statement that sets a test component verdict to INCONC or FAIL should be preceded by a log statement or log statement feature as first defined in TTCN-3 version 3.4.1 should be used, where the comment is part of the setverdict statement.

    ITS TTCN-3 Test suite preferably is regularly checked by the ETSI T3Q tool in order to keep the readability, consistency and maintainability of the TTCN-3 code.

8.3.4 TTCN-3 code documentation

In order to document the TTCN-3 code, the ITS TTCN-3 test suite must follow the ES 201 873-10 [i.4].

Some of the most important rules are the following:

- All TTCN-3 testcase, altstep and function statements should be documented at least with @desc tag and @param for each parameter.
- All TTCN-3 modulepar statements should be documented at least with @desc tag.
- TTCN-3 modules should be documented at least @desc and @version tag.

8.3.5 Test cases structure

In order to keep the readability, consistency and maintainability of the TTCN-3 code, ITS test cases are implemented following a certain structure. This framework defines this structure as follows:

- Local variables, contains the declaration of the variables to be used within the test case.
- Test control PICS, contains one or more logical expressions that use module parameters (PICS or PIXIT) in order to decide whether or not to run the test case.
- Initialization of component variables, contains zero or more assignments that initialize or modify the value of component variables.
- Test component configuration, initialized or configures the components needed to run the test.
- Test adapter configuration (optional), configures the test adapter as needed to run the test.
- Preamble, contains the actions needed to bring the IUT into the desired state.
- Test body, contains the actions needed to run the test case. This part will set the verdict of the test.
- Postamble, contains the actions needed to bring the IUT into the initial state or the desired final state.

These are the minimum sections that a test case must contain (even being empty) but this structure may be complemented or extended with another sections as needed.

8.4 Provision of IFS pro-forma

An "Interoperable Functions Statement" (IFS) identifies standardised functions that an EUT must support. These functions are either mandatory, optional or conditional (depending on other functions).

In addition, the IFS can be used as a proforma by a manufacturer to identify the functions an EUT will support when interoperating with corresponding equipment from other manufacturers.

The ideal starting point in the development of an IFS is the "Protocol Implementation Conformance Statement" (PICS) which should clearly identify the tested protocol's options and conditions. Like the PICS, the IFS should be considered part of the base protocol specification and not a testing document.

The guidance to produce IFS proforma is provided in EG 202 237 [i.5] and no extra guidance is required for the context of ITS.

8.5 Provision of Test Descriptions

A "Test Description" (TD) is a well detailed description of a process that pretends to test one or more functionalities of an implementation. Applying to interoperability testing, these testing objectives address the interoperable functionalities between two or more vendor implementations.

In order to ensure the correct execution of an interoperability test, the following information should be provided by the test description:

- The proper configuration of the vendor implementations.
- The availability of additional equipment (protocol monitors, functional equipment, …) requires to achieve the correct behaviour of the vendor implementations.
- The correct initial conditions.
- The correct sequence of the test events and test results.

TDs are based on the test scenarios.

In order to facilitate the specification of test cases an interoperability test description should include as a minimum [i.6], see table 17.

Table 17: Interoperability test description

Identifier	a unique test description ID
Summary	a concise summary of the test which should reflect the purpose of the test and enable readers to easily distinguish this test from any other test in the document
References	a list of references to the base specification section(s), use case(s), requirement(s), TP(s) which are either used in the test or define the functionality being tested
Applicability	a list of features and capabilities which are required to be supported by the SUT in order to execute this test (e.g. if this list contains an optional feature to be supported, then the test is optional)
Configuration or Architecture	a list of all required equipment for testing and possibly also including a (reference to) an illustration of a test architecture or test configuration
Pre-Test Conditions	a list of test specific pre-conditions that need to be met by the SUT including information about equipment configuration, i.e. precise description of the initial state of the SUT required to start executing the test sequence
Test Sequence	an ordered list of equipment operation and observations. In case of a conformance test description the test sequence contains also the conformance checks as part of the observations

The TDs play a similar role as TPs for conformance testing.

Interoperability Test Description			
Identifier:	TD_XYZ_ 0001		
Summary:	Roadside A only relays message from vehicle A to vehicle B, when vehicle B is attached to Roadside A		
Configuration:	CF_RS_XYZ		
SUT	ITS_B		
References	**Test Purpose**		**Specification Reference**
	XYZ (as applicable)		ES xxx yyy clause x.y.z
Use Case ref.:	XYZ (as applicable)		
Pre-test conditions:	• Roadside unit A is ready • Vehicle A is attached to the Roadside unit A • Vehicle B is attached to the Roadside unit A		
Test Sequence:	**Step**	**Type**	**Description**
	1	stimulus	Vehicle A sends message to Vehicle B
	2	verify	Vehicle B receives message
	3	check	Roadside A sends message with the same payload to vehicle B
	4	configure	Detach vehicle B from Roadside A
	5	stimulus	Vehicle A sends message to Vehicle B
	6	check	when { Vehicle A sends message to Vehicle B} then { Roadside A does not relay message with the same payload to vehicle B}

Types of events:

- A **stimulus** corresponds to an event that enforces an EUT to proceed with a specific protocol action, like sending a message for instance.
- A **verify** consists of verifying that the EUT behaves according to the expected behaviour (for instance the EUT behaviour shows that it receives the expected message).
- A **configure** corresponds to an action to modify the EUT configuration.
- A **check** ensures the receipt of protocol messages on reference points, with valid content. This "check" event type corresponds to the interoperability testing with conformance check method.

Table 18 presents a possible example of an interoperability test description for GeoNetworking.

Table 18: Possible example of interoperability TD for GeoNetworking

Interoperability Test Description			
Identifier:	TD_GEO_ 01		
Summary:	To check that the SUT exchanges correctly information using GeoUnicast forwarding mechanism		
Configuration:	CF#1		
SUT	Vehicle 1, Vehicle 2 and Vehicle 3		
References	**Test Purpose**		**Specification Reference**
	N/A		TS 102 636-4-1 [i.17]
Use Case ref.:	Geo_#1		
Pre-test conditions:	• Vehicle 1 and Vehicle 2 are in the same coverage area • Vehicle 2 and Vehicle 3 are in the same coverage area • Vehicle 1 and Vehicle 3 are not in the same coverage area • Vehicle 1, Vehicle 2 and Vehicle 3 are receiving beacon messages • Vehicle 2 has GeoUnicast forwarding mechanism activated		
Test Sequence:	**Step**	**Type**	**Description**
	1	stimulus	Vehicle 1 generates information to be sent to Vehicle 3
	2	check	Vehicle 1 sends the GeoUnicast message containing the information, as payload, to Vehicle 3 designating Vehicle 2 as next hop
	3	verify	Vehicle 2 receives and buffers the GeoUnicast message sent by Vehicle 1
	4	check	Vehicle 2 sends forward the GeoUnicast message previously received to Vehicle 3
	5	verify	Vehicle 3 receives the GeoUnicast message sent by Vehicle 2 containing the information sent by Vehicle 1

Message flows can also be used to illustrate the sequences of messages exchanged between the EUTs, see the example in figure 15.

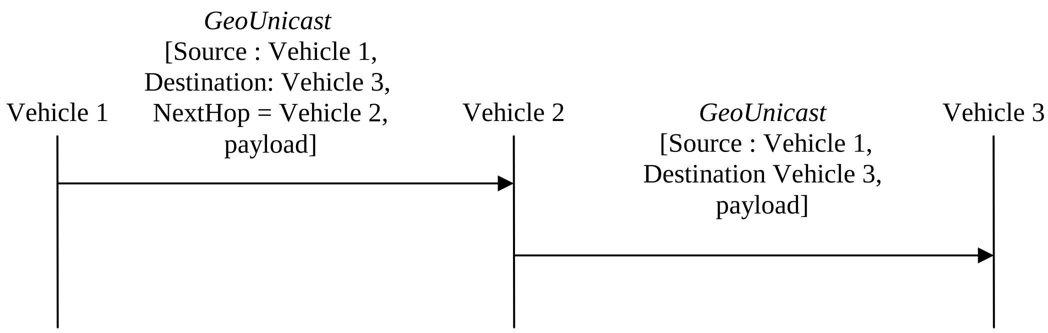

Figure 15: Example of a message sequence chart

Annex A: Introduction to Interoperability testing

A.1 Principles

Comparison with conformance testing helps to understand the goals and values of interoperability testing. Conformance testing requires using protocol tester and test scripts, which provides the behaviour corresponding to the test purposes. This enables testing of both, valid and invalid behaviours, but requires writing test scripts and implementing these test scripts in a protocol tester environment. This development process costs time and efforts. Furthermore, testing conformance to the protocol specifications may not guaranty interoperability.

Vendors are interested to test interoperability of their implementations with other vendor implementations, before any test scripts for conformance testing are ready to be executed. This is where Interoperability testing can provide a pragmatic solution, to start testing implementations.

As implementations conforming to a standard may not be interoperable, and interoperable implementations may not conform to a standard, experience shows that interoperability testing is complementary to conformance testing.

Interoperability testing consists simply in inter-operating different vendor implementations, which are supposed to be inter-operable according to the expected conformance with the base standards (see clause 4 in [i.5]). Even if this process looks easy, it requires specifying a complete environment enabling to operate vendors implementation as in real conditions. The complete set of all vendors implementation involved in interoperability tests, together with the set of equipment required to enable vendors implementations to execute the test process is named the "Test Bed".

A.2 Interoperability testing process

Usually, interoperability testing is executed during interoperability test events, where implementations from different vendors meet to execute test sessions, combining their implementations. Alternatively, interoperability test session can be executed in vendor's labs and also in accredited test labs as part of a certification process.

The execution of interoperability tests requires manual control by test operators (the automatic control will be presented later in the present document). Usually , test operators are representative of vendors, who take the following actions during the test execution:

- Configure their implementation to match other vendors configuration and to ensure the implementation will follow the expected test behaviour.
- Trigger the implementation to execute the expected test process (for instance: establish a link or start registering with a proxy, …).
- Verify that the implementation behaves according to the expected behaviour (for instance the implementation indicates having received a valid message, …).
- Optionally check for the compliance of messages received on certain interfaces by using a protocol monitor for instance.
- Etc.

Alternatively, test operators can also be representatives of a test labs.

The assessment of the correct test behaviour (pass verdict) results from the observation of the behaviour of the vendor implementations, and optionally the checking of protocol messages.

During an interoperability test session, several different implementation behaviours can be tested. Each test behaviour corresponds to a test scenario.

A test scenario is a non-formal description of the expected test behaviour. Test scenarios are usually prepared by vendors prior to interoperability test events.

A.3 Interoperability testing with conformance checks

One possible testing approach is combining well-known techniques of testing: conformance testing and interoperability testing. Although this approach cannot offer the same level of details as either conformance or interoperability testing do individually, this technique may be useful in some cases. This new approach is often used when conformance test specifications do not exist for a particular standardized protocol. For more details about combining conformance and interoperability testing see [i.5].

Conformance checks within interoperability testing are placed at normal communication interfaces rather than dedicated interfaces used for testing purposes. That is the main reason why conformance checks cannot provide the same level of detail as conformance testing individually.

A.4 Automated interoperability testing

Interoperability testing can also be executed automatically, using a test pilot. The benefit of the automated interoperability testing is as following:

- Accelerate the test execution, for instance checking message content requires several minutes for a test operator against few milliseconds for an automated test bed.
- Ensure that each execution of a test is identical.
- Reduce the required manpower to execute the interoperability tests.

The degree of test automation is often a compromise between testing requirements and feasibility. It may not be profitable, for example, to automate the entire testing process since the resources required to implement all or part of the tests may be prohibitively expensive. In addition, some interfaces may require significantly less effort to assess manually.

The test descriptions used for the manual interoperability testing are still valid for the automated interoperability testing. Actually, the test pilots may be implemented using test scripts in order to process all the actions processed by the test operators during the manual interoperability testing.

EG 202 810 [i.6] defines a general framework for automated interoperability testing mainly focused on the development of the test bed which is perfectly applicable and useful for ITS interoperability testing. This framework recommends following the present document which defines the process of developing automated test bed:

- Interoperability test bed design:
  - Define test configuration.
  - Define message structure.
  - Define test parameters.
- Specify the Test Cases:
  - Equipment operation repository.
  - Conformance check repository.
- Implementation of Codecs and Adaptation Functions.
- Validation.

Some of the actions that the test scripts should include are the following:

- Configuring and triggering the vendor implementations.
- Configuring the simulation environment.
- Checking protocol messages.

- Gathering all the information needed in order to assess the test result.
- Assigning a verdict.

EG 202 810 [i.6] provides guidance on the design and implementation of automated interoperability testing. The test descriptions used for the manual interoperability testing are still valid for the automated interoperability testing. The test pilots may be implemented using test scripts in order to process all the actions triggered by the test operators during the manual interoperability testing.

EG 202 810 [i.6] defines a general framework for automated interoperability testing mainly focused on the development of the test bed which is perfectly applicable and useful for ITS interoperability testing. This framework recommends following the present document which defines the process of developing automated test bed:

- Interoperability test bed design:
  - define test configuration;
  - define message structure;
  - define test parameters.
- Specify the Test Cases:
  - equipment operation repository;
  - conformance check repository.
- Implementation of Codecs and Adaptation Functions.
- Validation.

Some of the actions that the test scripts should include are the following:

- configuring and triggering the vendor implementations;
- configuring the simulation environment;
- checking protocol messages;
- gathering all the information needed in order to assess the test result;
- assigning a verdict.

EG 202 810 [i.6], provides guidance on the design and implementation of automated interoperability testing. Figure A.1 presents the general test bed architecture for automated interoperability testing directly extracted from [i.6].

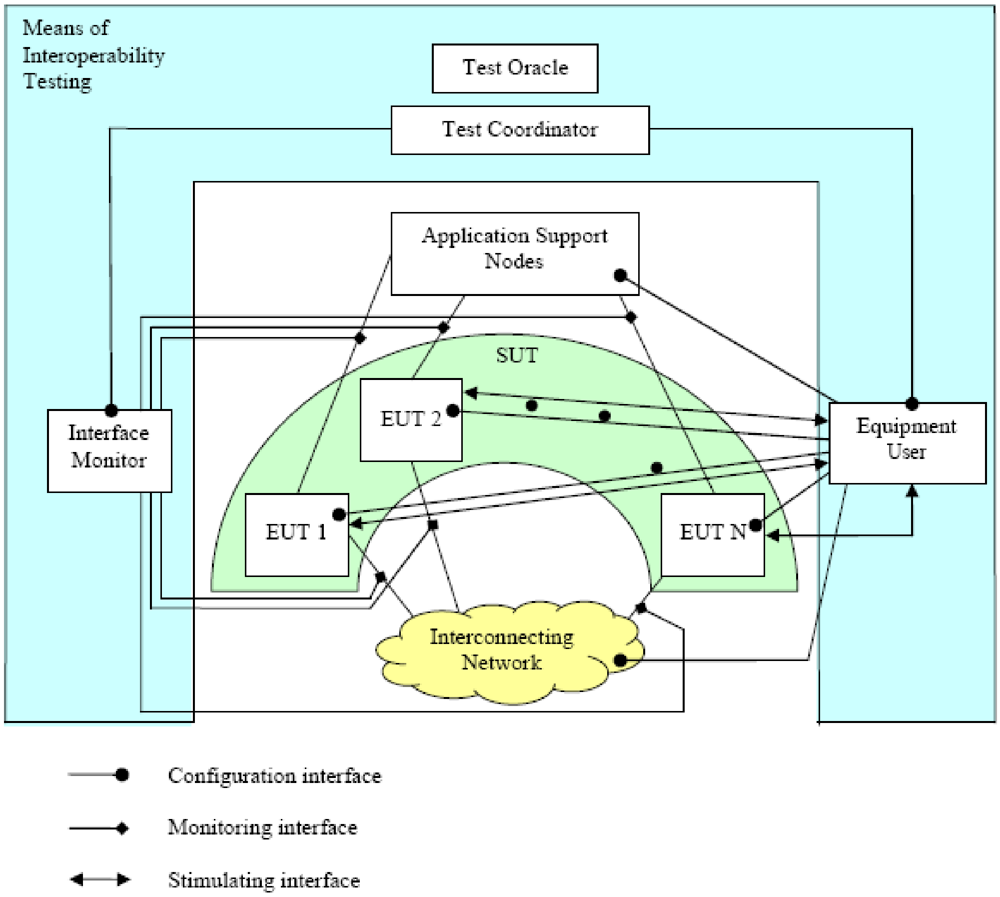

Figure A.1: General architecture for interoperability testing

The test bed architecture is composed of the following entities:

- System Under Tests (SUT): it is composed of two or more EUTs that are assessed for their interoperability.
- Application Support Nodes: they include all the devices involved in providing the service or functionality to the end user which are not object of the test.
- Interconnecting network: it includes all the devices needed to provide the required network connections.
- Means of Interoperability Testing: it includes entities (Test Coordinator, Test Oracle, Interface Monitor, Equipment User) that stimulate or control the SUT or configure the interconnecting network as well as the application support nodes.

History

Document history		
V1.1.1	November 2010	Membership Approval Procedure MV 20110104: 2010-11-05 to 2011-01-04
V1.1.1	January 2011	Publication