



BSI Standards Publication

Cooperative intelligent transport systems (C-ITS) — Global transport data management (GTDM) framework

National foreword

This Published Document is the UK implementation of CEN ISO/TS 21184:2021. It is identical to ISO/TS 21184:2021.

The UK participation in its preparation was entrusted to Technical Committee EPL/278, Intelligent transport systems.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

This publication is not to be regarded as a British Standard.

© The British Standards Institution 2021
Published by BSI Standards Limited 2021

ISBN 978 0 580 93449 0

ICS 03.220.01; 35.240.60

Compliance with a Published Document cannot confer immunity from legal obligations

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 30 April 2021.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

English Version

Cooperative intelligent transport systems (C-ITS) - Global
transport data management (GTDM) framework (ISO/TS
21184:2021)

Systèmes de transport intelligents coopératifs (C-ITS) -
Cadre de gestion globale des données de transport
(GTDM) (ISO/TS 21184:2021)

Intelligente Transportsysteme - Sichere
Fahrzeugschnittstelle - Datenverzeichnis für C-ITS
anwendungsrelevante Fahrzeuginformationen (ISO/TS
21184:2021)

This Technical Specification (CEN/TS) was approved by CEN on 12 December 2020 for provisional application.

The period of validity of this CEN/TS is limited initially to three years. After two years the members of CEN will be requested to submit their comments, particularly on the question whether the CEN/TS can be converted into a European Standard.

CEN members are required to announce the existence of this CEN/TS in the same way as for an EN and to make the CEN/TS available promptly at national level in an appropriate form. It is permissible to keep conflicting national standards in force (in parallel to the CEN/TS) until the final decision about the possible conversion of the CEN/TS into an EN is reached.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

European foreword

This document (CEN ISO/TS 21184:2021) has been prepared by Technical Committee ISO/TC 204 "Intelligent transport systems" in collaboration with Technical Committee CEN/TC 278 "Intelligent transport systems" the secretariat of which is held by NEN.

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

According to the CEN-CENELEC Internal Regulations, the national standards organizations of the following countries are bound to announce this Technical Specification: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

Endorsement notice

The text of ISO/TS 21184:2021 has been approved by CEN as CEN ISO/TS 21184:2021 without any modification.

Contents

Page

Foreword	v
Introduction	vi
1 Scope	1
2 Normative references	1
3 Terms and definitions	2
4 Abbreviated terms	3
5 Conventions	5
6 Global Transport Data Management (GTDM) framework	5
6.1 General	5
6.2 Applicable use case groups	7
6.2.1 General	7
6.2.2 GTDM framework operation	8
6.2.3 GTDM certificate-based access control	9
6.2.4 Proximity use case applications	10
6.2.5 On-board use case applications	11
6.2.6 Remote use case (ITS, product-related) applications	12
6.2.7 Common configuration data authoring process	13
6.3 GTDM Basic principles	14
7 ITS-S global transport data model	15
7.1 Overview	15
7.2 GT basic data model	15
7.2.1 Overview	15
7.2.2 GT basic configuration	16
7.2.3 GT language package	36
7.2.4 GT runtime data	37
7.2.5 GT condition	44
7.3 GT access control data model	48
7.3.1 <class> GTACProtocolConfig	48
7.3.2 <class> ACProtocol	48
7.3.3 <class> ACProtocolService	49
7.3.4 <class> ACProtocolRole	49
7.3.5 <class> GTACPermissionConfig	49
7.3.6 <class> ACEcuDataParamAccessSet	50
7.3.7 <class> ACEcuDataParamAccess	50
7.3.8 <class> ACAccessDetail	51
7.4 GT function monitor data model	51
7.4.1 <class> GTFMConfig	51
7.4.2 <class> FMBrick	53
7.4.3 <class> FMDataParamCopy	56
7.4.4 <class> FMMessageSequence	56
7.5 SCN data model (optional)	57
7.5.1 <class> SCNDFConfig	57
7.5.2 <class> ProtocolType	60
7.5.3 <class> DlcPin	61
7.5.4 <class> ProtocolAddressing	62
7.5.5 <class> EcuAddress	62
7.5.6 <class> PhysicalEcuAddress	63
7.5.7 <class> ProtocolService	63
7.5.8 <class> ProtocolResponseMessage	64
7.5.9 <class> PackageScheduler	64
7.5.10 <class> PackageSource	65
7.5.11 <class> PackageMapping	65

7.5.12	<class> DtcScheduler	66
7.5.13	<class> CompuMethod	67
7.5.14	<class> CompuScale	69
7.5.15	<class> CompuLimit	69
7.5.16	<class> CompuItem	70
7.5.17	<class> CompuRational	70
7.5.18	<class> CompuTextToValue	70
7.5.19	<class> ExceptionRule	71
7.5.20	<class> ExceptionRuleElement	71
8	ITS-S global transport data management	72
8.1	General	72
8.2	ITS-S Management	73
8.2.1	Configuration Provider (M1)	73
8.2.2	Startup sequence	74
8.3	ITS-S Security	75
8.3.1	General	75
8.3.2	ITS application identifier (ITS-AID)	76
8.3.3	Certificates and keys for use with ISO/TS 21177 secure sessions	76
8.3.4	Access control (S1)	77
8.3.5	Security sub-layer (S2)	78
8.4	ITS-S Facilities	78
8.4.1	Data Provider (F1)	78
8.4.2	SCN Handler (F2)	82
8.4.3	GTP Handler (F3)	84
8.4.4	Function Monitor (F4)	89
8.5	ITS-S Network layer and Transport layer	91
8.5.1	ITS Network Handler (N1)	91
8.5.2	SC Network Handler (N2)	92
8.6	ITS-S Applications	92
8.6.1	Application (A1)	92
9	Service Access Points (SAP)	93
9.1	General	93
9.2	SAP – Management to Application	93
9.3	SAP – Management to Security	93
9.4	SAP – Management to Facilities	94
9.5	SAP – Management to Network	94
9.6	SAP – Security to Facilities	94
9.7	SAP – Security to Network	95
9.8	SAP – Network to Facilities	95
10	ASN.1 definition	96
Annex A (normative) Global Transport ASN.1 definition		97
Bibliography		115

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*, in collaboration with the European Committee for Standardization (CEN) Technical Committee CEN/TC 278, *Intelligent transport systems*, in accordance with the Agreement on technical cooperation between ISO and CEN (Vienna Agreement).

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

This document is part of a family of deliverables from Standard Development Organizations (SDOs) for Cooperative Intelligent Transport Systems (C-ITS), which is a subset of standards for Intelligent Transport Systems (ITS).

ITS aims to improve surface transportation in terms of:

- **safety**
e.g. crash avoidance, obstacle detection, emergency calls, dangerous goods;
- **efficiency**
e.g. navigation, green wave, priority, lane access control, contextual speed limits, car sharing;
- **comfort**
e.g. telematics, parking, electric vehicle charging, infotainment; and
- **sustainability,**

by applying information and communication technologies (ICT).

The whole set of standards for deployment of C-ITS is difficult to understand for developers of equipment and software, especially ITS application software, and thus guidelines explaining a beneficial choice of standards (C-ITS release), the purpose and interaction of standardized features, beneficial implementation approaches, and guidance in developing ITS applications are a prerequisite for a fair and open market allowing early deployment of interoperable and future-proof solutions; see ISO/TR 21186-1. More details on the C-ITS domain can be found in the Brochure^[14] produced by CEN/TC 278.

Referencing other SDOs and their respective deliverables is in no way to be understood as an endorsement, but rather as an informative piece of information.

At the time of writing this document, no applicable Intellectual Property Rights (IPR) issues were known related to this document. However, this document references standards for which IPRs are known. Information on such IPRs is expected to be provided in those respective standards, which might be from any one of the SDOs working on ITS or C-ITS.

Cooperative intelligent transport systems (C-ITS) — Global transport data management (GTDM) framework

1 Scope

This document specifies a global transport data management (GTDM) framework composed of

- global transport basic data model,
- global transport access control data model,
- global transport function monitor data model, and
- sensor and control network data model

to support data exchange between applications.

This document defines standardized data classes in a Global Transport Data Format (GTDF), and the means to manage them.

Application and role-based access control to resources in GTDF are specified in accordance with IEEE 1609.2 certificates.

This document specifies GTDM as an ITS-S capability which is an optional feature (ITS-capabilities are specified in ISO 24102-6).

The GT access control (GTAC) data model specifies access permissions to data and function control by defining role-based mechanisms.

The GT function monitor (GTFM) data model specifies a configuration method to generate a flow logic for monitoring purposes, e.g. observing data parameters with respect of a defined limit.

2 Normative references

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

ISO/IEC 8824-1, *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation — Part 1:*

ISO/IEC 8825-7, *Information technology — ASN.1 encoding rules — Part 7: Specification of Octet Encoding Rules (OER)*

ISO 14229-1, *Road vehicles — Unified diagnostic services (UDS) — Part 1: Application layer*

ISO/TS 17429, *Intelligent transport systems — Cooperative ITS — ITS station facilities for the transfer of information between ITS stations*

ISO/TS 21177, *Intelligent transport systems — ITS station security services for secure session establishment and authentication between trusted devices*

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

ISO 22900-2, *Road vehicles — Modular vehicle communication interface (MVIC) — Part 2: Diagnostic protocol data unit (D-PDU API)*

ISO 24102-6, *Intelligent transport systems — Communications access for land mobiles (CALM) — ITS station management — Part 6: Path and flow management*

CEN/TS 17496, *Cooperative intelligent transport systems — Communication profiles*

IEEE 1609.2, *IEEE Standard for Wireless Access in Vehicular Environments — Security Services for Applications and Management Messages*

RFC 5646, *Tags for Identifying Languages*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/TS 21177, CEN/TS 17496, ISO 21217 and the following apply.

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

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

3.1 **class** **<class>**

an extensible programme-code-template for creating objects, providing initial values for state (member variables) and implementations of behaviour (member functions or methods) in object-oriented programming

3.2 **convention** **Cvt**

indicates if specification is "M" (mandatory), "O" (optional), or "C" (conditional)

3.3 **global transport protocol client** **GTP client**

entity that instigates the provision of the GTP service

3.4 **global transport protocol server** **GTP server**

entity that provides the GTP service

3.5 **raw data**

data of a sensor control network (SCN) in the untreated format as specified by the SCN owner

3.6 **raw data source identifier**

unique identifier of an entity in an SCN ([3.9](#))

3.7 **retrofit**

addition of new technology or features to existing systems

3.8 **secure data interface** **SDI**

gateway providing bidirectional means for security and access control

3.9**sensor control network****SCN**

communication bus with measured inputs and controlled outputs

4 Abbreviated terms

ADU	application data unit
AMQP	advanced message queuing protocol
ASN.1	abstract syntax notation one
BP	basic principle
CAN	controller area network
cId	common identifier
C-ITS-SU	central ITS station unit
C-ITS-S	central ITS station
CUP	CAN utility protocol
Cvt	convention
dId	data identifier
D-PDU	diagnostic protocol data unit
dpId	data parameter identifier
DTC	diagnostic trouble code
eCall	emergency call
ECU	electronic control unit
ecuId	electronic control unit identifier
enh-diag	enhanced diagnostic
FSH	facilities service handler
GNSS	global navigation satellite system
GPIO	general purpose input/output
GT	global transport
GTAC	global transport access control
GTBasic	global transport basic
GTDF	global transport data format
GTDM	global transport data management
GTDF2SCNPDU	global transport data format to sensor control network protocol data unit

GTFM	global transport function monitor
GTP	global transport protocol
i	internal
IEEE	Institute of Electrical and Electronics Engineers
IMU	inertial measurement unit
IP	internet protocol
IPR	intellectual property rights
itid	info type identifier
ITS-AID	ITS application identifier
ITS-SU	ITS station unit
IVN	in-vehicle network
K-line	communication line
KWP2000	Keyword Protocol 2000
lid	local identifier
LDM	local dynamic map
LSb	least significant bit
LSB	least significant byte
MIL	malfunction indicator light
MQTT	message queuing telemetry transport
MVCI	modular vehicle communication interface
OBD	on-board diagnostics
odt	object description table
OTL	on-board diagnostic threshold limits
PDU	protocol data unit
pgn	parameter group number
pid	parameter identifier
ProtBuf	protocol buffers
P-ITS-S	personal ITS-station
P-ITS-SU	personal ITS-station unit
PSID	provider service identifier
r	read

rid	routine identifier
R-ITS-S	roadside ITS-station
R-ITS-SU	roadside ITS-station unit
SAP	service access point
SCNDF	sensor and control network data format
SCNPDU2GTDF	sensor and control network protocol data unit to global transport data format
snsid	sensor identifier
SNI	secure network interface
spn	suspect parameter number
SPaT	signal phase and timing
SSP	service specific permission
tid	test identifier
TLS	transport layer security
u	user optional
UDS	unified diagnostic services
UGP	unified gateway protocol
VIS	vehicle information service
V-ITS-S	vehicle ITS-station
V-ITS-SU	vehicle ITS-station unit
w	write
W3C	world wide web
WWH-OBDD	world wide harmonized on-board-diagnostic
x	execute
XCP	universal measurement and calibration protocol
xml	extensible markup language

5 Conventions

This document conforms to the OSI Service Conventions specified in ISO/IEC 10731.

6 Global Transport Data Management (GTDM) framework

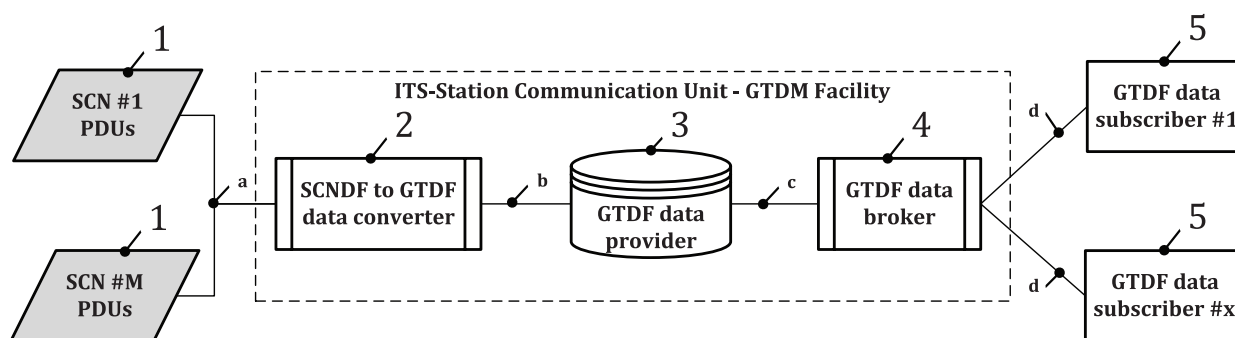
6.1 General

The data flow from sensor and control networks (e.g., in-vehicle networks, roadside station unit networks) to a data server, which functions as a data publisher, is illustrated in [Figure 1](#). The technical

solution specified in this document addresses the complexity of converting protocol data units (PDUs) with raw (device-specific) data of any sensor and control network, of any kind of technical equipment, as part of the ITS eco-system into a standardized data format, called global transport data format (GTDF).

The state-of-the-art conversion of sensor and control network raw data is an implementation of conversion routines hard-coded in software in a gateway/an electronic control unit (ECU). The gateway/ECU provides the PDUs upon request/response scheme to a client.

The global transport data framework (GTDM) fulfills the function of data conversion of sensor and control network raw data in a flexible way, using configuration data specific for the sensor and control network connected to. The advantage of the configuration concept is the flexibility to use the same implementation architecture for different sensor and control networks.



Key

- 1 sensor and control network #1 to #s protocol data units
- 2 sensor and control network PDUs converted into GTDF data
- 3 GTDF data provider containing all data parameter values specified in the configuration(s)
- 4 GTDF data broker performs event-based transfer of instant updates or historically measured data according to subscription(s)
- 5 clients (GTDF data subscribers) subscribe to data parameters based on the implementation of use cases
- a Sensor and control network protocol data units, containing raw signals in raw data format, forwarded to the SCNDF to GTDF data converter.
- b GTDF data parameter values forwarded to the GTDF data provider.
- c GTDF data provider provides data parameter values to the GTDF data broker for publishing to subscriber(#1 to #x).
- d The GTDF data broker publishes data parameter values based on subscription by clients.

Figure 1 — Data flow from SCN data format to global transport data format

[Figure 2](#) depicts the overall GTDM framework. Use cases are to be defined by stakeholders to identify the objectives, actors, input data and output data to achieve their objectives. Once the use cases are defined, the use case information is converted into requirement statements. Such requirements are implemented in use case-specific applications. The use case input and output information are converted into data parameters compatible to the GTDF specified in this document. GTDF defined data are based on data types specified in this document. One or more configurations are created based on use cases definitions. Applications can share their data in the same data format (GTDF). GTDF configurations are the data necessary to interpret data exchange between entities. The data flow is determined by the GTDF configurations.

NOTE Use cases and stakeholder task definitions are not within the scope of this document.

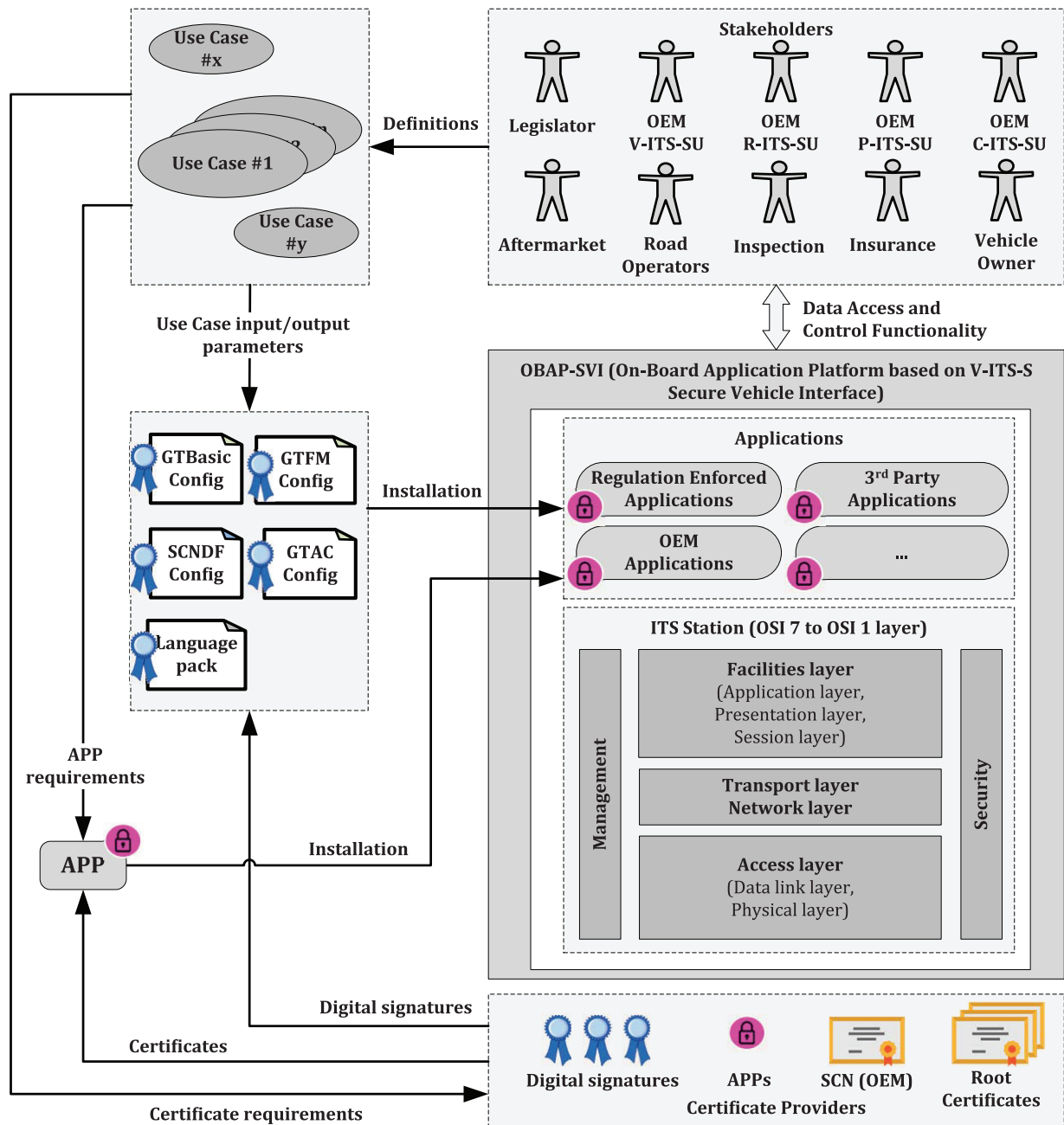


Figure 2 — GTDM framework

6.2 Applicable use case groups

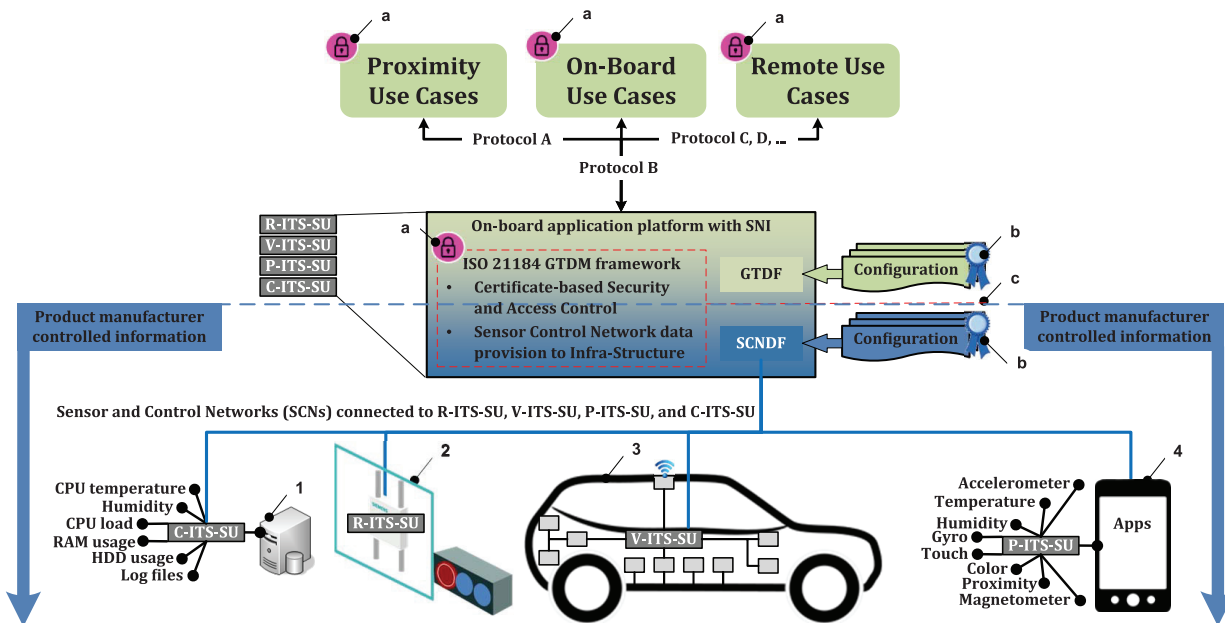
6.2.1 General

The use case groups (Proximity, On-Board, Remote) illustrated in [Figure 3](#) are candidates for diagnostics inspection and maintenance, on-board applications, remote ITS applications and remote ITS applications and neutral backend server(s) applications, which utilize the GTDM framework (this document) implemented on a secure network interface (SNI). The secure communication interface software implements at least the requirements stated in the following documents:

- ISO/TS 21177;
- ISO/TS 21184 (this document); and

— ISO/TS 21185.

SNI implementations utilize the GTDM framework in combination with ISO 22900-2 D-PDU, the diagnostic protocol data unit, to interface to the SCN



Key

- 1 central station unit (C-ITS-SU) e.g. traffic management system backoffice server specific sensor and control network
 - 2 roadside station unit (R-ITS-SU) specific sensor and control network
 - 3 on-board unit (V-ITS-SU) specific sensor and control network
 - 4 personal (smartphone) station unit (P-ITS-SU) specific sensor and control network
- a IEEE 1609.2 certificate with Provider Service Identifier (PSID) and Service-Specific Permission (SSP) identifier.
- b Digital signature for verifying the authenticity of digital configuration documents.
- c Integrated firewall based on GTDF and SCNDF configuration file information.

Figure 3 — Application use cases and sensor and control networks

R-ITS-SU, V-ITS-SU, P-ITS-SU and C-ITS-SU products, implemented in accordance with the documents specified in [Clause 2](#), are applicable for new designed products and retrofits.

The various use case-based applications illustrated are described in [6.2.4](#) to [6.2.6](#) and based on access to SCN examples. The same or similar use cases apply to R-ITS-SUs, for example.

6.2.2 GTDM framework operation

At startup, the GTDM framework software loads the GTDF configurations and optionally SCNDF configurations. Based on the content of the configurations, the software starts the communication with the SCN(s) and accesses raw information as specified in the SCNDF configurations. Once the protocol data units (PDUs) are available, the relevant information converted into the global transport data format (GTDF). Conversion information is contained in the SCNDF configuration. The timing of accessing raw data on the SCN(s) is defined in the scheduler as part of the SCNDF configuration. Various access methods (e.g. on event, periodic, request and response scheme) are available on how to access the data on the SCN(s). Each data item is assigned a raw data source identifier corresponding to its source function or physical unit, and a data parameter identifier. The combination of both identifiers makes it a unique identifier for a newly created or updated data item in the GTDM framework.

All data items (ECU data parameters) in GTDF are time stamped at the time of raw data reception. After conversion of the data from the raw data format to the standardized format, the information is forwarded to the GTDM Data Provider, which also supports a GTDM data broker function. The global transport protocol (GTP) services specified in this document are used to access and control data items. The GTP services provide a superset of functionality of the most common protocols (e.g. ISO 13185-2 UGP^[2], ISO/IEC 19464 AMQP^{[4][15]}, MQTT^[11], ProtBuf^[12], W3C VIS^[13]) used in the industry. A protocol service mapping and a data format converter needs to be installed for each protocol to interface with the GTP services.

6.2.3 GTDM certificate-based access control

The access control specified in this document is dedicated to the GTP services (see 8.4.3.2) and individual ECU data parameter depending on the content of the IEEE 1609.2 compatible certificates. The certificate contains a Provider Service Identifier (PSID) which is the equivalent of an ITS Application Identifier (ITS-AID), see 8.3.2. The certificate also includes a BitmapSsp (Bitmap Service-Specific Permission) structure, see 8.3.3. Figure 27 illustrates an example of a BitmapSsp including a role identifier. Examples of role are: predictive maintenance, read-only vehicle diagnostics, enhanced vehicle diagnostics and station software update. A prerequisite of a client application accessing data and controlling functions of a station as shown in Figure 3 (R-ITS-SU, V-ITS-SU, P-ITS-SU, C-ITS-SU) requires an ITS-AID (PSID) and a BitmapSsp with an appropriate role identifier, which is compatible with the ITS-AID.

This example describes a diagnostic application (ITS-AID) which is limited to "predictive maintenance" (BitmapSsp: role = predictive maintenance) and therefore is not authorized to clear diagnostic trouble code information.

Table 1 describes an example of access control applied to GTP services and EcuDataParams.

Table 1 — Example of access control applied to GTP services and EcuDataParams

GTP service	EcuDataParam	BitmapSsp (role id)		
		1 ^a	2 ^b	3 ^c
GetSupportedInfo(EcuDataParams)	provides a list of all supported combinations of ECU and data parameter: e.g., ECM.Calculated LOAD Value, ECM.Engine RPM, TCM.Vehicle Speed Sensor, ECM.Engine Idle Speed;	X ^d	X ^d	X ^d
GetSupportedInfo(DTCs)	provides a list of all supported DTCs by the SNI or a specific ECU connected to the SCN;	--- ^e	--- ^e	--- ^e
GetSupportedInfo(Messages)	provides a list of all supported messages by the SNI: e.g., eCall Minimum Set of Data;	--- ^e	--- ^e	--- ^e
SubscribeValues	ECM.Calculated LOAD Value	X ^d	X ^d	--- ^e
	ECM.Engine RPM	X ^d	X ^d	--- ^e
	TCM.Vehicle Speed Sensor	X ^d	X ^d	--- ^e
PublishValues	ECM.Calculated LOAD Value	X ^d	X ^d	--- ^e
	ECM.Engine RPM	X ^d	X ^d	--- ^e
	TCM.Vehicle Speed Sensor	X ^d	X ^d	--- ^e
SetValues	ECM.Engine Idle Speed	--- ^e	X ^d	--- ^e
ControlValues	ECM.Engine RPM	--- ^e	X ^d	--- ^e
SubscribeMessages	e.g. server application subscribes to eCall Minimum Set of Data message	--- ^e	--- ^e	--- ^e
^a Predictive maintenance application. ^b Enhanced vehicle diagnostics. ^c Station software update. ^d Cell selected/feature defined. ^e Empty cell/feature not selected.				

Table 1 (continued)

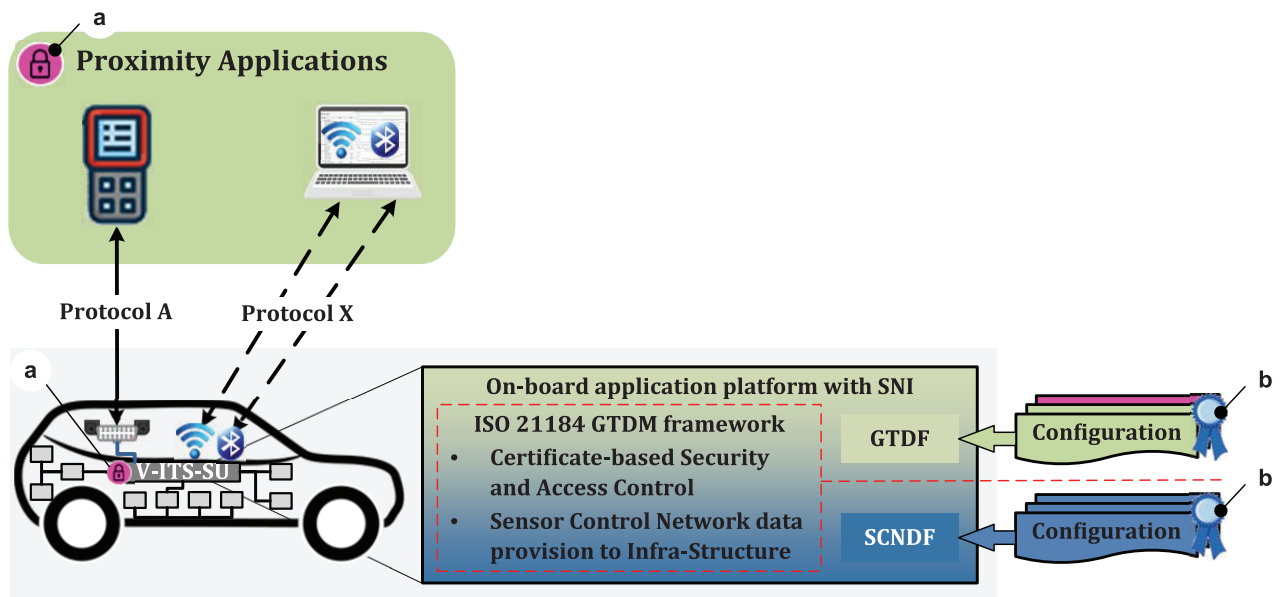
GTP service	EcuDataParam	BitmapSsp (role id)		
		1 ^a	2 ^b	3 ^c
PublishMessages	e.g. application platform publishes the eCall Minimum Set of Data message if eCall event occurs	--- ^e	--- ^e	--- ^e
SubscribeDtcInfo	e.g. client application subscribes to one, multiple or all DTCs	X ^d	X ^d	X ^d
PublishDtcInfo	e.g. SNI publishes DTCs	X ^d	X ^d	X ^d
ClearDtcInfo	e.g. clear all diagnostic information in a specific or all ECUs connected to the SCN	--- ^e	X ^d	X ^d
ListFiles	e.g. provides a list of all files accessible as per the role, which are stored in the ITS station unit or a specific element connected to the SCN	--- ^e	--- ^e	X ^d
ManageFiles(upload)	e.g. software update of SNI or a specific ECU connected to the SCN	--- ^e	--- ^e	X ^d
ManageFiles(download)	e.g. download event-based snapshot file	--- ^e	X ^d	X ^d
ManageFiles(delete)	e.g. delete all event-based snapshot files	--- ^e	X ^d	X ^d
Reset	e.g. software reset of OBAP-SVI or a specific ECU connected to the SCN	--- ^e	--- ^e	X ^d
^a	Predictive maintenance application.			
^b	Enhanced vehicle diagnostics.			
^c	Station software update.			
^d	Cell selected/feature defined.			
^e	Empty cell/feature not selected.			

6.2.4 Proximity use case applications

[Figure 4](#) illustrates secure nearby ITS-SU applications, which access a sensor and control network (SCN) with an SNI software implementation. Proximity use cases define the nearby access to data and control functions supported by the SCN(s) of, for example, the vehicle. The connection between devices/applications and the (for example) vehicle is established via the diagnostic link connector (DLC), WiFi or Bluetooth. Multiple different application layer protocols (see [6.2.1](#)) can be supported simultaneously.

Typical proximity applications are:

- diagnostic, repair and maintenance,
- electronic periodic technical inspection (ePTI), and
- roadside assistance.



Key

- a IEEE 1609.2 certificate with Provider Service Identifier (PSID) and Service-Specific Permission (SSP) identifier.
- b Digital signature for verifying the authenticity of digital configuration documents.

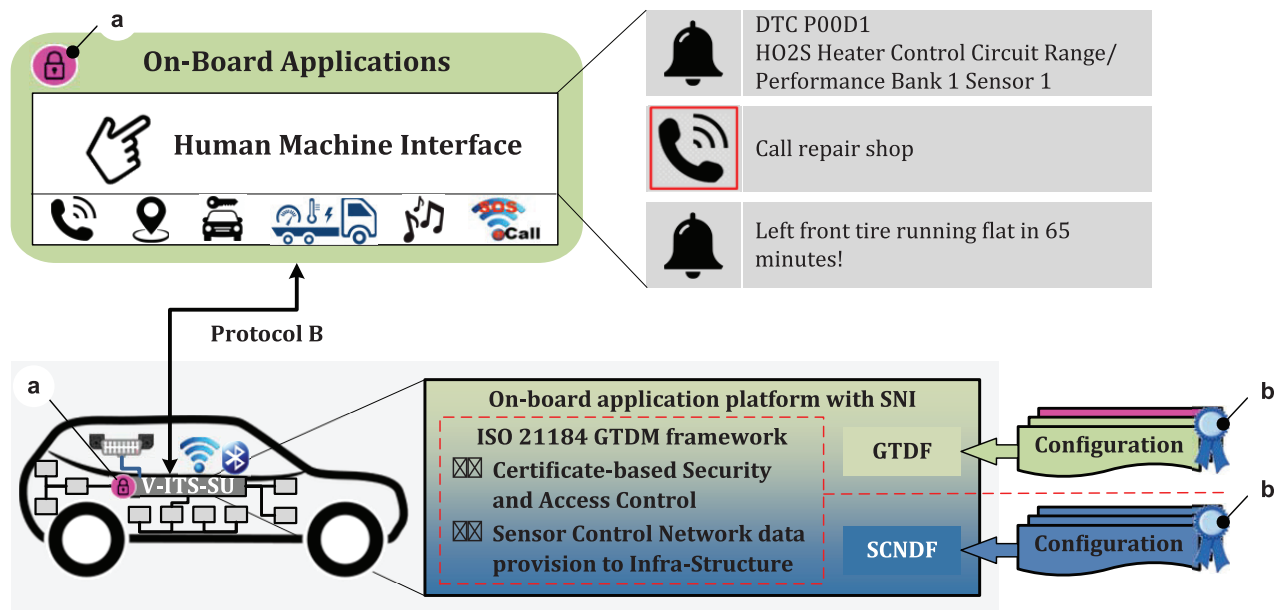
Figure 4 — Proximity applications

6.2.5 On-board use case applications

[Figure 5](#) illustrates secure ITS-SU on-board applications, which access a sensor and control network (SCN) with an SNI software implementation. On-board use cases define the vehicle's on-board access to data and control functions supported by the SCN(s). The connection between devices/applications and, for example, the vehicle's SCN(s) can be established via a hard-wired IP-based socket connection. Implementation of on-board applications are installed into the vehicle, for example, on-board infotainment system. Multiple different application layer protocols (see [6.2.1](#)) can be supported simultaneously.

Typical on-board applications are:

- vehicle health status,
- P-A-Y-D (Pay-As-You-Drive) data collection,
- insurance data collection,
- fleet management,
- rental car management,
- maintenance schedule, and
- any third party app data access.



Key

- a IEEE 1609.2 certificate with Provider Service Identifier (PSID) and Service-Specific Permission (SSP) identifier.
- b Digital signature for verifying the authenticity of digital configuration documents.

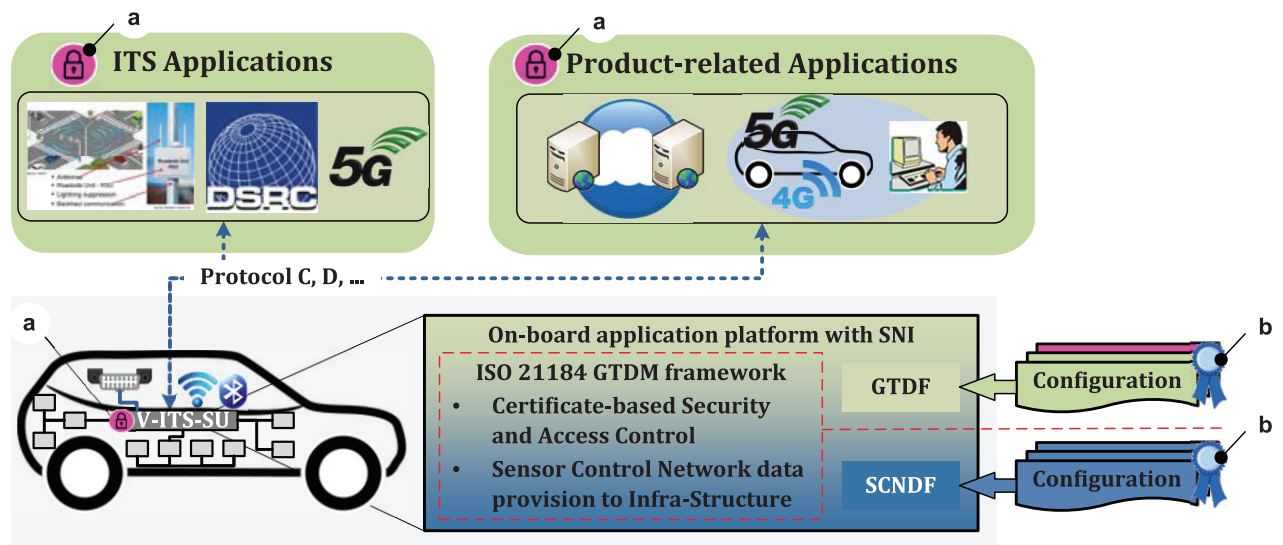
Figure 5 — On-board applications

6.2.6 Remote use case (ITS, product-related) applications

Figure 6 illustrates secure remote ITS-SU applications, which access a sensor and control network (SCN) with an SNI software implementation. Remote use cases define the vehicle's remote access to data and control functions supported by the SCN(s) and the data transfer to one or multiple neutral backend server(s). The connection between remote devices/applications and, for example, the vehicle's SCN(s) can be established via a wireless wide area network IP-based socket connection. Multiple different application layer protocols (see 6.2.1) can be supported simultaneously.

Typical remote applications are:

- remote diagnostic services,
- access provider (data plan, propose applications, promote services),
- cloud-based (neutral server) services,
- cooperative ITS services, and
- commercial services.



Key

- a IEEE 1609.2 certificate with Provider Service Identifier (PSID) and Service-Specific Permission (SSP) identifier.
- b Digital signature for verifying the authenticity of digital configuration documents.

Figure 6 — Remote (ITS, product-related) applications

6.2.7 Common configuration data authoring process

Figure 7 shows a common configuration data authoring process for worldwide standardized data and manufacturer product-specific data. The goal of this common process is to:

- achieve the capability of merging worldwide standardized data and cross-manufacturer product-specific data in cross-manufacturer-specific products (R-ITS-SUs, V-ITS-SUs, P-ITS-SUs, C-ITS-SUs);
- achieve the capability of data exchange between R-ITS-SUs, V-ITS-SUs, P-ITS-SUs and C-ITS-SUs and their connected applications;
- achieve the capability of secured communication between R-ITS-SUs, V-ITS-SUs, P-ITS-SUs and C-ITS-SUs;
- achieve the capability of access control to data and control functions according to ISO 21177, supported R-ITS-SUs, V-ITS-SUs, P-ITS-SUs and C-ITS-SUs by installing IEEE 1609.2 certificates, including PSID (ITS-AID) and SSP (role identifier);
- achieve protection of manufacturer product-specific intellectual property of their SCNs (no need to expose their raw data to the public);
- achieve harmonized integration of standards developed for Intelligent Transport Systems and Road Vehicles.

Key 1 to Key 4 represent the database servers in the Internet, where each fulfills a specific function in the definition and creation of worldwide, harmonized, and standardized use cases and data elements in GTDF.

Key 5 to Key 7 represent the database servers in the Internet, where each fulfills a specific function in the definition and creation of manufacturer-specific product-related SCN use cases and data elements in GTDF.

Key a to Key d are authors to create worldwide, harmonized, and standardized data elements in GTDF into the database servers in the Internet, which result in configuration packages including worldwide, harmonized, and standardized data elements for installation in R-ITS-SUs, V-ITS-SUs, P-ITS-SUs and C-ITS-SUs according to the data models specified in ISO 21184.

Key e to Key g are authors to create manufacturer-specific product-related SCN data elements in SCNDF into the database servers in the Internet, which result in configuration packages including manufacturer-specific product-related SCN data elements for installation in R-ITS-SUs, V-ITS-SUs, P-ITS-SUs and C-ITS-SUs according to the data models specified in ISO 21184.

Key i shows the information flow from authors of worldwide, harmonized, and standardized data elements and manufacturer authors of manufacturer-specific product-related SCN data elements into applicable Internet server and database(s).

Key j shows the information flow from the Central ITS data concept registry to create world-wide standardized and manufacturer-specific use cases;

Key k shows the information flow from use cases to authors of worldwide, harmonized, and standardized data elements and manufacturer authors of SCN data elements according to the data models specified in ISO 21184.

Key l shows the storage of worldwide, harmonized, and standardized ITS and Road Vehicles data elements in configuration files.

Key m shows the storage and distribution of worldwide, harmonized, and standardized configuration files and manufacturer-specific product-related configuration files.

Key n shows the data element exchange in GTDF between applications installed in manufacturer-specific products R-ITS-SUs, V-ITS-SUs, P-ITS-SUs and C-ITS-SUs.

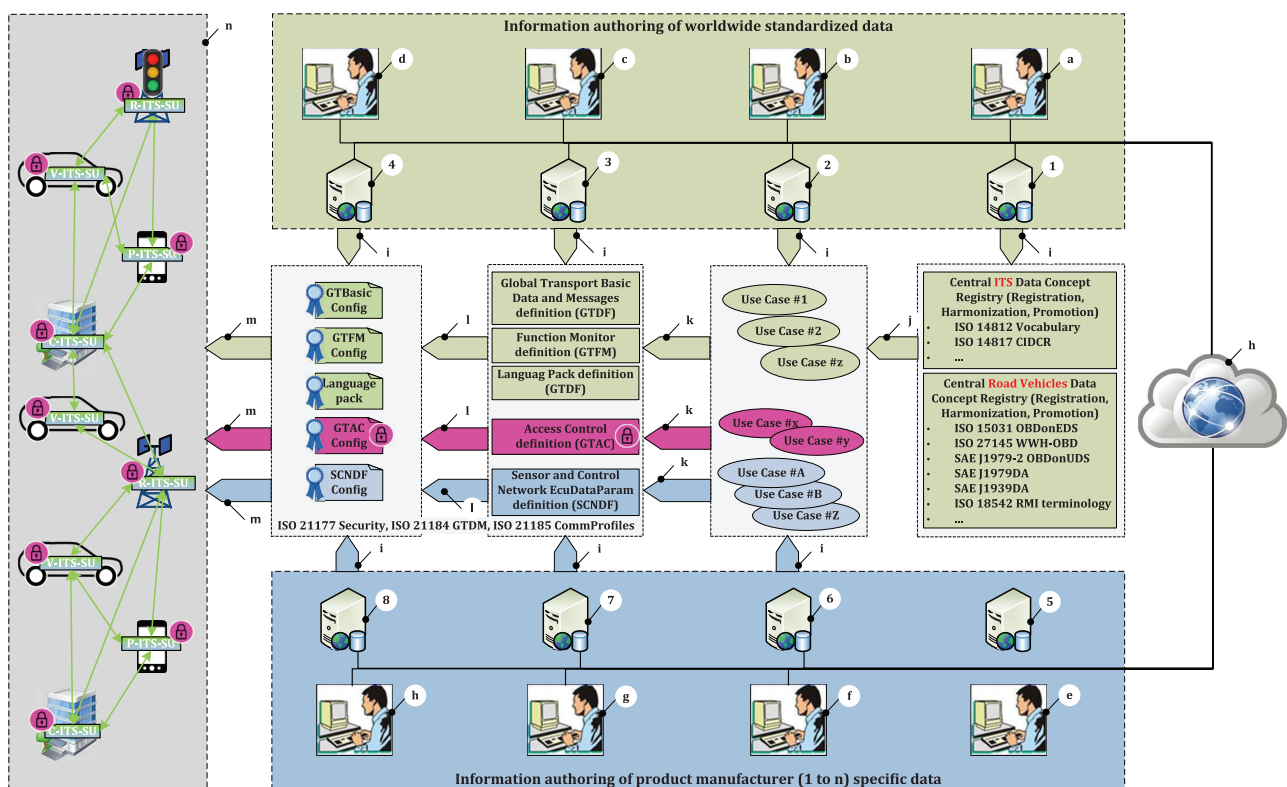


Figure 7 — Common configuration data authoring process

6.3 GTDM Basic principles

The requirements specified in this document are based on the following basic principles (BP) and assumptions:

- BP1: Raw sensor and control network (SCN) data are the intellectual property of the SCN owners.

- BP2: Data models of raw data may be kept confidential by SCN owners, for example to reduce vulnerability and manipulation of data in the SCN.
- BP3: To provide consistent interpretation across ITS applications, a consistent format known as the Global Transport Data Format (GTDF) supports third parties developing their own use cases and products to process or exchange such data.
- BP4: This document aims to support manufacturers of ITS-SUs for any installation context, e.g. R-ITS-S, V-ITS-S, P-ITS-S, and C-ITS-S.
- BP5: SCN device manufacturers create SCN configurations that are used to convert the raw SCN data into the GTDF. SCN configurations are not published.
- BP6: GTDF configurations are created by third party application developers to specify data exchange requirements to support their applications including data elements and performance requirements.
- BP7: To protect GTDF configurations these may be secured by means of encryption and authentication.
- BP8: Each application using a GTDF configuration is required to prove its permission to access and modify the data, for example, using IEEE 1609.2 certificates.
- BP9: Sensor quality information is required to be provided by the sensor and control network (SCN) owner the sensor is connected to. In this context, the sensor quality definition includes, but is not limited to:
 - accuracy (or error) in %: indicates how closely the measured sensor value is compared to the actual value;
 - precision: indicates how many consistent, repeatable measurements a sensor produces;
 - resolution: indicates the smallest degree of sensor value change that can be detected and recorded by the sensor;
 - sensitivity: an absolute quantity, indicates the smallest absolute amount of change that can be detected by a measurement;
 - drift: indicates susceptibility of sensor drift over time;
 - frequency of update: indicates the number of occurrences of a repeating event per unit of time.

7 ITS-S global transport data model

7.1 Overview

The ITS station facilities layer implements the following data models:

- GTBasic data model;
- GTAC data model;
- GTFM data model; and
- SCNDF data model.

7.2 GT basic data model

7.2.1 Overview

The GTBasic data model consists of:

- GT basic configuration data model;

- GT language package data model;
- GT runtime data model; and
- GT condition data model.

Figure 8 shows the legend for the class diagrams introduced in the following subclauses.

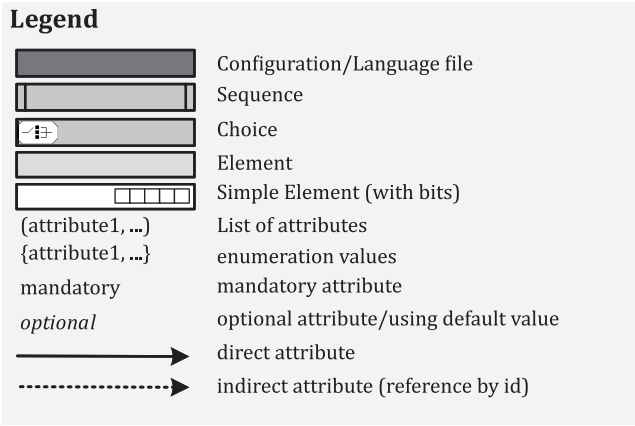


Figure 8 — Legend class diagram

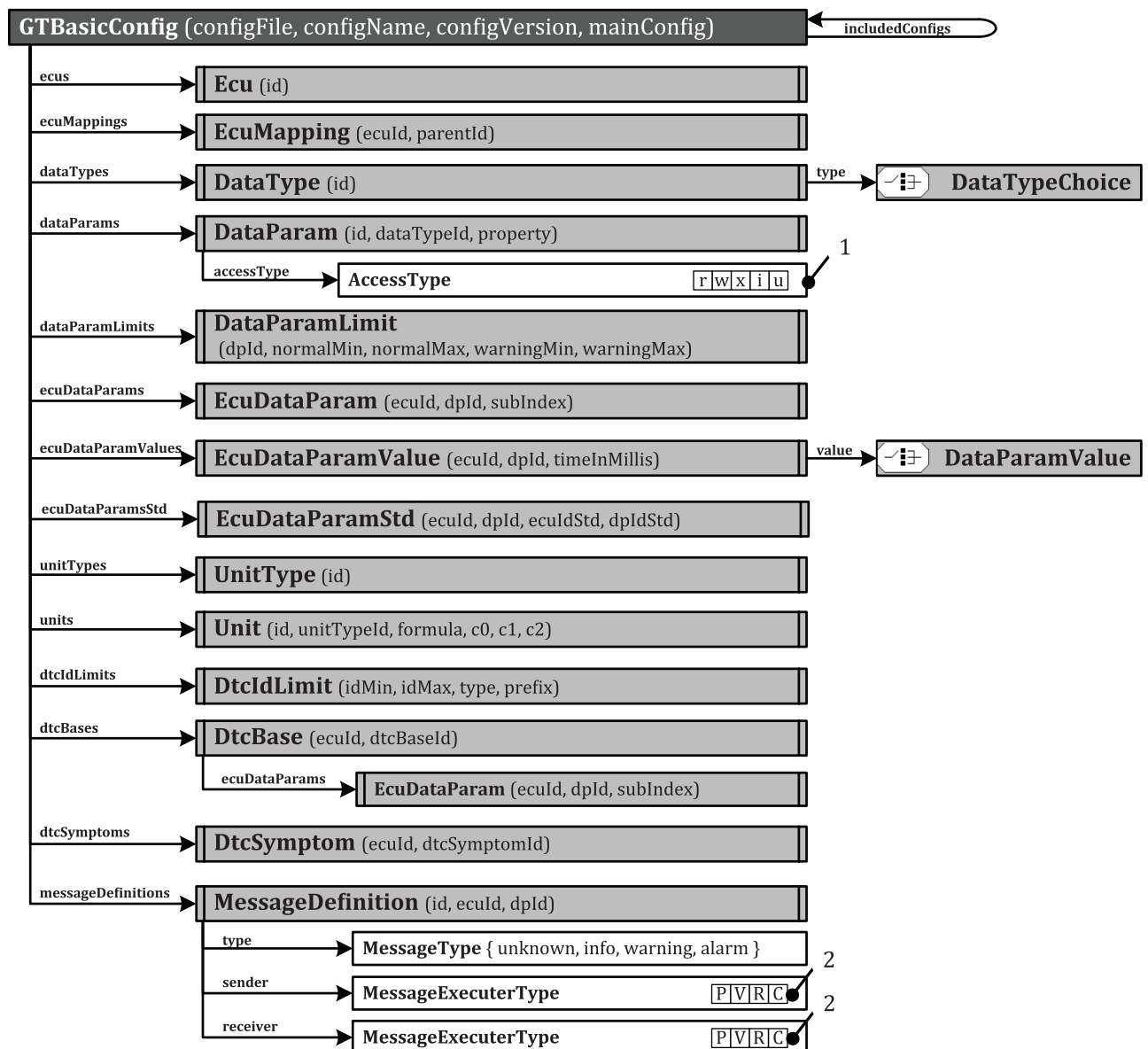
7.2.2 GT basic configuration

7.2.2.1 <class> GTBasicConfig

The <class> GTBasicConfig configuration file contains all information required by two or more communication entities in order to interpret messages and data received or to be transmitted by an entity. Such data is the filename of this configuration, name, version, the reference to its main configuration, list of names of included configurations, list of Sources, for example ECUs, computing nodes (see Figure 9), and LanguagePacks for any language provided. Once the GTBasicConfig file is made available in an ITS-SU, all information become savailable at runtime to the GTDM function of its facility.

REQ	6.1 GT basic configuration - <class> GTBasicConfig
The <class> GTBasicConfig shall be defined according to the ASN.1 definition specified in Table 2, with details specified in A.1.	

Figure 9 shows the <class> GTBasicConfig.

**Key**

- 1 AccessType: r = read, w = write, x = execute, i = internal, u = user optional (see Table 29)
- 2 P = P-ITS-SU, V = V-ITS-SU, R = R-ITS-SU, C = C-ITS-SU

Figure 9 — <class> GTBasicConfig

Table 2 specifies the <class> GTBasicConfig attributes.

Table 2 — <class> GTBasicConfig attributes

Name	Description	Cvt
configFile	File name of the configuration; shall fulfil the regular expression [a-z0-9-~@]+\.[a-z]+; e.g., passcar_dodge_charger.cfggtdfvm.	M
configName	Name of the configuration; e.g., Passenger car Dodge Charger.	M
configVersion	Version of the configuration; shall be defined to be comparable; e.g., contains date/time; e.g. 20190711Sv.	M

Table 2 (continued)

Name	Description	Cvt
parentConfigFile	The parent configuration file is referenced in this configuration file. A GTBasicConfig of a vehicle model shall reference its own configFile. A GTBasicConfig of a personalized vehicle (includes VIN, licence plate, etc.) references its vehicle model.	M
includedConfigs {}	A list of configuration names on which this configuration depends, e.g., pass-car_dodge_charger.cfggtdfvm depends on vehicle.cfggtdfg, and roadvehicles.cfggtdff, saej1979.cfggtdff.	M
ecus {}	List of ECU definitions (see 7.2.2.3).	M
ecuMappings {}	List of ECU mapping definitions (see 7.2.2.4).	M
dataTypes {}	List of data type definitions (see 7.2.2.5).	M
dataParams {}	List of data parameter definitions (see 7.2.2.6).	M
dataParamLimits {}	List of data parameter limit definitions (see 7.2.2.7).	M
ecuDataParams {}	List of ECU data parameter definitions (see 7.2.2.8).	M
ecuDataParamValues {}	List of preconfigured ECU data parameter values (see 7.2.2.9).	M
ecuDataParamsStd {}	List of mappings between manufacturer specific and standardized ECU data parameters (see 7.2.2.10).	M
unitTypes {}	List of unit type definitions (see 7.2.2.11).	M
units {}	List of unit definitions (see 7.2.2.12).	M
dtcIdLimits {}	List of DTC identifier limit definitions (see 7.2.2.13).	M
dtcBases {}	List of DTC base definitions (see 7.2.2.14).	M
dtcSymptoms {}	List of DTC symptom definitions (see 7.2.2.15).	M
messageDefinitions {}	List of message definitions (see 7.2.2.16).	M

EXAMPLE "GNSS" of a GTBasicConfig

```

configFile "gnss.cfggtdfg", configName "GNSS", configVersion "20190904RL",
parentConfigFile "",
ecus { { id '00000001'H } }, // Secure Data Interface
dataTypes {
  { id '00000001'H, type numeric4: {decimalPlaces 5, unitId '00000044'H
    , factor 1, quotient 8000000, addend 0, min -720000000, max 720000000} }, -- GNSS
latitude
  { id '00000002'H, type numeric4: {decimalPlaces 5, unitId '00000045'H
    , factor 1, quotient 8000000, addend 0, min -1440000000, max 1440000000} }, -- GNSS
longitude
  { id '00000003'H, type numeric4: {decimalPlaces 1, unitId '00000010'H
    , factor 1, quotient 10, addend -1000, min 0, max 16777215} }, -- GNSS altitude (1/10m
+1000)
  { id '00000004'H, type numeric4: {decimalPlaces 2, unitId '0000002B'H
    , factor 549, quotient 100000, addend 0, min 0, max 65535} }, -- Heading Direction
(0.00549°)
  { id '0000000A'H, type numeric2: { decimalPlaces 0, unitId '00000019'H
    , factor 1, quotient 100, addend 0 } }, -- Speed (0,01 kph)
  { id '00000024'H, type numeric4: {decimalPlaces 0, unitId '0000004C'H
    , factor 1, quotient 1, addend 0, min 0 } }, -- Time stamp in seconds since 1970
  { id '00000026'H, type numeric2: { decimalPlaces 3, unitId '00000010'H
    , factor 1, quotient 1000, addend 0 } }, -- GNSS accuracy (0,001 m)
  { id '00000035'H, type group: { paramsOut {
    { ecuId '00000001'H, dpId '00000013'H }, { ecuId '00000001'H, dpId '00000001'H },
    { ecuId '00000001'H, dpId '00000002'H }, { ecuId '00000001'H, dpId '00000003'H },
    { ecuId '00000001'H, dpId '00000014'H }, { ecuId '00000001'H, dpId '00000004'H },
    { ecuId '00000001'H, dpId '00000017'H } }
  } } -- GNSS
},
dataParams {
  { id '00000001'H, dataTypeId '00000001'H, accessType '10010'B
    , property sensor }, -- GNSS latitude
  { id '00000002'H, dataTypeId '00000002'H, accessType '10010'B
    , property sensor }, -- GNSS longitude

```

```

    { id '00000003'H, dataTypeId '00000003'H, accessType '10000'B
      , property sensor }, -- GNSS altitude
    { id '00000004'H, dataTypeId '00000004'H, accessType '10010'B
      , property ecu-internal-monitor }, -- Heading Direction
    { id '00000013'H, dataTypeId '00000024'H, accessType '10000'B
      , property sensor }, -- GNSS timestamp
    { id '00000014'H, dataTypeId '0000000A'H, accessType '10000'B
      , property sensor }, -- GNSS speed
    { id '00000017'H, dataTypeId '00000026'H, accessType '10000'B
      , property sensor }, -- GNSS accuracy
    { id '000000B3'H, dataTypeId '00000035'H, accessType '10000'B
      , property group,retrieve-dynamic,sdisensor-specific } -- GNSS
  },
  ecuDataParams {
    { ecuId '00000001'H, dpId '00000001'H }, { ecuId '00000001'H, dpId '00000002'H },
    { ecuId '00000001'H, dpId '00000003'H }, { ecuId '00000001'H, dpId '00000004'H },
    { ecuId '00000001'H, dpId '00000013'H }, { ecuId '00000001'H, dpId '00000014'H },
    { ecuId '00000001'H, dpId '00000016'H }, { ecuId '00000001'H, dpId '00000017'H },
    { ecuId '00000001'H, dpId '00000018'H }, { ecuId '00000001'H, dpId '000000B3'H }
  }

```

All referenced units can be included into one GTBasicConfig <class>, "Units".

The GTBasicConfig example "SDI" includes only message definitions. A GTBasicConfig <class> can include a single configuration information, e.g., update of a message or a data parameter.

EXAMPLE GTBasicConfig "SDI"

```

configFile "sdi", configName "SDI", configVersion "20190417TD",
messageDefinitions {
  { id '00000007'H, type unknown, sender '1110'B, receiver '0001'B,
    ecuId '00000001'H, dpId '000000C2'H }, -- SDI data
  { id '00000004'H, type info, sender '1110'B, receiver '0001'B,
    ecuId '00000001'H, dpId '000000BF'H }, -- SDI off
  { id '00000008'H, type warning, sender '1110'B, receiver '0001'B,
    ecuId '00000001'H, dpId '000000D7'H }, -- SDI file storage
  { id '00000009'H, type warning, sender '1110'B, receiver '0001'B,
    ecuId '00000001'H, dpId '000000D9'H }, -- Connection lost
  { id '0000000B'H, type warning, sender '1110'B, receiver '0001'B,
    ecuId '00000001'H, dpId '000000DC'H }, -- Operating range warning
  { id '0000000A'H, type alarm, sender '1110'B, receiver '0001'B,
    ecuId '00000001'H, dpId '000000DC'H }, -- Operating range failure
  { id '0000000C'H, type info, sender '1110'B, receiver '0001'B,
    ecuId '00000001'H, dpId '000000DC'H }, -- Operating range normal
  { id '00000006'H, type warning, sender '1110'B, receiver '0001'B,
    ecuId '00000001'H, dpId '000000C1'H } -- Maintenance mode
}

```

7.2.2.2 Identifier

The identifier number consists of the high byte followed by three lower bytes. The identifier identifies an ECU (see 7.2.2.3), a data type (see 7.2.2.5), a data parameter (see 7.2.2.6), a unit type (see 7.2.2.11), a unit (see 7.2.2.12), a DTC base (see 7.2.2.14), a DTC symptom (see 7.2.2.15), a message definition (see 7.2.2.16), etc.

REQ	6.2 GT basic configuration – Identifier
An identifier shall be specified as a 4-byte value divided into the source and the number of the identifier. The source of the information is identified by the high byte of the 4-byte identifier with the following predefined values:	
a	See Reference [7].
b	See Reference [8].
c	See Reference [9].

- 00₁₆ = worldwide,
- 01₁₆ = Manufacturer 1,
- 02₁₆ = Manufacturer 2,
- ...,
- FB₁₆ = Manufacturer 251,
- FC₁₆ = ISO ITS,
- FD₁₆ = ISO Road Vehicles,
- FE₁₆ = SAE J1939^a, and
- FF₁₆ = SAE J1979^b, SAE J2012^c.

The identifier shall be globally unique inside its type (e.g. unit).

^a See Reference [7].

^b See Reference [8].

^c See Reference [9].

7.2.2.3 <class> Ecu

The <class> Ecu defines an element or a function of the sensor and control network associated to the ITS-SU. ECUs can be, but are not limited to ECUs, functions, (computing) nodes in a vehicle and any identifiable elements or functions of the road infrastructure, etc.

REQ	6.3 GT basic configuration – <class> Ecu
The <class> Ecu shall be implemented according to the ASN.1 definition specified in Table 3 , with details specified in A.1 .	

NOTE A possible implementation of an ECU is an ITS-SCU (ITS station communication unit) according to the ITS station architecture specified in ISO 21217.

[Table 3](#) specifies the <class> Ecu attributes.

Table 3 — <class> Ecu attributes

Name	Description	Cvt
id ^a	Identifier ^b (see 7.2.2.2) of the ECU.	M
^a yyyy yyyy.		
^b xxxx xxxx.		

EXAMPLE Ecu

```
{ id 'FF000001'H }, -- emissions on-board diagnostic system
{ id 'FF000002'H }, -- OBD II-1
{ id 'FF000003'H }  -- OBD II-2
```

7.2.2.4 <class> EcuMapping

The <class> EcuMapping specifies the parent mappings between Ecus. A typical example is an emissions-related system in the vehicle. The system is called "OBD II" (virtual parent ECU) and is comprised of several physical child ECUs, e.g. engine control module, transmission control module. Data parameters of the child ECUs can be accessed via the parent ECU.

REQ	6.4 GT basic configuration – <class> EcuMapping
The <class> EcuMapping shall be implemented according to the ASN.1 definition specified in Table 4 , with details specified in A.1 .	

Table 4 specifies the <class> EcuMapping attributes.

Table 4 — <class> EcuMapping attributes

Name	Description	Cvt
ecuId	Identifier (see 7.2.2.2) of the ECU.	M
parentId	Identifier (see 7.2.2.2) of a parent ECU.	M

EXAMPLE EcuMapping

```
Example
{ ecuId 'FF000002'H, parentId 'FF000001'H }, -- OBD II-1 -> emissions on-board diagnostic
system
{ ecuId 'FF000003'H, parentId 'FF000001'H } -- OBD II-2 -> emissions on-board diagnostic
system
```

7.2.2.5 <class> DataType

7.2.2.5.1 <class> DataType attributes

The <class> DataType specifies the data type of a DataParam.

REQ	6.5 GT basic configuration - <class> DataType
The <class> DataType shall be implemented according to the ASN.1 definition specified in Table 5, which specifies the DataType attributes, with details specified in A.1.	

Figure 10 specifies the <class> DataType.

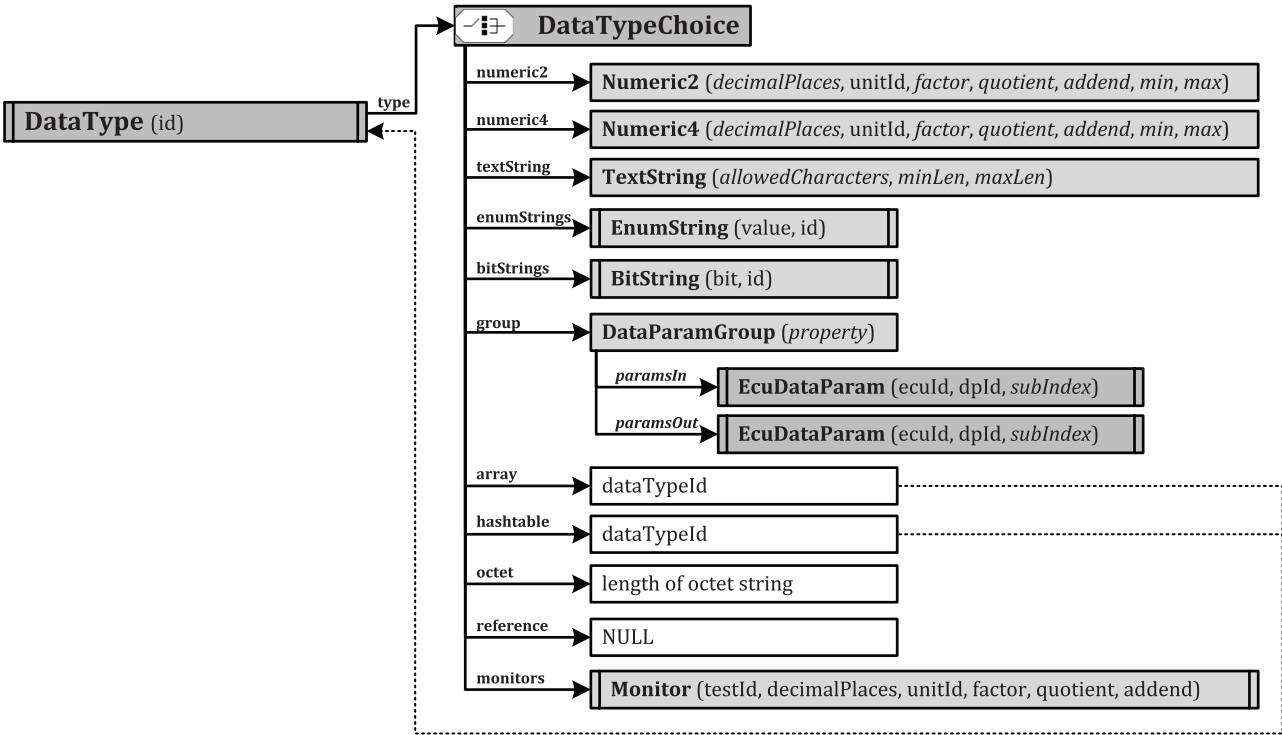


Figure 10 — <class> DataType

Table 5 specifies the DataType attributes.

Table 5 — DataType attributes

Name	Description	Cvt
id	Identifier (see 7.2.2.2) of the data type.	M
type	Type choice of the following sub types:	M
numeric2	— Defines a 2-byte numeric value (see 7.2.2.5.2).	C0 ^a
numeric4	— Defines a 4-byte numeric value (see 7.2.2.5.3).	C1 ^a
textString	— Defines a text string (see 7.2.2.5.4).	C2 ^a
enumStrings {}	— Defines a list of enumeration strings (see 7.2.2.5.5).	C3 ^a
bitStrings {}	— Defines a list of bit strings (see 7.2.2.5.6).	C4 ^a
group	— Defines a data parameter group (see 7.2.2.5.7).	C5 ^a
array	— Defines an array of elements (see 7.2.2.5.8); only the identifier of the contained DataType shall be specified.	C6 ^a
hashtable	— Defines a hashtable of elements (see 7.2.2.5.9); only the identifier of the contained DataType shall be specified.	C7 ^a
octet	— Defines an octet string (see 7.2.2.5.10); only the length of the octet string shall be specified (or 0 if the octet has variable length).	C8 ^a
reference	— Defines a reference to a EcuDataParam or DtcInfo (see 7.2.2.5.11).	C9 ^a
monitors {}	— A list of monitors (see 7.2.2.5.12).	C10 ^a
^a C0, C1, C2, C3, C4, C5, C6, C7, C8, C9 and C10 are the choices of the dataTypes' type. Only one of C0, C1, C2, C3, C4, C5, C6, C7, C8, C9 or C10 shall be used.		

The following subclauses list the different DataType types.

7.2.2.5.2 numeric2

A numeric2 data type is used to support 2-byte integer and floating-point values.

REQ	6.6 GT basic configuration – <class> DataType – numeric2
	The DataTypeChoice numeric2 shall be used to realize a 2-byte integer and floating-point value. A numeric2 value shall be defined with the <class> Numeric2 (see Table 6). To minimize the data parameter value length, all floating-point values shall be defined as SNUM16 (signed numeric 2-byte value). The transformation from a value of SNUM16 data type into a number with decimal places shall be implemented as specified in Formula (1).

The attributes 'factor' (f), 'quotient' (q), and 'addend' (a) are used to calculate the values v_{real} , v_{minreal} , and v_{maxreal} . The real value shall be rounded to the given decimalPlaces. The shortName of the unit shall be displayed after the value.

$$v_{\text{real}} = \frac{v \times f}{q} + a \quad (1)$$

Table 6 — <class> Numeric2

Name	Description	Cvt
decimalPlaces	Number of decimal places for the displayed value.	0
unitId	Unit identifier (see 7.2.2.2); references to the unit used (see 7.2.2.12).	M
factor	Factor with which to multiply; default value is 1.	0
quotient	Quotient with which to divide; default value is 1.	0
addend	Addend to which to add; default value is 0.	0
min	Minimal numeric value.	0
max	Maximal numeric value.	0

The example defines a numeric2 object representing a percentage of 1/10 %. The DataParamValue numeric2: 251 is read as "25,1 %".

EXAMPLE numeric2

```
dataTypes {
  { id 'FF000002'H, type numeric2: {
    decimalPlaces 1, unitId '0000001A'H, factor 1, quotient 10, addend 0 } } },
units {
  { id '0000001A'H, unitTypeId '00000002'H, formula 0, c0 1, c1 0, c2 0 } },
displayNames {
  { idType datatype, id 'FF000002'H, longName "Percentage (1/10 %)" },
  { idType unit, id '0000001A'H, shortName "%", longName "Percent" } }
```

NOTE All examples use the English language pack.

7.2.2.5.3 numeric4

A numeric4 data type is used to support 4-byte integer and floating-point values.

REQ	6.7 GT basic configuration - <class> DataType - numeric4
	The DataTypeChoice numeric4 shall be used to realize a 4-byte integer and floating-point value. A numeric4 value shall be defined with the <class> Numeric4, which shall have the same attributes as Numeric2 but different data types for the attributes (see A.1). To minimize the data parameter value length, all floating-point values shall be defined as SNUM32 (signed numeric 4-byte value). The transformation from a value of SNUM32 data type into a number with decimal places shall be implemented as specified in Formula (1) .

The attributes factor (default = 1), quotient (default = 1) and addend (default = 0) are used to calculate the values v_{real} , v_{minreal} , and v_{maxreal} . The real value shall be rounded to the given decimal places. The shortName of the unit shall be displayed after the value.

The numeric4 example describes a voltage that is specified with a resolution of 0,001 V. Three decimal places are displayed. The DataParamValue numeric4: 3 500 is interpreted as "3,5 V".

EXAMPLE numeric4

```
dataTypes {
  { id '00000011'H, type numeric4: {
    decimalPlaces 3, unitId '00000012'H, factor 1, quotient 1000, addend 0 } } },
units {
  { id '00000012'H, unitTypeId '0000000B'H, formula 0, c0 1, c1 0, c2 0 } },
displayNames {
  { idType datatype, id '00000011'H, longName "Voltage (0,001V)" },
  { idType unit, id '00000012'H, shortName "V", longName "Volt" } }
```

7.2.2.5.4 textString

A textString data type is used to support limited and unlimited strings.

REQ	6.8 GT basic configuration - <class> DataType - textString
	The DataTypeChoice textString shall be specified by the class TextString (see Table 7) and realizes limited and unlimited strings.

The optional attribute allowedCharacters contains a regular expression of the allowed characters in the string. The optional attributes minLen and maxLen define the range of the character count.

Table 7 — <class> TextString with examples

Name	Description	Cvt
allowedCharacters	A list (regular expression) of the allowed characters	0
minLen	Minimal length of the string	0

Table 7 (continued)

Name	Description	Cvt
maxLen	Maximal length of the string	0

EXAMPLE textString "Vehicle Identification Number"

```
dataTypes {
  { id 'FF000015'H, type textString: {
    allowedCharacters "A..HJ..NPR..Z0..9", minLen 17, maxLen 17 } } },
displayNames {
  { idType datatype, id 'FF000015'H, longName "Vehicle Identification Number" } }
```

The first example describes a 'Vehicle Identification Number' textString with exactly 17 characters (minLen = maxLen = 17) and may contain only capital letters from A - H, J - N, P, R - Z or the digits 0 - 9.

EXAMPLE textString "Unlimited string"

```
dataTypes {
  { id '00000014'H, type textString: {} } },
displayNames {
  { idType datatype, id '00000014'H, longName "Unlimited string" } }
```

The second example describes an unlimited textString. The DataParamValue textString: "VHJGH11763B65I860" can be identified as a Visible String for both examples.

7.2.2.5.5 enumStrings

The enumStrings data type is used to support enumerations of values in text form.

REQ	6.9 GT basic configuration - <class> DataType - enumStrings
	The DataTypeChoice enumStrings shall be specified as a SEQUENCE OF EnumString (see Table 8) and realizes enumerations of values in text form.

Only one of the enumeration strings can be selected.

Table 8 — <class> EnumString attributes

Name	Description	Cvt
value	Value of the enumeration string (sequential started by 0).	M
id	Identifier (see 7.2.2.2) of the enumeration string.	M

The example describes an enumeration of two text strings 'no' and 'yes'. If the enumeration value is 0, 'no' is selected, if the enumeration value is 1, 'yes' is selected.

EXAMPLE enumStrings

```
dataTypes {
  { id '00000012'H, type enumStrings: {
    { value 0, id '0000001C'H },
    { value 1, id '0000001D'H }
  } } },
displayNames {
  { idType datatype, id '00000012'H, longName "Answer (no, yes)" },
  { idType enumstring, id '0000001C'H, longName "no" },
  { idType enumstring, id '0000001D'H, longName "yes" } }
```

7.2.2.5.6 bitStrings

A bitStrings data type is used to support combinations of enumerations of values in text form.

REQ	6.10 GT basic configuration – <class> DataType – bitStrings
	The DataTypeChoice bitStrings shall be specified as a SEQUENCE OF BitString (see Table 9) and realizes combinations of enumerations of values in text form.

Any of the bit strings can be selected.

Table 9 — <class> BitString attributes

Name	Description	Cvt
bit	Value of the bit string (starts at 0)	M
id	Identifier (see 7.2.2.2) of the bit string name	M

The example describes a vehicle propulsion matrix with bits for Gasoline, Diesel, Compressed Natural Gas, Liquid Propane Gas, Electric Energy Storage, and Hydrogen. Any of the vehicle propulsion bits can be selected.

EXAMPLE bitStrings

```
dataTypes {
  { id '00000023'H, type bitString: {
    { bit 0, id '0000000A'H }, { bit 1, id '0000000B'H }, { bit 2, id '0000000C'H },
    { bit 3, id '0000000D'H }, { bit 4, id '0000000E'H }, { bit 5, id '0000000F'H }
  } },
  displayName {
    { idType datatype, id '00000023'H, longName "Vehicle propulsion " },
    { idType bitstring, id '0000000A'H, shortName "Gasol", longName "Gasoline" },
    { idType bitstring, id '0000000B'H, shortName "Diesel", longName "Diesel" },
    { idType bitstring, id '0000000C'H, shortName "CNG", longName "Compressed Natural Gas" },
    { idType bitstring, id '0000000D'H, shortName "LPG", longName "Liquid Propane Gas" },
    { idType bitstring, id '0000000E'H, shortName "EESstor", longName "Electric Energy Storage" },
    { idType bitstring, id '0000000F'H, shortName "Hydro", longName "Hydrogen" } }
}
```

The general $v_{\text{bitString}}$ is calculated by the Formula (2).

$$v_{\text{bitString}} = \sum_{\text{bit}=0}^{\text{length}-1} a_{\text{bit}} 2^{\text{bit}} \quad (2)$$

where

a_{bit} is the value of the bit position. a can take on the binary values 0 or 1.

The value for 'Gasoline and Electric Energy Storage' is calculated by $v_{\text{GE}} = 20 + 24 = 17$. The bit string value is coded as $v_{\text{bitString}}: 17$.

7.2.2.5.7 group

A group data type is used to support the summation of data parameters as input and output parameters or group by property.

REQ	6.11 GT basic configuration – <class> DataType – group
	The DataTypeChoice group shall be specified as DataParamGroup (see Table 10) and realized summation of data parameters shall be specified as input and output parameters or group by property. For the input and output parameters, the <class> EcuDataParam (see 7.2.2.8) shall be used.

Table 10 — <class> DataParamGroup

Name	Description	Cvt
paramsIn	Input parameter as SEQUENCE OF EcuDataParam (see 7.2.2.8).	C ₁ ^a
paramsOut	Output parameter as SEQUENCE OF EcuDataParam (see 7.2.2.8).	C ₂ ^a
property	Define the group by the property.	C ₃ ^a
^a A DataParamGroup can be defined as param or property specific. The following combinations are allowed: C ₁ , C ₁ and C ₂ , C ₂ or C ₃ . The most common is C ₂ .		

The example parameter list `group "GNSS position"` describes a parameter list with the output parameters "GNSS latitude", "GNSS longitude", and "GNSS altitude". A sub-parameter can also be a group. Recursions are not allowed, i. e., a parameter may not have itself as direct or indirect child.

EXAMPLE `group GNSS position`

```
dataTypes {
  { id 'FC000002'H, type group: { paramsOut {
    { ecuId 'FC000001'H, dpId '00000001'H },
    { ecuId 'FC000001'H, dpId '00000002'H },
    { ecuId 'FC000001'H, dpId '00000003'H }
  } } } },
dataParams {
  { id 'FC000001'H, dataTypeId 'FC000001'H, accessType '11000'B, property sensor },
  { id 'FC000002'H, dataTypeId 'FC000002'H, accessType '10000'B, property sensor },
  { id 'FC000003'H, dataTypeId 'FC000003'H, accessType '10000'B, property sensor } },
displayNames {
  { idType datatype, id 'FC000002'H, longname "GNSS position" },
  { idType dataparam, id '00000001'H, longname "GNSS latitude" },
  { idType dataparam, id '00000002'H, longname "GNSS longitude" },
  { idType dataparam, id '00000003'H, longname "GNSS altitude" } }
```

The example group "SDI sensor-specific data" describes the data parameter property "sdisensor-specific". During runtime, all parameter groups which have the property "sdisensor-specific" are collected into this group.

EXAMPLE `group SDI sensor-specific data`

```
dataTypes {
  { id '0000003F'H, type group: { {}, property sdisensor-specific } } },
displayNames {
  { idType datatype, id '0000003F'H, longName "SDI sensor specific data" } }
```

7.2.2.5.8 array

An array data type is used to support arrays of data types.

REQ	6.12 GT basic configuration – <class> DataType – array
The DataTypeChoice array shall be specified as Identifier (see 7.2.2.2) and realized via arrays of data types. Recursions shall not be allowed, i.e. a data type shall not have itself as direct or indirect child.	

The identifier references the underlying DataType. The contained data type can be of any type.

The example describes the data type with identifier `FC00 000A16` as an array of the data type `FC00 000216`. A DataParamValue array: 2 defines an array with two elements.

EXAMPLE `array`

```
dataTypes {
  { id 'FC00000A'H, type array: 'FC000002'H },
  { id 'FC000002'H, type group: {
    { { ecuId 'FC000001'H, dpId '00000001'H }, { ecuId 'FC000001'H, dpId '00000002'H } } } },
displayNames {
  { idType datatype, id 'FC00000A'H, longName "waypoints as GNSS position" } }
```

7.2.2.5.9 hashtable

A hashtable data type is used to support hashtables of data types.

REQ	6.13 GT basic configuration – <class> DataType – hashtable
The DataTypeChoice hashtable shall be specified as Identifier (see 7.2.2.2) and realized hashtables of data types.	

The identifier references the underlying dataType. Hash tables are not sorted and have no continuous ascending order numbering, but are accessed like arrays.

7.2.2.5.10 octet

An octet data type is used to support binary data as octet strings.

REQ	6.14 GT basic configuration – <class> DataType – octet
The DataTypeChoice octet shall be specified as SNUM32 and realizes binary data as octet strings.	

The SNUM32 value specifies the length of the octet string or 0 if the octet is of variable length.

The octet example describes an octet of variable size and a data type of octets with exactly 8 bytes.

EXAMPLE octet

```
dataTypes {
  { id 'FC00F301'H, type octet: 0 },
  { id 'FC00F302'H, type octet: 8 } ,
displayNames {
  { idType datatype, id 'FC00F301'H, longName "octetByte of variable size" },
  { idType datatype, id 'FC00F302'H, longName "octetByte with 8 bytes " } }
```

7.2.2.5.11 reference

A reference data type is used to support references to EcuDataParams and DtcInfo elements.

REQ	6.15 GT basic configuration – <class> DataType – reference
The DataTypeChoice reference shall realize references to EcuDataParams (see 7.2.2.8) and DtcInfo (see 7.2.4.4). No additional value shall be defined.	

The example "reference" describes a data type as reference.

EXAMPLE reference

```
dataTypes {
  {id '00000008'H, type reference: NULL } },
displayNames {
  { idType datatype, id '00000008'H, shortName "DPRef", longName "Parameter reference" } }
```

7.2.2.5.12 monitors

The data type monitor is used to support monitors as specified in SAE J1979.

REQ	6.16 GT basic configuration – <class> DataType – monitors
The DataTypeChoice monitors shall be specified as <class> monitor. Table 11 specifies the <class> monitor attributes.	

Table 11 — <class> Monitor attributes

Name	Description	Cvt
testId	Test identifier (see 7.2.2.2) of the monitor as UNUM16.	M
decimalPlaces	Number of decimal places for the displayed value.	0

Table 11 (continued)

Name	Description	Cvt
unitId	Unit identifier (see 7.2.2.2). References to the used unit (see 7.2.2.12).	M
factor	Factor to multiply with; default value is 1.	0
quotient	Quotient to divide with; default value is 1.	0
addend	Addend to add to; default value is 0.	0

The example describes a monitor with two tests. Each test has at least a testId and a unitId.

EXAMPLE monitor

```
dataTypes {
  { dataTypeId '0000FFFF'H, type monitors: {
    { testId 1, decimalPlaces 3, unitId '00000012'H, factor 1, quotient 1000, addend 0 },
    { testId 5, decimalPlaces 3, unitId '00000012'H, factor 1, quotient 1000, addend 0 }
  } } },
units {
  { id '00000012'H, unitTypeId '0000000B'H, formula 0, c0 1, c1 0, c2 0 },
  { id '00000020'H, unitTypeId '00000006'H, formula 3, c0 3600, c1 0, c2 0 } },
displayNames {
  { idType datatype, id '0000FFFF'H, longName " oxygen sensor monitor " },
  { idType unit, id '00000012'H, shortname "V", longname "Volt" },
  { idType unit, id '00000020'H, shortname "s", longname "Second(s)" } }
```

7.2.2.6 <class> DataParam

The <class> DataParam specifies a data parameter (signal) or a group (see 7.2.2.5.7) of data parameters from any ECU. The data parameter is not yet assigned to an ECU. The fusion of ECU and data parameters shall be done in <class> EcuDataParam (see 7.2.2.8).

REQ	6.17 GT basic configuration – <class> DataParam
The <class> DataParam shall be implemented according to the ASN.1 definition specified in Table 12, with details specified in A.1.	

Table 12 specifies the <class> DataParam attributes.

Table 12 — <class> DataParam attributes

Name	Description	Cvt
id	Identifier (see 7.2.2.2) of the data parameter.	M
dataTypeId	Reference to the data type identifier (see 7.2.2.5.1, 7.2.2.2).	M
accessType	The access type of the data parameter is a bit string with a combination of the following values: r (0) = read, w (1) = write, x (2) = execute, i (3) internal, u (4) = user entry	M
property	The data parameter property is a SNUM32 value with free definable values	M

EXAMPLE DataParam "Acceleration X (longitudinal)"

```
{ id '00000005'H, dataTypeId '00000005'H, accessType '10000'B, property sensor }
```

7.2.2.7 <class> DataParamLimit

The <class> DataParamLimit specifies limits for a data parameter. There are three lifetime areas in which a data parameter value can exist: normal, warning, and error. The normal lifetime area is the area where the data parameter provides values within normal operating range. The warning lifetime

area is the area where the data parameter provides values within warning operating range. All values not included in the normal and warning operating range represent the error operating range area.

REQ	6.18 GT basic configuration – <class> DataParamLimit
The <class> DataParamLimit shall be implemented according to the ASN.1 definition specified in Table 13 , with details specified in A.1 .	

[Table 13](#) specifies the <class> DataParamLimit attributes.

Table 13 — <class> DataParamLimit attributes

Name	Description	Cvt
dpId	Identifier (see 7.2.2.2) of the data parameter.	M
normalMin	SNUM32 defining the minimum of the normal operating range area.	O ₀ ^a
normalMax	SNUM32 defining the maximum of the normal operating range area.	O ₁ ^a
warningMin	SNUM32 defining the minimum of the warning operating range area.	O ₂ ^a
warningMax	SNUM32 defining the maximum of the warning operating range area.	O ₃ ^a
^a At least one of the values O ₀ , O ₁ , O ₂ or O ₃ shall be set to define a DataParamLimit. Combinations are allowed as well.		

EXAMPLE DataParamLimit

```
dataTypes {
  { id '00000013'H, type numeric2: {
    decimalPlaces 1, unitId '0000000C'H, factor 1, quotient 10, addend 0 } } },
dataParams {
  { id 'FF000018'H, dataTypeId '00000013'H, accessType '10000'B, property sensor } },
dataParamLimits {
  { dpId 'FF000018'H, normalMin -300, normalMax 1200, warningMin -380, warningMax 2000 }
},
displayNames {
  { idType datatype, id '00000013'H, longname "Temperature (0.1°C)" },
  { idType dataparam, id 'FF000018'H, shortname "ECT", longname "Engine Coolant Temp." } }
```

[Figure 11](#) shows a graphical representation of the above example with a normal Engine Coolant Temperature area of –30 °C to 120 °C. The warning areas are defined from –38 °C to –30 °C and from 120 °C to 200 °C. All values outside the normal and warning operating range are defined as error values. In [Figure 11](#), the error values are below –38 °C and above 200 °C.

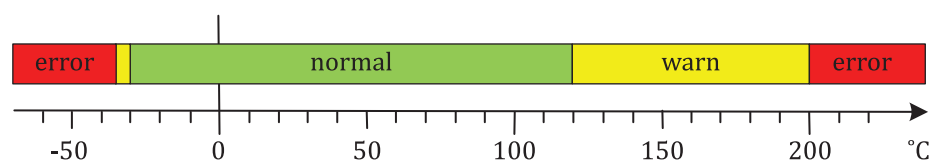


Figure 11 — DataParamLimit example

7.2.2.8 <class> EcuDataParam

The <class> EcuDataParam specifies the fusion of ECU and data parameter by identifier reference. A value is exactly defined if both ECU and data parameter are given.

REQ	6.19 GT basic configuration – <class> EcuDataParam
The <class> EcuDataParam shall be implemented according to the ASN.1 definition specified in Table 14 , with details specified in A.1 .	

[Table 14](#) specifies the <class> EcuDataParam attributes.

Table 14 — <class> EcuDataParam attributes

Name	Description	Cvt
ecuId	Identifier (see 7.2.2.2) of an ECU (see 7.2.2.3).	M
dpId	Identifier (see 7.2.2.2) of a data parameter (see 7.2.2.6).	M
subIndex	If the DataType of the data parameter is an array or a hashtable, the subIndex references the corresponding sub element..	0

7.2.2.9 <class> EcuDataParamValue

The <class> EcuDataParamValue assigns a value and a time stamp to an ECUDataParameter (7.2.2.8) defined by ECU identifier and data parameter identifier. <class> EcuDataParamValue is used to set-up fix and initial values for data parameters in configurations and to set values within the GTP service *SetValuesCall* (see ISO 13185-2).

REQ	6.20 GT basic configuration – <class> EcuDataParamValue
The <class> EcuDataParamValue shall be implemented according to the ASN.1 definition specified in Table 15, with details specified in A.1.	

Table 15 specifies the <class> EcuDataParamValue attributes.

Table 15 — <class> EcuDataParamValue attributes

Name	Description	Cvt
ecuId	Identifier (see 7.2.2.2) of an ECU (see 7.2.2.3).	0
dpId	Identifier (see 7.2.2.2) of a data parameter (see 7.2.2.6).	M
value	The fix or initial value of the data parameter.	M
timestamp	Timestamp; Time stamp in milliseconds, between the current time and midnight, January 1 st , 2004 UTC (coordinated universal time).	0

The example is an extract of a personalized vehicle configuration, including values for the licence plate ("265G21"), the vehicle model year (2018), and the vehicle propulsion type (1: Diesel).

EXAMPLE EcuDataParamValue

```
ecuDataParamValues {
  { ecuId '00000002'H, dpId '000000E5'H, value textString: "265G21" }, -- License plate
  { ecuId '00000002'H, dpId '0000004E'H, value numeric2: 2018 }, -- Vehicle model year
  { ecuId '00000002'H, dpId '00000045'H, value bitString: 1 } -- Vehicle propulsion type
}
```

7.2.2.10 <class> EcuDataParamStd

The <class> EcuDataParamStd assigns a standardized ECU data parameter identified by ecuIdStd and dpIdStd to a manufacturer-specific ECU data parameter, identified by ecuId and dpId. REQ 6.21 derives from the use case which requests an ITS-SU to support a new standardized ECU data parameter. The configuration installed in the ITS-SU already supports a manufacturer-specific ECU data parameter of the same name. The ITS-SU configuration requires an update to include the new assigned standardized ECU data parameter specification with a mapping to the manufacturer-specific ECU data parameter.

REQ	6.21 GT basic configuration – <class> EcuDataParamStd
The <class> EcuDataParamStd shall be implemented according to the ASN.1 definition specified in Table 16 with details specified in A.1.	

Table 16 specifies the <class> EcuDataParamValue attributes.