INTERNATIONAL STANDARD

ISO 13185-3

First edition
2018-06

Intelligent transport systems — Vehicle interface for provisioning and support of ITS Services —

Part 3: Unified vehicle interface protocol (UVIP) server and client API specification

Systèmes intelligents de transport — Interface véhicule pour la fourniture et le support de services ITS —

Partie 3: Serveur du protocole unifié pour l'interface véhicule (UVIP) et spécification de l'API client

Reference number
ISO 13185-3:2018(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

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 on 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 the following URL: www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*.

Introduction

This document has been established to define the UGP client and server Java API of a common software interface to a vehicle UGP server to easily exchange vehicle information data amongst nomadic and/or mobile devices, cloud servers, vehicle servers and the vehicle's Electronic Control Units (ECUs).

Applications supporting service provision use via nomadic and mobile devices need vehicle information data through an in-vehicle interface access method as well as the harmonization of existing standards to support a single vehicle data access solution.

A Nomadic Device (ND) becomes a Personal ITS station (P-ITS-S) if a Hardware Security Module (HSM) and software, that prohibits unauthorized access to an ITS-secured domain inside the ND, has been implemented.

This document defines the UGP client and server Java API protocol between the P-ITS-S and the UGP server in the vehicle.

The protocol specified in this document is based on the Open Systems Interconnection (OSI) Basic Reference Model specified in ISO/IEC 7498-1 and ISO/IEC 10731, which structures communication systems into seven layers.

This document may be used by vehicle manufacturers to implement an interoperable UGP server in on-board communications modules that are allowed to interface with P-ITS-S(s). Through this interface, P-ITS-S(s) can access in-vehicle information provided to the UGP server. The means by which the UGP server obtains the in-vehicle information is outside the scope of this document.

The P-ITS-S applications need vehicle information data through an in-vehicle interface access method.

This document supports ITS applications based on a client-server model which allows clients on P-ITS-S to obtain data from ECUs in the in-vehicle networks (IVNs) through a common interface to a server located in a Vehicular ITS station (V-ITS-S) which in turn is acting as a gateway to the IVNs. The protocol implementation in the vehicle's UGP server may include the following features:

— the denial of access to the vehicle's UGP server data by unauthorized on-board and off-board test equipment;

— the denial of access to parts of the vehicle's UGP server data by unauthorized on-board and off-board test equipment (privacy);

— the identification of the vehicle's UGP server and the vehicle it is installed in;

— the list of in-vehicle connected ECUs to the vehicle's UGP server and their data parameters;

— methods to configure the access to vehicle data.

Intelligent transport systems — Vehicle interface for provisioning and support of ITS Services —

Part 3:
Unified vehicle interface protocol (UVIP) server and client API specification

1 Scope

This document specifies the server and client APIs of the Unified Gateway Protocol (UGP). Figure 1 shows an overview of the UGP client and server API. A UGP client application on a P-ITS-S communicates with a UGP server application on a V-ITS-S. The UGP client application implements the UGP client API using ISO 13185-2. The UGP server application implements the UGP server API using ISO 13185-2.

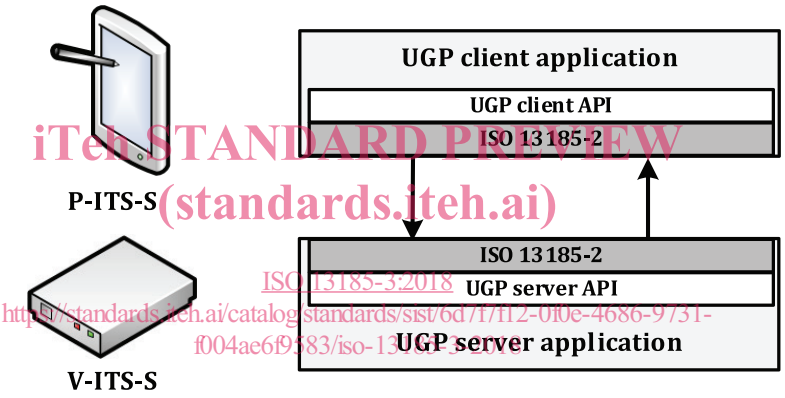

Figure 1 — UGP client and server API

NOTE This document does not define the UGP client and server API in other languages than Java.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 13185-1, *Intelligent transport systems (ITS) — Vehicle interface for provisioning and support of ITS services — Part 1: General information and use case definition*

ISO 13185-2, *Intelligent transport systems (ITS) — Vehicle interface for provisioning and support of ITS services — Part 2: Unified gateway protocol (UGP) requirements and specification for vehicle ITS station gateway (V-ITS-SG) interface*

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

3 Terms, definitions, symbols and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 13185-1, ISO 21217 and the following apply.

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

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at http://www.iso.org/obp

3.1
authentication
cryptographic service that provides assurance that the sender of a communication is who they claim to be

Note 1 to entry: Authentication might involve confirming the identity of a person or software program, tracing the origins of an artefact, ensuring that a product is what it's packaging and labelling claims to be.

3.2
authorisation
prescription that a particular behaviour shall not be prevented

[SOURCE: ISO 17419, 3.1]

3.3
nomadic device
ND
provides communications connectivity via equipment such as cellular telephones, mobile wireless broadband (WIMAX, HC-SDMA, etc.), WiFi, etc. and includes short range links, such as Bluetooth, Zigbee

Note 1 to entry: Nomadic devices do not necessarily implement ITS-specified security, e.g. hardware security module.

3.4
privacy
choice made by the vehicle owner to grant information access for a special tool or user, or if the data should be used in the vehicle/off-board systems or not

Note 1 to entry: The privacy/authorization information is kept as master information off-board and synchronised to the on-board V-ITS-S.

3.5
UGP client
entity that implements the UGP client services

EXAMPLE ND, P-ITS-S.

3.6
UGP server
entity that implements the UGP server services

EXAMPLE V-ITS-S.

3.7
unified gateway protocol
UGP
application layer protocol to enable UGP clients to access data from the UGP server

4 Abbreviated terms

API	application programming interface
ASN.1	abstract syntax notation one
A_PDU	application layer protocol data unit
A_SDU	application layer service data unit
Conf	confirmation
CRC	cyclic redundancy check
DTC	diagnostic trouble code
ECU	electronic control unit
GUI	graphical user interface
Ind	indication
ITS	intelligent transport systems
ND	nomadic device
OSI	open systems interconnection
P-ITS-S	personal – intelligent transport system – station
Req	request
Resp	response
UGP	unified gateway protocol
V-ITS-S	vehicle – intelligent transport system – station

5 Conventions

This document is based on the conventions discussed in the OSI Service Conventions (ISO/IEC 10731:1994)[1] as they apply for communication services.

These conventions specify the interactions between the service user and the service provider. Information is passed between the service user and the service provider by service primitives, which may convey parameters.

Figure 2 shows a high level flow diagram of UGP.

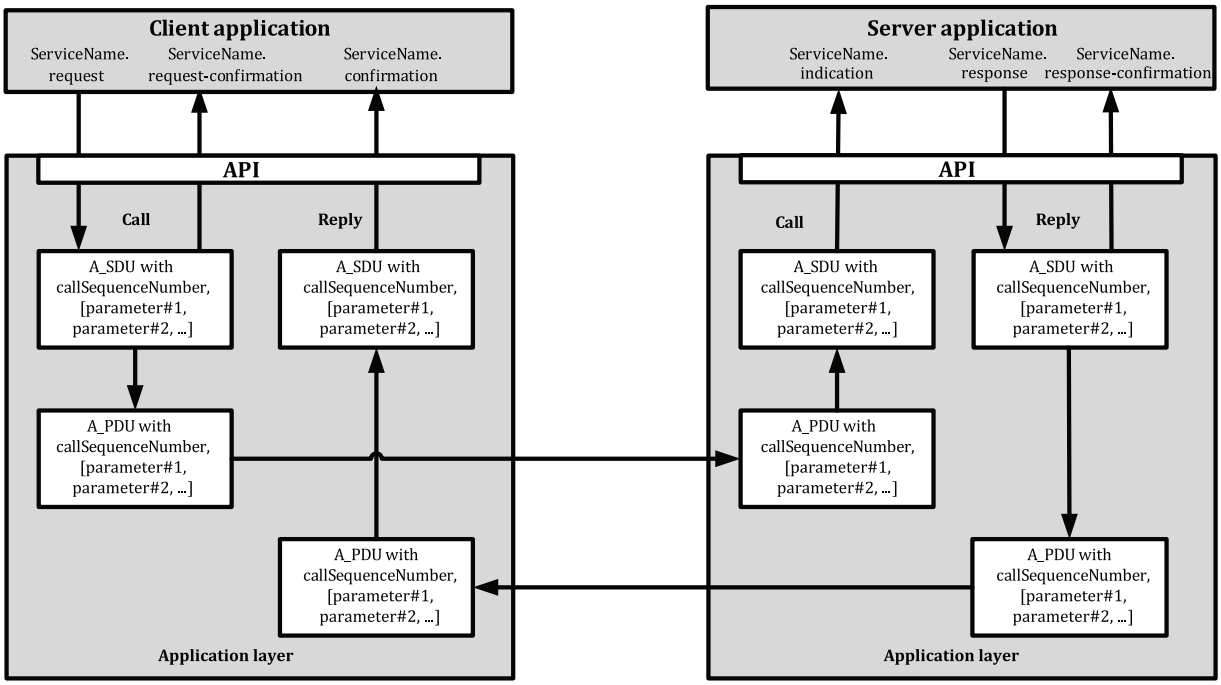

Figure 2 — UGP client-server relationship

This document defines services using the six service primitives: request, req_confirm, indication, response, rsp_confirm and confirmation as defined in ISO 13185-2:2015, Clause 6.

The request and indication service primitives always have the same format and parameters. Consequently for all services the response and confirmation service primitives (except req_confirm and rsp_confirm) always have the same format and parameters. When the service primitives are defined in this document, only the request and response service primitives are listed.

6 UGP API concept

6.1 UGP client and server architecture

Figure 3 shows the UGP client and server architecture. On the P-ITS-S there is a UGP client application and on the V-ITS-S there is a UGP server application.

The UGP client application uses the interface IUGPRequest to request services and the interface IUGPConfirmation to receive the corresponding confirmation. In the confirmation implementation another request can be called or the GUI can be updated. The UGPRequest Implementation shall encode the request into a corresponding ISO 13185-2 Call. Before calling the IUGPConfirmation the UGP client application shall decode and interpret the ISO 13185-2 Reply.

The UGP server application shall decode ISO 13185-2 Calls and interpret it to call its interface IUGPIndication. The corresponding implementation shall get data, DTCs, etc. and to create the corresponding response by calling the interface IUGPResponse. The UGPResponseImplementation shall encode the response into an ISO 13185-2 Reply.

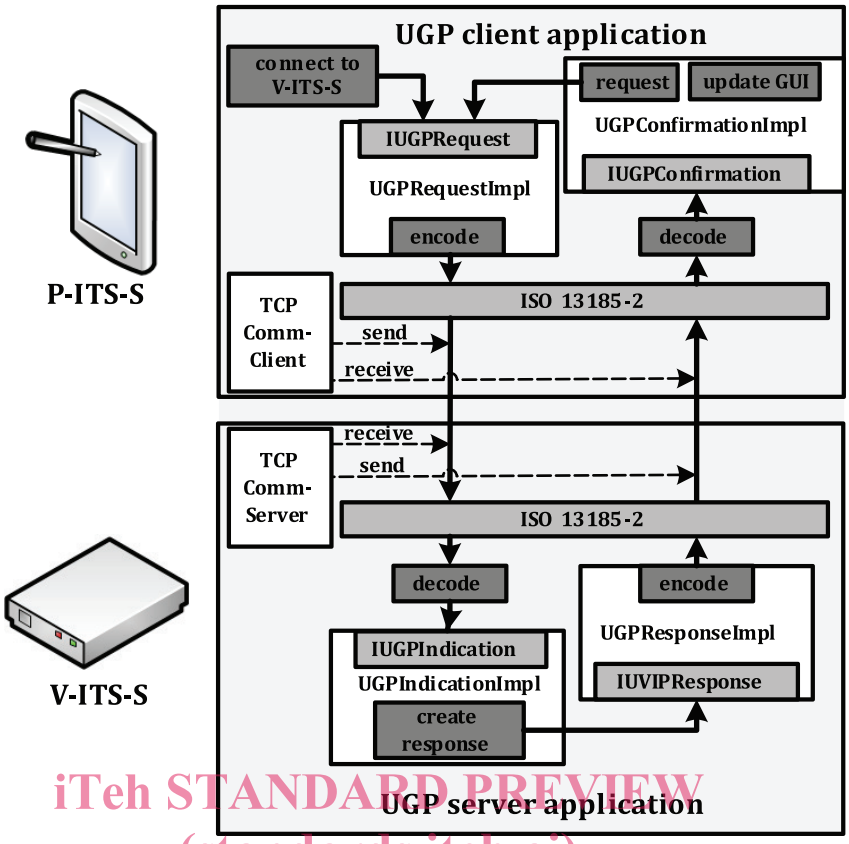

Figure 3 — UGP client and server architecture

6.2 UGP communication flow

Figure 4 shows the UGP communication flow. The UGP client application shall create a connection to the V-ITS-Ss UGP server application. Then it calls an authenticationRequest which is encoded into an ISO 13185-2 authenticationCall. The UGP server application decodes the authenticationCall and calls the authenticationIndication. If the authenticationKey is not OK for the V-ITS-S, the UGPIndication implementation creates a negativeResponse which will be decoded as ISO 13185-2 globalNegativeReply. The UGP client application decodes the globalNegativeReply and calls a negativeConfirmation. Usually the UGP client application should update its GUI with an error message “Not allowed to access”.

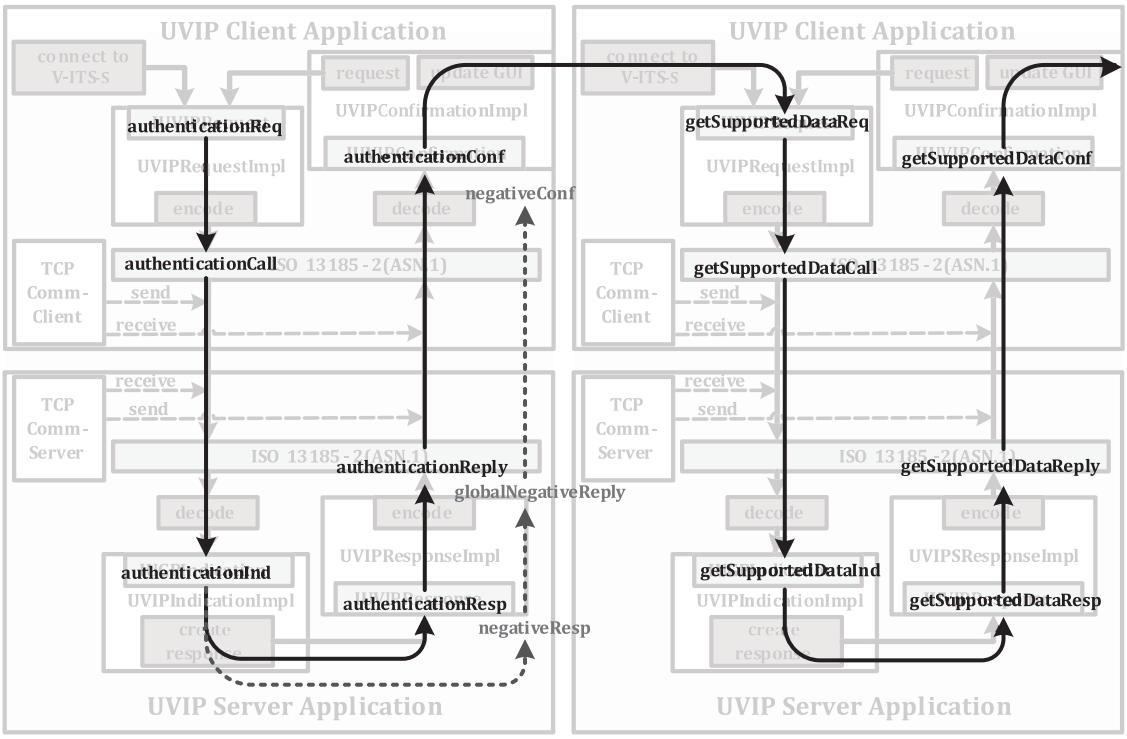

Figure 4 — UGP communication flow

If the authenticationKey is OK for the V-ITS-S, the UGPIndication implementation calls an authenticationResponse which will be decoded as ISO 13185-2 authenticationReply. The UGP client application decodes the authenticationReply and calls an authenticationConfirmation. The authenticationConfirmation can indicate the authentication authorization bits but shall call now a getSupportedDataRequest.

The UGP client application encodes the getSupportedDataRequest into an ISO 13185-2 getSupportedDataCall. The UGP server application receives and decodes it and calls a getSupportedDataIndication which creates the requested supported data infos in a getSupportedDataResponse. This is decoded into an ISO 13185-2 getSupportedDataReply. The UGP client application receives and decodes the getSupportedDataReply and calls a getSupportedDataConfirmation.

The data parameters or DTCs can be requested by getValueReq or getDtcInfoReq, etc.

6.3 UGP interfaces and service primitives

Table 1 defines the UGP interfaces and service primitives.

Table 1 — UGP interfaces and service primitives

IUGPRequest	IUGPIndication	IUGPResponse	IUGPConfirmation
authenticationReq	authenticationInd	authenticationResp	authenticationConf
getSupportedDataReq	getSupportedDataInd	getSupportedDataResp	getSupportedDataConf
getValueReq	getValueInd	getValueResp	getValueConf
setValueReq	setValueInd	positiveResp	positiveConf
controlValueReq	controlValueInd	controlValueResp	controlValueConf
getDtcInfoReq	getDtcInfoInd	getDtcInfoResp	getDtcInfoConf
clearDtcInfoReq	clearDtcInfoInd	positiveResp	positiveConf

Table 1 *(continued)*

IUGPRequest	IUGPIndication	IUGPResponse	IUGPConfirmation
enablePassThruReq	enablePassThruInd	positiveResp	positiveConf
listFileReq	listFileInd	listFileResp	listFileConf
manageFileUploadReq	manageFileInd	manageFileResp	manageFileConf
manageFileDownloadReq	manageFileInd	manageFileResp	manageFileConf
manageFileDeleteReq	manageFileInd	manageFileResp	manageFileConf
resetReq	resetInd	positiveResp	positiveConf
stopServiceReq	stopServiceInd	positiveResp	positiveConf

7 UGP Java client API

7.1 UGP Java client API requests

7.1.1 Overview

Corresponding to the UGP services defined in ISO 13185-2, the UGP Java client requests API (IUGPRequest) contains request services that can be called by any application to execute a UGP service (Req). The UGP Java client API confirmations (see 7.2) handle the corresponding replies.

Table 2 defines the UGP Java client API requests (IUGPRequest).

Table 2 — UGP Java client API requests (IUGPRequest)

Service	API definition
authenticationReq	UGPPackage authenticationReq(String authenticationKey) throws VIException;
getSupportedDataReq	UGPPackage getSupportedDataReq(SupportedDataFilter supportedDataFilter) throws VIException; UGPPackage getSupportedDataReq(SupportedDataFilter supportedDataFilter, Integer[] ecuIds, AccessType accessType, Integer dataParamProperty) throws VIException;
getValueReq	UGPPackage getValueReq(int testInterval, Integer rvId, ComplexCondition condition) throws VIException; UGPPackage getValueReq(int testInterval, Integer[] rvIds, ComplexCondition condition) throws VIException; UGPPackage getValueReq(int testInterval, Vector<DataParamMapping> mappings, ComplexCondition condition) throws VIException;
setValueReq	UGPPackage setValueReq(Vector<DataParamValueMapping> valueMappings) throws VIException;
controlValueReq	UGPPackage controlValueReq(int testInterval, Integer rvId, Vector<DataParamValue> values, ExecutionType execute) throws VIException; UGPPackage controlValueReq(int testInterval, Integer[] rvIds, Vector<DataParamValue> values, ExecutionType execute) throws VIException;
getDtcInfoReq	UGPPackage getDtcInfoReq(int testInterval, Integer ecuId, boolean withEnvData, ComplexCondition condition) throws VIException; UGPPackage getDtcInfoReq(int testInterval, Integer[] ecuIds, boolean withEnvData, ComplexCondition condition) throws VIException;
clearDtcInfoReq	UGPPackage clearDtcInfoReq(Integer ecuId) throws VIException; UGPPackage clearDtcInfoReq(Integer[] ecuIds) throws VIException;
enablePassThruReq	UGPPackage enablePassThruReq(String label, String key) throws VIException;