VDV Die Verkehrsunternehmen

VDV-recommandation 301-2

01/2014

Internet Protocol based Integrated On-Board Information System IBIS-IP

Part 2: Interface Specification V1.0

Gesamtbearbeitung

Overall editing:
Expert committee for Telematics and Information Systems ATI
Sub-committee for Communications and Information Systems UA-KIS
Contributors:
Dipl.-Ing. Dirk Weisser, Init, Karlsruhe
Dr. Torsten Franke, IVU, Aachen
Dr. Holger Bandelin, Scheidt & Bachmann, Mönchengladbach
Dipl.-Ing. Berthold Radermacher, VDV, Cologne
Dipl.-Ing. Andreas Wehrmann, VDV, Cologne
Dipl.-Ing. ETH Walter Meier-Leu, we, Schaffhausen
Translation was supported by:
Dipl.-Ing. (FH) Karsten Baumeister, Annax Anzeigesysteme GmbH, Brunntal
Dr. Torsten Franke, IVU, Aachen
Dipl.-Ing. ETH Patrik Studer, Trapeze, Neuhausen am Rheinfall, Switzerland
Dipl.-Ing. Samuel Weibel, Gorba, Oberbüren, Switzerland
Dipl.-Ing. Dirk Weißer, Init, Karlsruhe

Gefördert durch:

Bundesministerium für Wirtschaft und Energie

aufgrund eines Beschlusses des Deutschen Bundestages

The project IP-KOM-ÖV, which is the basis of this VDV recommendation, was supported with means of the Federal Ministry for Economy and Energy under the support code 19P10003. The responsibility for the contents of this publication lies with the authors.

Table of contents

Foreword

Initiated by the VDV and promoted by BMWi, the research and standardization project Internet Protocol based Communication Services in Public Transportation (in German: *Internet Protokoll basierte Kommunikationsdienste im öffentlichen Verkehr (IP-KOM-ÖV)*) was started in September 2010.

The project is supported by 14 partners from industry, universities, and transportation companies. It is used for developing modern communication concepts for comprehensive and continuous passenger information.

Nowadays, comprehensive passenger information is a decisive competitive feature in public transportation, not only compared to other transportation companies, but also compared to individual traffic.

It is already common that transportation companies do not only inform their passengers about the planned trips, but also provide real-time information e.g. regarding current delays, events, or changes in destination. This information is provided to all present passengers via public displays and/or announcements in the vehicles or at the stops. Such information can also be individually inquired by single persons using special applications (apps) or via Web offerings.

However, so far it has not been possible to provide passengers in public transportation with information regarding their personal trip, i.e. to guide the passenger to his destination as quickly as possible using public transportation in the case of an interruption.

Commonly used smartphones and tablets offer various possibilities and enable high user acceptance. In this context, information transfer is IP-based and should preferably be realized between a central information server and the customer end device. In the case that the central data server cannot be reached respectively the vehicle is not connected with such, communication should also be possible directly between customer end device and vehicle.

For this reason, the research and standardization project IP-KOM-ÖV works on three main subjects (cf. Figure 1).

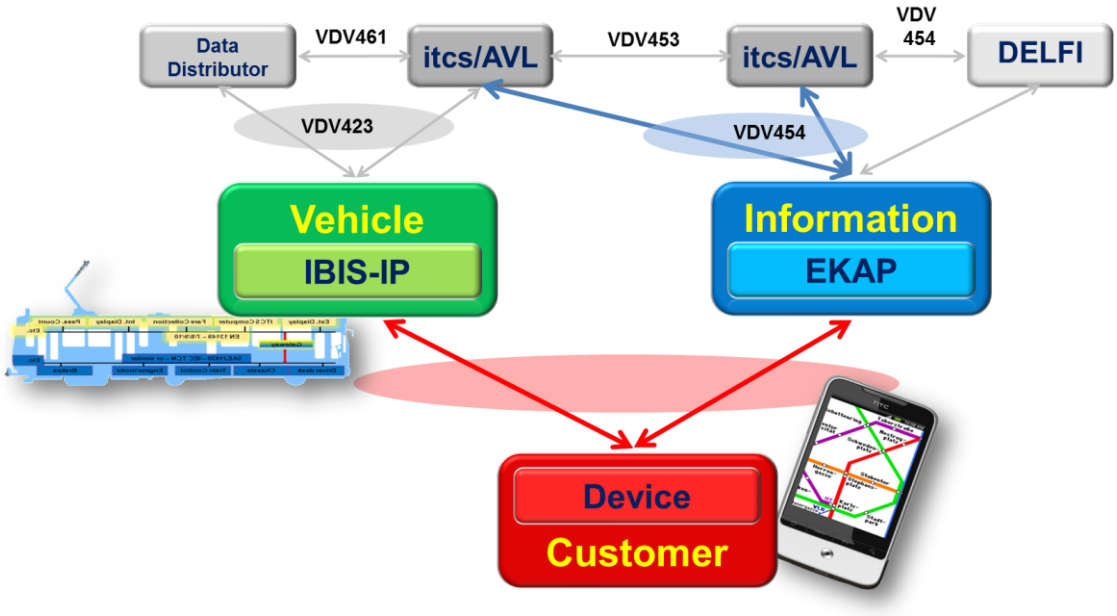

Figure 1: Environment and main subjects in the IP-KOM-ÖV project

The first main subject (green in Figure 1) is the specification of a performant IP-based communication protocol in the vehicle (IBIS-IP). The objective to meet the increased needs of vehicle-internal passenger information and communication. For this purpose, the IBIS Wagenbus developed in the eighties (VDV recommendation 300) is converted to a modern IP information structure. Another objective is the definition of an IP-based interface for transferring information from the vehicle to the mobile customer end device. This document is the result of this part of the research project.

The second main subject (red in Figure 1) is the definition of the required interfaces in order to provide future applications for individual passenger information using the mobile devices of the passenger (smartphones, tablet PCs, etc.) in a standardized manner. For this purpose, the passenger needs regarding individual information were determined in the first step. In the second step, standardized interfaces were developed between the real-time communication and information platform (EKAP, cf. third main subject) and the mobile customer end devices and/or between the EKAP and the background systems. Here, data modeling and architectures were researched and specified only. The development of an application for mobile end devices was explicitly not planned and was merely realized in the simplest form for test purposes. The results of this part of the research project are documented in VDV recommendation 430.

The third main subject (blue in Figure 1) is the definition and creation of an EKAP. The EKAP bundles information from itcs and other information systems, and provides the applications on the customer end devices with a variety of information via suitable interfaces in a standardized manner. This platform enables apps developers to dynamically provide customers with messages about interruptions. The results of this part of the research project are also documented in VDV recommendation 431.

Beyond that, the practical suitability of this new standard is verified in lab and field tests.

0 Translation Disclaimer

This document is an English translation of the German document “Internetprotokoll basiertes integriertes Bordinformationssystem IBIS-IP, Teil 2: Schnittstellenspezifikation, v1.0” released in January 2014. It is provided for convenience only and has no legal effect.

The translation of this document was done with the greatest possible accuracy. In case of any inconsistency the German original document applies.

1 Introduction

Please refer to the introduction in VDV recommendation 301-1.

2 Scope

Please refer to the scope in VDV recommendation 301-1.

3 Introduction to IBIS-IP

Several introductory implementation definitions are described at the beginning of this document. Knowledge of part 1 (VDV recommendation 301-1) is assumed.

3.1 Requirements on Devices in IBIS-IP

The requirements on devices and services resulting from the system drafted in part 1 (VDV recommendation 301-1) are described in the following sections.

Thus, a device must meet the following minimum requirements to participate in IBIS-IP:

- Ethernet interface
- Detection of the installation position
- TCP/IP and/or UDP/IP stack implementation
- Ability to process HTTP protocols
- Memory for device-specific configuration data
- Implementation of DNS-SD functionality
- Ensures no feedback into the connected system (if connection exists)
- Provision of a *DeviceManagementService* (cf. chapter 5.2)
- Optional: Provision of a web-based manufacturer-specific maintenance interface that can be addressed by URL (in *DeviceManagementService*).

3.2 Configuration Parameters

A large part of the system behavior can be configured in an IBIS-IP system. Using the system configuration, it can be defined, which services run on which devices in which IBIS-IP version.

Thus, an IBIS-IP system configuration contains information,

- which devices belong to the system (consisting of device class (cf. chapter 3.4) and installation identifier (cf. chapter 3.3),
- which services belong to the system in which IBIS-IP version, and
- on which device the service is running.

Vehicle-specific information (vehicle number, radio address, and device-specific configuration parameters) are of no importance for the IBIS-IP system.

3.3 Installation Identifier

The installation identifier is used to determin the mounting positon of a device by reading information stored at the device installation location. The technical implementation (plug coding, EEPRROM, USB stick, or similar) is not important. The *DeviceManagementService* of a device can then communicate the information via the installation identifier to other services, and allows an assignment of a technical device identifier (IP address, DNS name) to a functional device identifier (consisting of device class and installation position).

3.4 Device Classes

Device classes covering all currently conceivable devices were specified in the IBIS-IP system for the unique identification of the connected devices. The classification was already defined in English, as these terms are common as "enumeration" in the English-language XML (cf. chapter 9 pp).

The device classes used on IBIS-IP are defined as follows:

Device class	**Device**	**Description**
OnBoardUnit	On-board computer	Corresponds to the central unit according to VDV 300.
Validator	Ticket Validator	Stamping machine or electronic validator for E-tickets
SideDisplay	Side display	Describes the exterior side display on a vehicle.
FrontDisplay	Front display	Corresponds to the vehicle destination display.
InteriorDisplay	Interior displays	Describes the displays in the vehicle interior, these can be passenger information displays or digital advertisement displays.
TicketVendingMachine	Ticket vending machine	Describes the ticket vending machines inside the vehicle. These can be driver-operated as well as passenger-operated machines.
AnnouncementSystem	Electroacoustic system	Describes the electroacoustic system in vehicles
MMI	Driver (operation) display	Corresponds to the driver's operating unit with display possibility (e.g. touch display).
VideoSystem	Video monitoring system	Describes the video monitoring system, which is controlled via IBIS-IP and/or transfers video data.
APC	Automated passenger counting system	Describes the passenger counting system inside the vehicle. The technical counting implementation is not differentiated.
MobileInterface	Customer communication interface	Describes the interface, where passengers can retrieve information from the vehicle using their mobile end devices.
TestDevice	Test devices	Placeholder for test devices, which will be connected for system verification.

Other	Other devices	Devices which were not classified yet in this standard can be connected with this class.

Table 1: IBIS-IP device class classification (any sorting)

Together with the installation identifier, the device class ensures the unique functional identification of a device in an IBIS-IP system (cf. chapter 7.1.4.1)

3.5 Notations

For improved readability, the following fonts are used with special meaning in the further course of this document.

Operations:

The following format is used for operations:

Operation in Courier New, font size 11 and bold

Services:

The following format is used for services:

Service in Courier New, font size 11 and italic

Syntax:

A syntax is represented in the following format:

<Syntax in Courier New, font size 11, and enclosed in <>>

Path:

A path is represented in the following format:

Path in Courier New, font size 11 without leading or trailing /

3.6 IBIS-IP Version

The IBIS-IP standard is described for the first time in VDV recommendation 301-2. This version 1.0 (cf. chapter 9 pp) was specified separately from the research and standardization project IP-KOM-ÖV. To enable migration configurations of different versions in one system, this version information is used during data exchange. Further information, cf. chapter 5, 5.3.2 or the example in 7.1.4.2.2.

4 Communication Protocols used

This chapter describes, under which aspect in IBIS-IP respective communication protocols are used. First, the basics of the lower-level exchange protocols are defined. Next, a short discussion of the functional context and the resulting definition of the communication protocol located in the application layer are given. Finally, the status of IBIS-IP regarding other communication protocols is described.

4.1 Addressing

4.1.1 IP Addresses

The IP addresses are allocated decentralized in IBIS-IP using a part of "Zero Conf" (cf. RFC 2927 for automatic address allocation). Possibly fixed address ranges must be de-

fined project-specifically. With respect to the address ranges, RFC already provides specifications (169.254.xxx.xxx), which must be observed for an interoperable network. However, the only important requirement is that these address ranges must be consistent among all participants.

4.1.2 Subnet Masks/Gateways

The addresses to be used must be defined project-specifically.

4.2 Configuration Parameters for TCP and UDP

The communication parameters of the TCP and UDP protocols must not be specially adapted for IBIS-IP. The definition of a port is not required, as this information is communicated via DNS-SD (cf. chapter 5.3.1). Every application and/or device manufacturer can freely select the ports. In terms of standard conformity, ports in the range of 0-1024 should be omitted (cf. RFC 6335).

4.3 Information Validity Period

As described in VDV recommendation 301-1, the information can mainly be divided into two classes based on their validity period. There are

- periodic information, and
- event triggered information.

If, during the implementation of a functional component in the service (cf. chapter 5), it is determined that this functional component provides cyclic as well as event-controlled information, different services must be provided.

4.3.1 Periodic Information

There is information

- that changes or can change periodically in short intervals (~every second), and
- for which a fast and concurrent transport of information to the consumer is more important than reliable information transfer.

This is normally information that is provided by devices or interfaces in short time intervals. Among others, this includes

- Time (every second)
- Odometer (<=every second)
- GNSS coordinates (<=every second)

Periodic information normally changes without causing a change in the operating or functional state of a vehicle. It is normally provided by the services of the base functions.

Example:

In a vehicle that moves at walking speed, the odometer values, GPS coordinates, and the time change every second. However, the operating or functional state of the vehicle is changed occasionally only, e.g. when a new stop is reached.

4.3.2 Event triggered Information

There is information

- that changes less frequently than every second and in an event-driven manner,

- that normally provides information about the functional or operating state of a vehicle,
- for which it must be ensured that all consumers are reliably informed about this change.

Example:

When a new stop is reached, the functional state of a vehicle changes. It should be ensured that all services and applications are informed, for which this change is relevant, such as interior passenger displays, passenger announcement, driver MMI, mobile customer end devices, automatic passenger counting, itcs, etc.

4.4 Use of the UDP and HTTP Protocols

The following definitions are given for IBIS-IP in order to support the different functional contexts of periodic and event triggered information:

- The UDP protocol is used for transferring periodically changing data. The data is sent in suitable time intervals to a multicast address.
  The respective service is then referred to as UDP service.
  The time intervals are selected according to the functional requirements and the technical possibilities of the interfaces delivering information. Based on the current technical circumstances, a cycle time of less than 1 sec must be avoided for UDP services.
- The HTTP protocol is used for transferring data that changes in an event-driven manner.
  The respective service is then referred to as HTTP service.
  Based on this decision, every device that offers an HTTP service, must have the functionality of an own HTTP server. In the case of HTTP services, a point-to-point connection is available for the communication between two participants. Establishment and reliable information transfer are ensured by the HTTP protocol stack.

4.5 Other Protocols

The established standard protocol for time synchronization "SNTP" is used for the time-server in the system. The Simple Network Time Protocol (SNTP) allows the distribution of the current time in a network in a simple manner using UDP. Further information regarding the implementation, cf. chapter 9.12.

During the implementation of other functional components in services, it is expected that further IP-based communication protocols will be used (e.g. RTP protocol for audio and video data streaming). Protocols going beyond that are not considered in version 1.0 of IBIS-IP.

5 Service Publication and Communication

Applications and services within an IBIS-IP system know the

- service name, and
- IBIS-IP version

of the services, with which they must communicate.

These properties are application and/or service-inherent, i.e. must not be configured.

Example:

The application of a passenger display requires the `CustomerInformationService` *service (represents the Customer Information Determination) in version X, in order to be able to display the passenger information data. The passenger display cannot handle the data of other services or data of version X+1.*

However, they are not sufficient to uniquely address a service system-wide (cf. chapter 7.1.4.2.1), and thus to use it. To uniquely find the service system-wide, the information, on which device the required service is running, is needed. Where and on which device which service is running is defined in the specific configuration of a system, i.e. it is not inherent to the application or service.

5.1 From functional Components to Services

An IBIS-IP system consists of several functional components, which exchange data. A functional component can be

- an abstract interface to another system or device,
- an application,
- a service, or
- a device

(cf. VDV recommendation 301-1).

All applications and services of an IBIS-IP system run on devices, which communicate by IP. In this context, IBIS-IP applications and IBIS-IP services represent the software implementation of functional components.

Applications and services differ in that applications in terms of the IBIS-IP system architecture are pure information recipients, i.e. they do not provide data via an IBIS-IP interface to other services or applications. Services, on the other side, offer data exchange interfaces for other services and applications of the IBIS-IP system. The services offer a list of operations for this purpose. Using these operations, data can be exchanged with the services or actions triggered in the services.

Every device that participates in an IBIS-IP system and that should be monitored by the IBIS-IP system regarding its status and addressability features at least one `DeviceManagementService` service, which represents the device management functional component. Device management in particular provides operations, using which further services can be started on a device.

Data is exchanged with the services in IBIS-IP in the form of XML data. Depending on the functional context, the XML data structures are exchanged via the HTTP or UDP protocol.

Example:

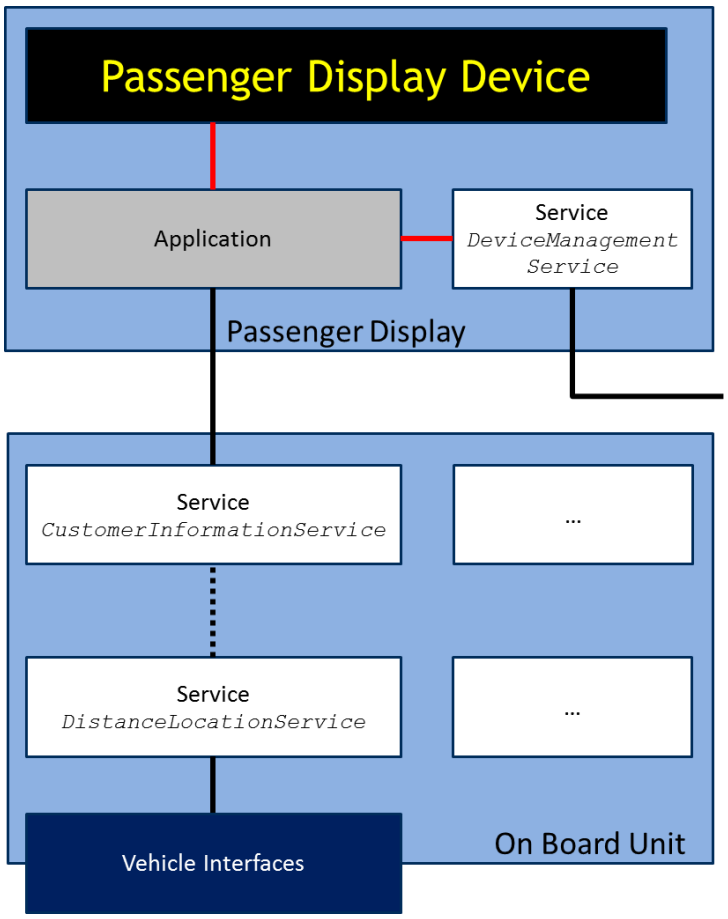

Figure 2: Example for the different representations of a functional component. Black lines: IBIS-IP, red lines: proprietary. Please observe that the passenger display functional component has device properties, application properties, and service properties.

Two devices participate in the system (cf. chapter 3.4):

- *passenger display, and*
- *on-board computer.*

The following services run on the on-board computer:

- `CustomerInformationService` *(represents the customer information determination), and*
- `DistanceLocationService` *(represents a part of physical locating).*

The `DistanceLocationService` *provides data (via several intermediate steps) that contributes to the determination of the correct information for the customer information determination. For this purpose, the* `DistanceLocationService` *gets data (odometer information, GNSS position data) from the vehicle interfaces.*

The `CustomerInformationService` *provides the respectively current passenger information.*

An application runs on the passenger display, which

- *retrieves data from the* `CustomerInformationService`*, and*
- *represents it via a proprietary interface on the display unit of the passenger display.*

The application does not provide any further information to other IBIS-IP functional components within the scope of IBIS-IP. Thus, it is not a service.

A `DeviceManagementService` runs on the passenger display, using which the passenger display can be monitored by the IBIS-IP system in terms of the application and the device.

5.2 Specified Services

With respect to the system architecture presented in VDV recommendation 301-1, the services specified in version 1.0 are now presented. Table 2 shows the connection between services and functional components, and a brief description. The detailed XML definition can be found in chapter 9 pp.

The following services are specified in version 1.0.

Service	**Functional component**	**Brief description**
BeaconLocationService	Physical locating	The service transfers the information of the location beacon via an HTTP connection. Cf. chapter 9.1
CustomerInformationService	Customer information determination	As central information source for all matters of passenger information, this service ensures in IBIS-IP the consistent provision of all data. Cf. chapter 9.2
DeviceManagementService	Device management	Runs on every device, and is used to manage further services on the device and to provide information about the device and the running services. Cf. chapter 9.1
DistanceLocationService	Physical locating	This service evaluates the odometer mileage impulses in IBIS-IP. Cf. 9.4
GNSSLocationService	Physical locating	For the distribution of the current vehicle location describable in coordinates, based on the NMEA telegram. Cf. chapter 9.5
JourneyInformationService	Journey information determination	This service provides timetable data to inquiring services. Cf. chapter 9.6
NetworkLocationService	Network locating	The service provides in-

		formation about the current location on a planned line route in a public transportation network. Cf. 9.7
PassengerCountingService	Passenger counting	This service provides the passenger counting data of every door to the IBIS-IP system. Cf. chapter 9.8
SystemDocumentationService	System documentation	The service is responsible for documenting system messages and providing the system configuration in IBIS-IP. Cf. 9.9
SystemManagementService	System management	The main tasks are starting of services according to the system configuration, monitoring of the current device and service status, and provision of system state information. Cf. chapter 9.10
TicketingService	Ticketing data determination	The service provides ticketing functionalities in the IBIS-IP system. Cf. chapter 9.11
TimeService	Time determination	The time service provides the current time, date, and time zone to the system via SNTP. Cf. chapter 9.12

Table 2: Introduction into the services specified in version 1.0

5.3 Publication via DNS-SD

Known standard network technology mechanisms are used in this method for information use and distribution to the services. Using so-called SRV records (cf. chapter 5.3.1) and TXT records (cf. chapter 5.3.2), the communication method is generally summarized under the name DNS-SD (Domain Name System – Service Discovery). Free-of-charge libraries [6] are available for the different operating systems, which allow working with this technology. When these libraries are used, it is normally transparent to the user, which records are used for transferring which data.

5.3.1 Use of SRV Records

Based on the standardized IP communication protocol UDP and the mechanisms, using which information is exchanged about computer names (so-called DNS records) in IP-based networks, an extension of this name exchange method was specified by the In-

ternet Engineering Task Force (IETF) in RFC 2782 specifically for the publication of services.

Using this standardized extension of the DNS records, the so-called SRV records, it can be automatically announced within IP networks of the devices, which services are offered on these devices, and how these services can be addressed.

The structure of such SRV records follows the pattern

```
<_Service._Proto.Name TTL Class SRV Priority Weight Port Target>.
```

The following is an example

```
<CustomerInformationService._ibisip_http.local.  3600  IN  SRV
10  0  389  OnboardUnit_1.local. >
```

The meaning of the fields is described below.

Data field (according to RFC 2782)	**Example value**	**Description**
`Service`	`CustomerInformationService.` (with . at the end)	Service name (possible values for services in IBIS-IP can be found in chapter 5.2)
`Proto`	`_ibisip_http.` (with . at the end)	Communication protocol, there are two values only in IBIS-IP: `_ibisip_udp` and `_ibisip_http` (compare also to 4.2, 5.4 and 5.5)
`Name`	`local.` (with . at the end)	Domain of the service
`TTL (time to live)`	`3600`	Validity period of the record (in seconds)
`Class`	`IN`	Class according to RFC 1035, SRV records belong to the `IN` class, `IN` stands for Internet
`SRV`	`SRV`	Type of the record, i.e. here a `SRV` record
`Priority`	`10`	Priority for the content (here the service). If there are two services with the same name, the service with lower priority is preferred
`Weight`	`0`	Weight for the content (here the service). If there are two services with the same name, the service with the lower weight is preferred
`Port`	`389`	Indicates, on which port the announced service can be accessed

`Target`	`OnboardUnit_1.` (with `.` at the end)	DNS name or IP address of the destination host computer, where the service runs

Table 3: Meanings of the SRV record in DNS-SD

5.3.2 Use of TXT Records

As not all information for the service identification relevant to IBIS-IP can be accommodated in the standard SRV record, further information is announced using so-called TXT records (cf. RFC 1464).

The TXT record allows to provide any attribute-value pairs in the form

`<Attribute Name>=<Attribute Value>.`

The following attributes are used in IBIS-IP:

Attribute	Optional/ Mandatory	Example value	Description
`ver`	Mandatory for all IBIS-IP services	`1.0`	IBIS-IP version of the service (cf. chapter 3.6)
`path`	Optional	`testversion_1.1`	Path information, if available, must be considered in addition to port and destination host from the SRV record for technical addressing of a service (cf. chapter 7.1.4.2pp, compare also to the example in chapter 5.5)
`multicast`	Mandatory for UDP services	`239.0.0.1`	Multicast address, using which the UDP service data is published.
`sntp-server`	Mandatory for the time synchronization service	`192.168.0.22`	IP address, under which the SNTP server can be found in the IBIS-IP network.

Table 4: Meanings of the TXT record in DNS-SD (including the definitions for IBIS-IP)

5.4 Publication of UDP Services

UDP services indicate at least the following service-specific information during their publication via DNS-SD:

In the SRV record:

- Service name
- Protocol = `_ibisip_udp`

In the TXT record:

- `ver`
- `multicast`

In order to receive information of the UDP services, the client must join the announcing multicast group. There, the respective messages are distributed by the services via UDP telegram. After joining the multicast group, the respective telegrams can be received.

An explicit information request or other mechanisms available for HTTP services (cf. chapter 4.4) are not planned.

5.5 Publication of HTTP Services

HTPP services indicate at least the following service-specific information during their publication via DNS-SD:

In the SRV record:

- Service name
- Protocol = `_ibisip_http`
- Port
- Destination host

In the TXT record:

- `ver` (mandatory)
- `path` (optional)

Example:

In order to execute operation **`GetDeviceConfiguration`** *of the* *`DeviceManagementService`* *on a device called "tst01", the decisive address for the HTTP communication is*

`<tst01:1234/DeviceManagementService/GetDeviceConfiguration>`

If an optional "`path=IBIS-IP`" is additionally entered in the TXT record of this service, the address to be addressed is

`<tst01:1234/IBIS-IP/DeviceManagementService/GetDeviceConfiguration>`

6 Conventions for Services and Data Structures

This document exclusively describes the communication with IBIS IP services, i.e. in particular

- Structuring of data exchanged in IBIS-IP,
- Data (structures) used by operations as input and output parameters (request/response structures), and
- Operations provided by the different services.

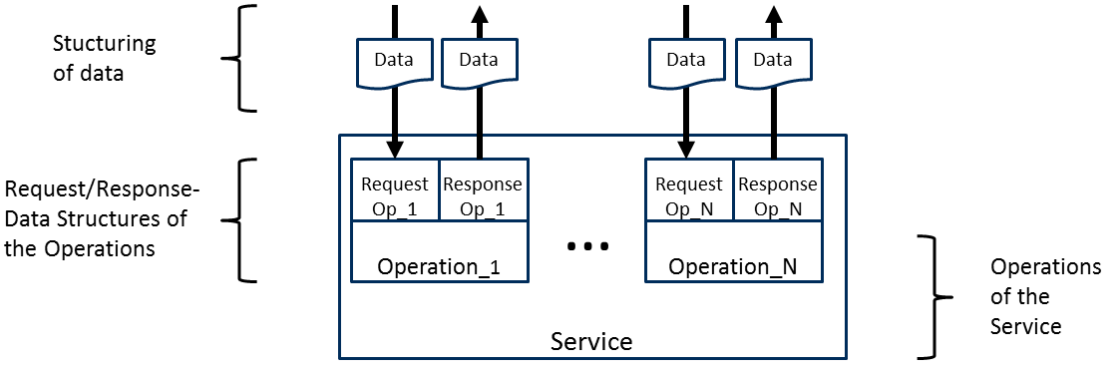

Figure 3: Representation of the structuring of data in services

In this context, the general principles of service communication are presented, and the concrete implementation of functional components in the services is described.

Devices are considered,

- As they are relevant to the context, in which the service runs, and
- As they are monitored by the IBIS-IP system.

This document does **not discuss the**

- Functional component implementation in the form of an application, and
- Abstract interfaces to other systems and devices.

6.1 Conventions for HTTP Services

HTTP services in IBIS-IP provide their functionality via operations.
The operations are used to trigger actions for the service, which concern certain objects that are managed by the service.

6.1.1 Use of HTTP-POST and HTTP-GET

IBIS-IP services offer operations,

a) To which data is provided. In this case, the operation is called via HTTP-POST.
b) To which no data is provided. In this case, the operation is called via HTTP-GET.

6.1.1.1 Operation Calls with HTTP-POST

Operation calls with HTTP-POST are always used if data should be provided to the service.

In this context, the service defines which data types and/or data structure types can be accepted and interpreted by the service.

Example:

Using the **`SetBlockNumber`** *operation of the* `JourneyInformationService` *service, a new block can be adjusted at the* `JourneyInformationService`*. For this purpose, a respective BlockNumber data structure is provided to the* `JourneyInformationService`*.*

The call is realized as HTTP-POST.

6.1.1.2 Operation Calls with HTTP-GET

Operation calls with HTTP=GET are always used, if no data is provided to the service, i.e. if the operation is a pure data request.

In particular, all HTTP-GET operations can be called using a browser via a suitable address entry.

Example:

Using the **`GetDeviceStatus`** *operation of the* `DeviceManagementService` *service, the status of the device is determined, on which* `DeviceManagementService` *is running. No parameters are required for this request.*

The call is realized as HTTP-GET.

The request can be realized using a browser by entering address line

`[…] DeviceManagementService/GetDeviceStatus.`

Based on the name convention in chapter 6.1.3, the rule that HTTP-GET must be used for all **`Get`** operations applies in particular.

6.1.2 Naming Conventions for Operations

Operation names contain at least one verb describing the action, and one object, to which the action refers.

The following verbs have been used in different areas:

In the case of purely data-oriented operations, the verbs are:

- Set,
- Retrieve,
- Get,
- List,
- Subscribe, and
- Unsubscribe.

In the case of control operations, in particular of the `DeviceManagementService`, the verbs are

- Start,
- Stop,
- Restart,
- Activate, and
- Deactivate.

For validation operations, an additional verb is

- Validate.

In order to prevent that the same verbs are used for different actions or different verbs for the same action, brief conventions were developed for the use of the verbs.

The service name construction and the conventions for using the verbs are explained below.

6.1.2.1 Purely data-oriented Operations

If the objects are data and/or data structures of the service, the following applies

Operation name = Verb + Relevant data structure

Example:
The operation for requesting the device configuration is
***`GetDeviceConfiguration`** = Get (verb) + DeviceConfiguration (relevant data structure).*

The data can be functional data of any type, e.g.

- Configuration data,
- Location data,
- Passenger information data,
- Status information,
- etc.

The purpose of data-oriented actions is normally the

- Provision of data and/or data structures OR
- Single request of data and/or data structures OR
- Subscription to data and/or data structures OR
- Termination of a subscription to data and/or data structures.

The individual actions will be described in detail below.

6.1.2.1.1 Value Adjustment for a Service

Convention:

Operations used to define a service-internal value start with verb

Set.

The name of such an operation also consists of a description of the data and/or data structure object, which is to be defined.

Example:
The operation for setting (adjusting) the device configuration is

`SetDeviceConfiguration`

6.1.2.1.2 Single Request of currently valid Data and/or Data Structures

Convention:

Operations used to request currently valid, service-internal data and/or data structures start with verb

Get.

The name of such a request operation also consists of a description of the data and/or data structure object, which is requested.

Example:
The operation for requesting the device status is

`GetDeviceStatus`

6.1.2.1.3 Subscription to Data and/or Data Structures

To ensure that a service user is informed about changes to a data structure of a service, the service user can periodically request the service using Get requests. As long as the value of the data structure does not change, data is unnecessarily exchanged. For this reason, such a method is to be avoided in IBIS-IP by using subscriptions.

With the subscription to a data structure of a service, the subscriber is automatically informed by the service about a data structure change.

Convention:

Operations used for setting up the subscription start with verb

Subscribe.

Example:
The operation for setting up a subscription to the status of a device is

`SubscribeDeviceStatus`

6.1.2.1.4 Termination of a Subscription to Data and/or Data Structures

In order to terminate the set up subscription, another operation is required.

Convention:

Operations used for canceling the subscription start with verb

Unsubscribe.

Example:
The operation for canceling a subscription to the status of a device is

`UnsubscribeDeviceStatus`

6.1.2.1.5 Operations for requesting specific Information

Services cannot only provide operations, using which access is possible to the values of currently valid data and/or data structures of the service, but also operations, using which further specific data is determined, which is not part of the currently valid data version. Such subscriptions to data are not meaningful, and are thus not planned.

Convention:

Operations used to determine specific information that depend on a provided parameter, start with verb

Retrieve

If no parameter is provided, a list of all matching information is returned in the response structure.

Examples:
The operation for determining all information for a certain block is

`RetrieveSpecificBlockInformation.`

The operation for determining all log data entries (IBIS requires the last 10) is

`RetrieveLogMessages`

6.1.2.1.6 Operations for requesting Data Lists from the basic Data Supply

Convention:

Operations used to determine all elements of a certain information available in the system, start with verb

List

The name of such a request operation also consists of a description of the data and/or data structure object, which is requested.

Example:
The operation to obtain all routes known to the system is

`ListAllRoutes`

As no parameters can be provided with the request for these operations, the possibilities presented in chapter 6.1.1.2 apply.

6.1.2.2 Control Operations

In addition to the data-oriented operations described in chapter 6.1.1, there are operations, where the object of the operation is not data or data structures, but using which the behavior of a service or device is controlled. This is particularly the case or device management.

In the case of device management, services can be stopped and/or restarted on the device using operations, or even the complete device. In this case, the object is the device or the service.

Object naming follows the pattern (verb and object) described above.

6.1.2.2.1 Start

Operations used for starting a process start with verb

Start

Example:
The operation for starting a service on a device is

`StartService`.

6.1.2.2.2 Trigger Restart

Operations used for restarting a process start with verb

Restart

Example:
The operation for triggering a device restart is

`RestartDevice`.

6.1.2.2.3 Stop

Operations used for stopping a process start with verb

Stop

Example:
The operation for stopping a service on a device is

`StopService`.

6.1.2.2.4 Device Deactivation

Operations used for deactivating a device start with verb

Deactivate

Example:
The operation for deactivating a device is
DeactivateDevice

A device deactivated via "Deactivate" can only return into is previous operating state by calling the Activate operation.

6.1.2.2.5 Device Activation

Operations used for activating a device start with verb

Activate

Example:
The operation for activating a device is
ActivateDevice

6.1.2.3 Operations for Validity Checking

Operations used for checking a provided dataset for validity start with verb

Validate

Example:

The operation for checking the validity of a ticket data record is

ValidateTicket

6.1.3 Conventions for Get/Subscribe/Unsubscribe

A Subscribe and/or Unsubscribe operation exists for all data and/or data structures that can be called up using a **Get** operation.

For all data and/or data structures, which can be subscribed using a **Subscribe** operation, there is an **Unsubscribe** operation for canceling the subscription, and a **Get** operation for a single request of a data structure value.

For every **Unsubscribe** operation for a data structure, there is a **Subscribe** operation as well as a **Get** operation for the same data structure.

6.1.4 Differentiation between Get and Retrieve

Get operations always access the currently valid data. No data or parameters are provided for the access, as the object name of the operation defines the data structure, the service responds with. For this reason, subscriptions can be set up as well.

Retrieve operations do not refer to currently valid data, but always provide a dataset, whose content defines the service response.

6.1.5 Convention for Deactivate/Activate

A device deactivated via a **Deactivate** operation distinguishes itself by its behavior as deactivated device within the scope of IBIS-IP communication. In particular, this means that none of the services on the device communicated via the IBIS-IP protocol.

The **Activate** operation switches a previously deactivated device into the state, it had after the device start. In particular, no services except the *DeviceManagementService* are active after execution of the Activate command.

6.2 Conventions for UDP Services

Similar to all functional services, UDP services are started by system management. UDP services publish their data to a multicast IP address. Other services or applications that would like to retrieve this data subscribe to the multicast IP address and receive new data periodically. Service or individual operation addressing does not exist for this service.

6.3 Conventions for special Service Behavior

6.3.1 Conventions for States

Services in IBIS-IP can be in one of the following states:

- **notrunning**
  All services are in this state at the time of system start. They remain in this state, unless they are started by the *SystemManagementService*.
- **running**
  A service only enters this state, if it can retrieve all
  - Data, it depends on, from other services, and determine and provide
  - Valid data based on this data,
  - In addition, if it was published as service per DNS-SD.
- **defective**
  A service enters this state, if it cannot connect with
  - All required services after expiration of internal timeouts, or
  - If it cannot provide valid data anymore (e.g. due to a loss of connection to required services or as the delivered data cannot be evaluated)
- **starting**
  A service that was started, but is not yet in the "running" state, is in this state. If the "running" state has not been reached after expiration of a timeout, state "defective" is reported.
- **standby**
  A service enters this state, if it can retrieve all
  - Data, it depends on, from other services, and determine and provide
  - Valid data based on this data,
  - In addition, if it was NOT published as service per DNS-SD.

The service status can be individually requested from the particular *DeviceManagementService* of the device, on which the service is running. It is furthermore periodically requested by SystemManagementService.

6.3.2 Conventions for Error Messages

There are different layers within IBIS-IP, where a communication error can occur, which are differently handled depending on the layer.

If errors occur in the network communication layer, the TCP and/or HTTP and UDP error handling mechanisms are applied. For example, if a request is sent to a wrong address, this will be answered with a normal HTTP error 404 "File not found".

If network communication is successfully established, but a faulty XML is sent for a request, the inquired service responds with a XML message, which contains further information regarding the faulty request. This information is transferred in the form of a freely selectable text.

If a service does not have all basic data required for a complete response, or if services are missing, on which its own information is based, it must be possible to send a response with correct syntax in IBIS-IP. This response must indicate any faulty or missing data with another attribute. For this purpose, the XML standard data types were replaced in IBIS-IP with own IBIS-IP types, which can contain an error code in addition to the actual value. For example, if the GPS locating fails in an IBIS-IP system, as the vehicle is in a tunnel, the *NetworkLocationService* can continue to send data; however, the values transferred in XML are marked with a "DataEstimated" error code. In this case, the data-consuming service can either discard the received information or relay it with a respective note.

6.4 Conventions for Data Structuring

In order to map the data within IBIS-IP via XML schema data in a standardized manner, the following conventions apply to the description of the XML structures for operations:

- If an operation has specific request-response structures, they must be named according to the following schema

  For requests: `Service name + „.“ + Operation name + "RequestStructure"`
  For responses: `Service name + „.“ + Operation name + "ResponseStructure"`

Example:

```
<JourneyInformation-
Service.RetrieveAllRoutesPerLineRequestStructure>

<JourneyInformation-
Service.RetrieveAllRoutesPerLineResponseStructure>
```

- The response structures always consist of a "choice" (selection possibility in XML) between the actual data content and a free error message. Here, the actual data content must be respectively named and a structure created, which must be named according to the following schema
  `Service name + "." + Data content + "Data"`
  .

Example:

```
<JourneyInformationService.AllRoutesData>
```

7 Principles in IBIS-IP

System start, the subsequent start of all services according to configuration (cf. chapter 3.2), and service discovery among themselves is realized largely automatically in IBIS-IP.

Discovery of the complete system is based on different basic principles, which are applied in IBIS-IP. These principles are briefly described below.

7.1 Basics

7.1.1 System Configuration must be available

The system configuration defines the following at a purely functional level:

- The devices participating in the IBIS-IP system. This information can generate an error message, if one of the devices cannot be reached.
- The services that shall run on which device in which IBIS-IP version. Using this information, SystemManagementService can ensure that the correct services are started in the devices.

Please note that it must not explicitly be configured in the system configuration that a `DeviceManagementService` (cf. chapter 9.1) is running on a device. The declaration of a device implies the presence of a `DeviceManagementService` (cf. chapter 3.1)

Example:

The system configuration specifies that the `CustomerInformationService` *(cf. chapter 9.2) should run in IBIS-IP version 1.0 on the OnBoardUnit 1 device (cf. chapter 3.6).*

The author of the configuration is responsible for the consistency of the configuration. If services or devices are missing in the configuration, or if IBIS-IP versions are declared for the services, no other service user can handle, the system will not function as planned. If contrary to the IBIS-IP specification, services with the same IBIS-IP version are started twice (e.g. on different devices), this can result in inconsistencies (also cf. chapter 7.1.2). Thus, such a state must be strictly avoided.

The system configuration is provided by the `SystemDocumentationService` (cf. chapter 9.9) via the **`GetSystemConfiguration`** operation. In this context, the format as well as the location of the system configuration data can be implemented manufacturer-specifically. However, the contents can only be addressed and changed via an interface of the `SystemDocumentationService`.

7.1.2 One Service only per Functionality

One service only may be started per functionality in the system in IBIS-IP. This avoids states, in which ambiguity occurs due to services existing multiple times.

Example:

If two `CustomerInformationServices` *would run in a system, it would not be clear (due to chapter 7.1.3), which passenger display gets its information from which of the two* `CustomerInformationServices`*. In particular, states can occur, where two displays show - at least temporarily - different information.*

This does not mean that there cannot be two services of the same type in an IBIS-IP system. However, they must assume different functional tasks in the IBIS-IP system, and must be addressable via different functional paths.

Example 1:

In an IBIS-IP system with 5 devices, five `DeviceManagementServices` *exist. However, they fulfill functionally different tasks (they manage different devices), and can be technically addressed via different paths.*

Example 2:

If three passenger counting devices are installed in a vehicle with three doors, each of these devices can provide a service for retrieving the passenger numbers. The devices meet functionally different tasks (they count passengers entering/exiting through the different doors), and can be technically addressed via different paths.

Example 3:

If an older passenger display in an IBIS-IP system can only handle data of the `CustomerInformationService` *of IBIS-IP version 1.0, but features of* `CustomerInformationService` *of IBIS-IP version 2.0 should be used at the same time on a modern passenger display, two* `CustomerInformationServices` *can be operated in the IBIS-IP system. The services fulfill functionally different tasks can be technically addressed via different paths. Different versions of a service are addressed on a system by using different ports or paths for the different versions of the service.*

Information about the devices present in an IBIS-IP and all required services in the required IBIS-IP versions are provided by the `SystemDocumentationService`.

The `SystemManagementServivce` plays a special role. It must only exist once within the complete IBIS-IP system (hence, also in exactly one IBIS-IP version only). This way, it is ensured that the parallel operation of two `SystemManagementServices` does not result in two parallel IBIS-IP systems.

7.1.3 Knowledge of the required Services

The system configuration does NOT define, which application and/or service should be connected with which service. The information, from which other services with which IBIS-IP version an application or service depends, is always included in the implementation of the application and/or service itself.

The service and/or application always connects with the service, which is found first with matching service name and matching IBIS-IP version via DNS-SD (cf. chapters 5.1 and 7.1.4.2). For this reason, it is important that the system configuration ensure that only one service exists per functionality.

If a service and/or application can handle the data of different IBIS-IP version of an offering service, the service and/or application using the data must define which version should be used and/or preferred.

Example:

The application on a passenger display must be implemented such that it always searches for a `CustomerInformationService` *in IBIS version 2.0 and retrieves and displays data from this service, as soon as the service is available to the application on the passenger display.*

7.1.4 Service and Device Identification in IBIS-IP

In the following it is described, how devices and services are identified in IBIS-IP.

In this context, it is differentiated between a functional and technical identification.

Functional Identification

The functional identification is used for system description and configuration. The focus is on a comprehensible representation of the system. Devices and services can be con-

figured using the functional identification. For this purpose, no information must be known about the specific technical implementation, i.e. IP addresses, ports or paths of the services.

Functional identification data is in particular exchanged at the time of system start between the *SystemManagementService* and the *SystemDocumentationService* in order to determine the system configuration of the IBIS-IP system and to start the correct services. Thus, functional identification data is relevant to the system and/or service start.

Technical Identification

Technical identification is used for IP-based communication with the services in IBIS-IP. Technical identification data can be determined per DNS-SD and is relevant for service use.

7.1.4.1 Devices

SystemManagementService assigns the functional identification of a device to the technical identification. *SystemManagementService* is the only service that monitors and manages devices (cf. chapter 9.10) and/or that assigns functional identifications to technical identifications.

All other services and applications only communicate with the services that can be addressed per DBS-SD based on their technical identification. This means in particular, without considering the devices, on which the services are running, or other functional identification data.

7.1.4.1.1 Functional Identification

A device is clearly identified across the IBIS-IP system using a

- Device type, and a
- Device ID

.

Possible device types are listed in section 3.4.

A device ID is a unique identification number that allows a conclusion regarding the installation location of the device. The combination of device type and device ID allows a clear functional identification across the system.

Example:

A validator located at installation position 3 can functionally be referred to as validator 3.

7.1.4.1.2 Technical Identification

A device can be technically identified within the network

- Using its system-wide unique IP address or
- Its system-wide unique DNS name.

7.1.4.2 Services

7.1.4.2.1 Functional Identification

A service is functionally uniquely identified across the entire system by

- Service name
- IBIS-IP version

- Device type, and
- Device ID.

The concept in particular allows that multiple versions of an IBIS-IP service are present on a device and operate in parallel. (Exception SystemManagementService, cf. chapter 9.10)

Example:

The `CustomerInformationService` *with IBIS-IP version 1.0 is clearly identified across the entire service in device class OnBoardUnit with device ID 1.*

7.1.4.2.2 Technical Identification

A service is technically uniquely identified across the entire system by

- The IP address unique across the system or the DNS name unique across the system of the device the service runs on,
- Service name,
- Ports, and
- (optional) Path, under which the service can be addressed.

Different versions of a service must be published by a device under different ports or different paths.

The protocol type (_ibisip_udp and/or _ibisip_http, cf. chapters 5.4 and 5.5) is not used for identification, as the protocol type directly results from the service and a service can only be of one of these two types.

Example 1:

Service `CustomerInformationService` *from 9.2 offers its operations of IBIS-IP version 1.0 under port 34567 and path "/1.0/". Operations of IBIS-IP version 2.0 are offered under port 34567 and path "/2.0/".*

`CustomerInformationService` *is always a _ibisip_http service.*

Example 2:

`CustomerInformationService` *from 9.2 is technically uniquely identified across the entire system by*

`<192.168.47.11:34567/1.0/CustomerInformationService>` *or*

`<vdv-test-dev:34567/1.0/CustomerInformationService>`

where `192.168.47.11` *is the IP address*, `vdv-test-dev` *the corresponding DNS name,* `34567` *the port*, `1.0` *the path, and* `CustomerInformationService` *the service name.*

7.2 The System Start Concept

The start of an IBIS-IP system is a two-stage process. The following sub-steps are mainly executed in both stages:

- Start services
- Publish services

- Connection establishment with the published services and execution of functional tasks

In the first stage, sub-step "Start services" is automatically executed, when the device is started. In this context, the system-relevant services of the vehicle operation functionalities (*DeviceManagementService*, *SystemManagementService* and *SystemDocumentationService*) are started only.

In the second stage, sub-step "Start services" is controlled by the running *SystemManagementService*.

7.2.1 Stage 1: Start of system-relevant Vehicle Operation functionalities

7.2.1.1 Start of the Services of the Vehicle Operation Functionalities

When the devices of an IBIS-IP system are started, service *DeviceManagementService* starts first on every device (cf. chapter 9.1).

For every one of these devices it is defined, whether further services of the vehicle operation functionalities (i.e. *SystemDocumentationService* and/or *SystemManagementService*) should be automatically started on the device. The definition that and possibly which services are automatically started, is provided by a device- and/or manufacturer- specific configuration, which is not standardized within IBIS-IP, but which can be retrieved and changed via an interface of the *DeviceManagementServices*.

The definition must ensure that the following situation occurs after all devices are started and all services of the vehicle operation functionalities to be automatically started are started:

- Exactly one *DeviceManagementService* runs on every device of the IBIS-IP system
- Exactly one *SystemDocumentationService* runs on exactly one of the devices of the IBIS-IP system. No further *SystemDocumentationService* exists in the IBIS-IP system (also cf. chapter 9.9).
- Exactly one *SystemManagementService* runs on exactly one of the devices of the IBIS-IP system. No further *SystemManagementService* exists in the IBIS-IP system (also cf. chapter 9.10).

Example:

In a system of one on-board computer and several displays, by the devices start all *DeviceManagementServices* *are started. The* *DeviceManagementService* *starts the* *SystemDocumentationService* *and* *SystemManagementService* *in addition on the on-board computer. However, for the time being, the* *DeviceManagementServices* *remain the only services running on the displays.*

(Remark: In Figure 1, chapter 7.3, the slightly more complex case is described, where *SystemDocumentationService* and *SystemManagementService* are operated on different devices.)

7.2.1.2 Publication of the Services of the Vehicle Operation Functionalities

All services in IBIS-IP use the publication mechanism via DNS-SD described in chapter 5.

With the publication, the following information regarding this service is announced across the system:

- Service name
- IBIS-IP version of the service
- Protocol, using which the service can be addressed (in the case of vehicle operation functionalities services, always `_ibisip_http`
- IP address and/or DNS name of the device, on which the service is running
- Port, under which the service must be reached
- Path, under which the service must be reached (as needed)

7.2.1.3 Establish Connection with the published Services of the Vehicle Operation Functionalities and Execution of functional Tasks

After the services are published, in particular the services of the *SystemManagementService* can be addressed. In this context, the three sub-steps described below are executed:

7.2.1.3.1 System Configuration Request

The *SystemManagementService* first requests the configuration of the IBIS-IP system from the *SystemDocumentationService* and receives a functional description of the system. This description defines, which device (identified via device type and device ID) should run which service (identified via service name and IBIS-IP version) (also cf. chapter 7.1.1).

7.2.1.3.2 DeviceManagementServices Request

Next, the SystemManagementService connects with all available *DeviceManagementServices*. Now, the *SystemManagementService* does not know yet, which *DeviceManagementService* is used on which device for communication. However, it knows their technical connection data based on the publication via DNS-SD.

The *SystemManagementService* requests from the *DeviceManagementServices* of the device in particular the following

- Device type of the corresponding device,
- Device ID, and
- Services available on the device.

7.2.1.3.3 Assignment of technical and functional Connection Data

The combination of the technical connection data from the publication of the functional information about device type and device ID allows the *SystemManagementService* to assign both sets of data with one another.

7.2.1.3.4 Start Commands to the DeviceManagementServices

Using this assignment, the *SystemManagementService* can technically correctly connect with the *DeviceManagementServices* of the individual devices according to the functional configuration data of the *SystemDocumentationService* in order to start further service on the respective devices.

If the *SystemManagementService* determines that services are not available on devices, where they should be running according to the configuration, the respective error state is stored by the *SystemManagementService*.

If the *SystemManagementService* determines that devices are not available, which should be available according to the configuration, the respective error state is stored by the *SystemManagementService*.

7.2.2 Stage 2: Start functional Services

7.2.2.1 Start further Services on the Devices

Once the *SystemManagementService* knows the system configuration and can assign and address all *DeviceManagementServices* of the involved devices, the *SystemManagementService* will be able to start further services of the IBIS-IP system on the devices.

For this purpose, the *SystemManagementService* calls up the **StartService** operation for the *DeviceManagementServices* of the involved devices. This way, all further services declared in the configuration are sequentially started.

7.2.2.2 Publish further Services

All services in IBIS-IP use the publication mechanism via DNS-SD described in chapter 5.

With the publication, the following information regarding these services is announced across the system:

For HTTP services

- Service name,
- IBIS-IP version of the service,
- Protocol, over which the service can be addressed,
- IP address and/or DNS name of the device, where the service is running,
- Port, under which the service must be reached,
- Path, under which the service must be reached (as needed),

For UDP services

- Service name,
- IBIS-IP version of the service,
- Protocol, over which the service can be addressed,
- IP address of the multicast group, over which the information of the service is distributed.

7.2.2.3 Establish Connection with the published Services and Execution of functional Tasks

After publishing all required and/or configured services, the information-consuming services and applications will set up the respective requests or the desired subscriptions to data.

In order to ensure the proper functioning of subscriptions to data (as data is only sent in the case of information changes, the data consumers normally do not know, whether a service is still functioning), it is recommended that services and applications that set up subscriptions to a service, are also informed about the current system state.

Example:

The application on a passenger display must be implemented such that it always searches for a `CustomerInformationService` in IBIS version 2.0 and retrieves and displays data from this service, as soon as the service is available to the application on the passenger display. In particular, it is not necessary to define this via a manual configuration

7.3 Example of a System Start in IBIS-IP

The system start of an IBIS-IP system for an example configuration is described, which involves two devices, device A and device B.

The following services are running on device A

- *`DeviceManagementService`*
- *`SystemManagementService`*
- *`Service X`*

The following services are running on device B

- *`DeviceManagementService`*
- *`SystemDocumentationService`*
- *`Service Y`*, where a subscription can be set up.

The *`SystemManagementService`* also regularly issues a status request to both *`DeviceManagementServices`* with respect to the service status on the devices and the device status (cf. chapter 9.1, 9.10).

In the example, system-relevant services of the vehicle operation functionalities are also started by the *`DeviceManagementService`* of the device. However, this is also possible differently (e.g. by script, application, or other autostart functions).

7.3.1 Stage 1: Start of system-relevant Vehicle Operation Functionalities

7.3.1.1 Start of the Services of the Vehicle Operation Functionalities

The *`DeviceManagementServices`* of both devices are automatically started via the device start.

On device A, the device-specific configuration ensures that the *`DeviceManagementService`* of device A also starts the *`SystemManagementService`*.

On device B, the device-specific configuration ensures that the *`DeviceManagementService`* of device B also starts the *`SystemDocumentationService`*.

7.3.1.2 Publication of the Services of the Vehicle Operation Functionalities

This stage is executed with a time offset. Every service publishes itself, as soon as it is operational. Thus, the *`DeviceManagementServices`* are published earlier than the other services of the vehicle operation functionalities.

7.3.1.3 Establish Connection with the published Services of the Vehicle Operation Functionalities and Execution of functional Tasks

The *`SystemManagementService`* is informed via DNS-SD, if a new service is added to the DNS-SD list of services. As soon as a *`SystemDocumentationService`* ap-

pears in this list, the *SystemManagementService* connects with this service and request the system configuration (in Figure 4, SDS/GetSystemConfiguration).

As soon as a *DeviceManagementService* appears in the list of services of the DNS-SD, the *SystemManagementService* connects with this service and retrieves device information (for the device class), device configuration (for the device ID), and service information (for the list of available services). This way, the SystemManagementService can

- Assign the functional identification (device class and device ID) to the technical identification (service addressing according to DNS-SD) and verify, whether
- All required services are available on the devices specified in the system configuration.

7.3.2 Stage 2: Start functional Services

7.3.2.1 Start of further Services

Service X on device A is started by the *SystemManagementService* by calling the service start operation of the *DeviceManagementService* of device A and providing the name of the service (cf. chapter 9.3.24). *Service Y* on device B is analogously started via the *DeviceManagementService* on device B.

7.3.2.2 Publish further Services

This stage is executed with a time offset. Every service is published, as soon as it is operational. Services *Service X* and *Service Y* are published, as soon as they are operational.

7.3.2.3 Establish Connection with the published Services and Execution of functional Tasks

The *SystemManagementService* periodically requests the status of all services and devices from the *DeviceManagementServices* of the devices.

In the example, *Service Y* also establishes a subscription to *SystemManagement-Service* in order to be continuously informed about changes in the system state.

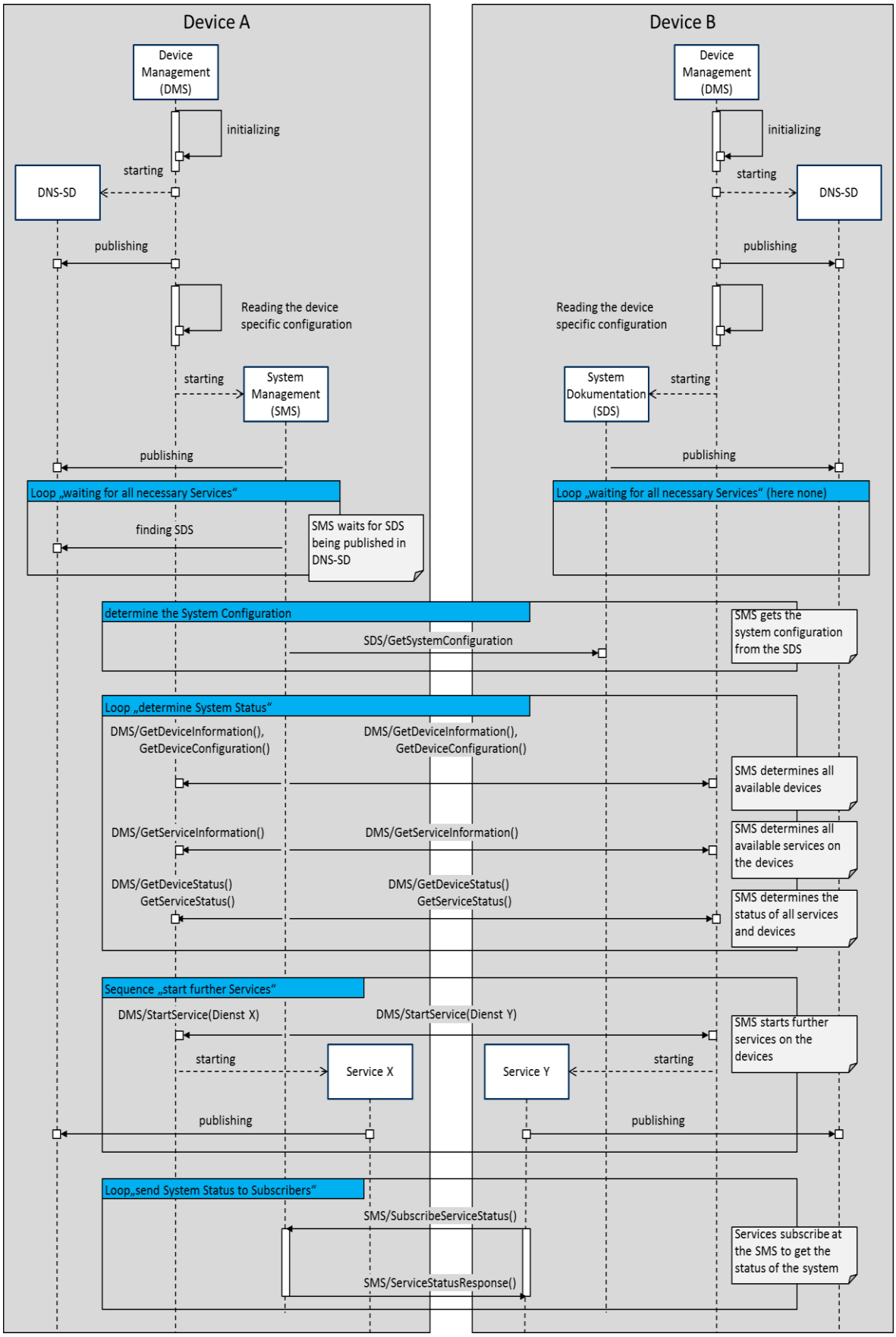

Figure 4: Example of the startup of an IBIS-IP system

8 Structuring of the information contents

In IBIS-IP, data contents are transferred in plain text. This means in particular that multiple information must not be coded and transferred on one data field. It must be respectively broken down into several data fields of a data structure.

The use of established and commonly used methods for the structured data transfer is important.

Thus, information contents are transferred in IBIS-IP with the help of XML data structures, and can be respectively validated using a XML schema (XSD = XML schema definition). In addition to the form represented here, VDV provides the XSD files of the specified services free of charge on their webpage.

8.1 Notation of XML Elements and Structures

The IBIS-IP interfaces presented in this document are defined with the help of a XML schema. Thus, the objects that are exchanged via the interface are available as XML elements. In this document, the XML elements are described in tabular form that originates from SIRI (CEN, TS 15531 Part 1). It is very compact, clear and offer numerous structural information, which is otherwise only visible on the XML schema definition. This chapter explains the notation of the table form used starting from chapter 9.

All names of elements, data types, and attributes are in English to prepare a possible standardization at the European level and to facilitate the exchange with European partners.

8.1.1 Representation of XML Elements in Text

A consistent notation of the XML elements should facilitate the provision of technically important information when reading.

- XML elements are written in upper-lower-case notation (Upper Camel Case), bold, and italic, e.g. ***VehicleJourneyRef***. Whenever possible and meaningful, the element names are based on terms from TransModel. If there is no suitable term available in TransModel for a concept or object, it was attempted to apply the respective term from JourneyWeb or the suitable concept from DELFI.
- Data types are written in italic, e.g. *xsd:boolean****.***
- Code examples are presented in smaller font.

8.1.2 Table Notations for Operations

Beyond that, operations of a service are describe in a table of the following form.

Operation	**Request/ Response**	**Data type, data structure used**
GetData	Req.	-
	Resp.	BeaconLocationService. GetDataResponseStructure
SubscribeData	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeData	Req.	TerminateSubscribeRequestStructure

	Resp.	TerminateSubscribeResponseStructure

Table 5: Example of an operation notation in IBIS-IP

8.1.3 Table Notation of XML Structures

In this document, XML structures are represented in a table notation (cf. Table 6). There is a separate table for every important service request/response element. Further tables are specified for all important child elements, which are used in structures that are more complex. In order to save space, the column headers are only shown in the example in Table 6. They will not be repeated in any of the following tables. A consistent set of rules is used in the tables in order to describe the XML elements and the associated conditions.

Grouping	Element name	Min : Max	Data type	Explanation
ContinuousServiceStructure			*+Structure*	A passenger movement with the help of a continuous means of transportation not bound to a schedule.
	a ***ContinuousMode***	**-1:1**	*walk | parkAndRide | demandResponsive*	Modality for continuous traffic
	b ***IndividualMode***		*walk | cycle | taxi | self-drive-car | others-drive-car | motorcycle | truck*	Modality of means of traffic for individual traffic
DatedService	***OperatingDay***	**1:1**	*→OperatingDay*	Operating day of the journey
	VehicleRef	0:1	*→Vehicle*	Vehicle ID.
ServiceJourney	***JourneyRef***	**1:1**	*→Journey*	Journey ID. .
LineIdentity	***LineRef***	**1:1**	*→Line*	Line ID. .
	DirectionRef	**1:1**	*→Direction*	Direction ID.
Service	***Mode***	**1:1**	*+Mode*	Type of means of transportation. .
	PublishedLineName	**1:1**	*InternationalText*	Line number or name as known by the public.
	OperatorRef	0:1	*→Operator*	Operator ID. .
	RouteDescription	0:1	*InternationalText*	Description of the route.
	Via	0:*	*+ServiceViaPoint*	Important stops on the route.
	Attribute	0:*	*+GeneralAttribute*	Notes and attributes (with classifications) regarding the journey.
ServiceOrigin	*OriginStopPointRef*	0:1	*→StopPoint*	ID of the first stop point of the journey, i.e. point of departure.
	OriginText	0:1	*InternationalText*	Name of the first stop point of the journey, i.e. point of departure.

Table 6: Example (for chapter 9 for the tabularized notation of a XML structure)

8.1.3.1 Grouping

There is sometimes an identifier in the first column, which classifies the elements into meaningful groupings, e.g. *Service* or *ServiceOrigin*. This is purely used for documenta-

tion purposes only. In most cases, this corresponds to the names of a XML group used in the XML schema. Groupings are used to organize the elements and thus, to provide more clarity and better reusability.

8.1.3.2 Element Name

Element names are shown in italic in the second column, e.g. *OperatingDay*. If the element is a mandatory element, it is printed in **bold**. Optional elements are not printed in bold. The name of the structure can be found in the top left of the table.

Elements, which are derived (XML" “derived by extension”) or used anonymously, are marked in the name field with three colons “:::”.

8.1.3.3 Multiplicity & Choice (Min:Max)

The conditions, whether an element is mandatory or optional, or whether it can occur one or several times within the superordinate element, are indicated in the third column Min:Max. In this context, the common "min:max:" UML conventions are used. E.g. "0:1" stands for an optional, simple element, "**1:1**" indicates a mandatory, simple element. "0:*" stands for an optional, multiple element etc. Mandatory elements are printed in **bold**.

In some cases, an element must be selected from its set (XML choice). This is indicated by a prefixed minus sign, e.g. "**-1:1**". In this case, the element name is prefixed by a lower case letter indicating the list of selection possibilities. In the case of optional selection options (choices), a zero is contained in the min value: "-0:1".

8.1.3.4 Data Type

The data types are indicated in italic in the fourth column, e.g. *InternationalText*. If the namespace deviates from the standard XML namespace, it is indicated as well, e.g. "*xs:dateTime*" or "*siri:PtSituationElement*".

- A complex data type, which contains structures as child elements, is marked with "+*Structure"* in the Data type column.
- If elements are used as references (foreign keys) to other objects, the type of the referenced element with a prefixed arrow is used as data type. For example, "→*StopPoint*" as type of a reference (*StopPointRefStructure*) to an object of type "StopPointType".
- In the most cases, enumerated types are immediately represented with the usable values, e.g. "*walk* | *cycle*". A type is only declared and referenced in some cases with very extensive enumerations, which are used at several locations.
- In order to save space, abbreviations are used for the data types. E.g., the endings "Structure" and "Type" are always omitted. E.g., "*InternationalText*" is always used instead of "*InternationalTextStructure*".

8.1.3.5 Explanation

The last column contains for all elements an explanation of their purpose. In many cases, it is referenced to other locations in the text. E.g. in the case of complex child elements, the location is referenced, where their table definition can be found. In some cases, the explanation is too extensive and would explode the tabular form. In these cases, the remarks can be found in the text below the table.

A XSD file with the contents in digital form exists as supplement to this document. If not enclosed with this document, this file can be downloaded under www.vdv.de/ip-kom-oev.aspx.

9 Services Specification

In the following sections of this chapter the services of IBIS-IP, their operations and corresponding data structures are described in detail.

Services based on IP-KOM

Note: Display, MMI and Announcement are from our point of view just service consumer, they do not provide an active and functional service in the network.

9.1 BeaconLocationService

The service *BeaconLocationService* is one part of the implementation of the functional component *Physical Locating*.

It is used to transmit the information of a registered beacon signal to the IBIS-IP system. Within the IBIS-IP system there are other services e.g. representing the *Network Locating*, which associate the beacon information to a specific location in the public transport network.

Most of the information related to the physical locating change cyclically (e.g. GNNS data at least once per second, cf. chapter 9.5). For these information the transmission via UDP is favoured.

In contrast, the determination of the location of a beacon happens only at some specific point in a public transport network. The reliable transmission is of that information is of higher relevance than the speed of the transmission. Therefore the *BeaconLocationService* is designed as an http-service.

9.1.1 Operations of BeaconLocationService

The *BeaconLocationService* can be requested over in Table 7 described Operations. The data structures an then presented in chapter 9.1.2ff.

Operation	**Request/ Response**	**Used Datatype, Datastructure**
GetData	Req.	-
	Resp.	BeaconLocationService. GetDataResponseStructure
SubscribeData	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeData	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure

Table 7: Operations of BeaconLocationService

9.1.2 Data Structures of GetData Operation

9.1.2.1 Request

Because of being a **Get** operation, there is no request structure for this operation.

9.1.2.2 Response

BeaconLocationService.GetDataResponse				*+Structure*	**Response structure of BeaconLocationService**
				choice	One of the following structures
	a	***Data***	**–1:1**	*+BeaconLocation-Ser-vice.DataContent*	Detailed response structure
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 8: Description of BeaconLocationService.GetDataResponse

BeaconLocationService.DataContent			*+Structure*	**Structure which describes the data content of the BeaconLocationService**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response Time stamp
	BeaconCode	**1:1**	*IBIS-IP.NMTOKEN*	Beacon-Code content
	BeaconTime	0:1	*IBIS-IP.time*	Beacon passing moment
	BeaconDistance	**1:1**	*IBIS-IP.double*	Information about the counter state of the odometer in the passing moment (value in meter)

Table 9: Description of BeaconLocationService.DataContent

9.1.3 Data Structure of SubscribeData Operation

For the subscription the data structures from chapters 10.51 and 10.52 are used.

9.1.4 Data Structures of UnsubscribeData Operation

To terminate a subscription are the structures from chapters 10.57 and 10.58 used.

9.2 CustomerInformationService

This service represents the functional component Customer Information Determination. It is the central source of information for all tasks of the passenger information by providing consistent information to all stakeholders in the system.

The data of the CustomerInformationService do not change cyclically. Therefore it is designed as an http-service.

9.2.1 Operations of CustomerInformationService

Operation	**Request/ Response**	**Used Datatype, Datastructure**
GetAllData	Req.	-
	Resp.	CustomerInformationService. GetAllDataResponseStructure
SubscribeAllData	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeAllData	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure

Operation	**Request/ Response**	**Used Datatype, Datastructure**
GetCurrentAnnouncement	Req.	-
	Resp.	CustomerInformationService. GetCurrentAnnouncementResponseStructure
SubscribeCurrentAnnouncement	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeCurrentAnnouncement	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetCurrentConnectionInformation	Req.	-
	Resp.	CustomerInformationService. GetCurrentConnectionInformationResponseStructure
SubscribeCurrentConnectionInformation	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeCurrentConnectionInfor-mation	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetCurrentDisplayContent	Req.	-
	Resp.	CustomerInformationService. GetCurrentDisplayContentResponseStructure
SubscribeCurrentDisplayContent	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeCurrentDisplayContent	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetCurrentStopPoint	Req.	-
	Resp.	CustomerInformationService. GetCurrentStopPointResponseStructure
SubscribeCurrentStopPoint	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeCurrentStopPoint	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetCurrentStopIndex	Req.	-
	Resp.	CustomerInformationService. GetCurrentStopIndexResponseStructure
SubscribeCurrentStopIndex	Req.	SubscribeRequestStructure

Operation	Request/ Response	Used Datatype, Datastructure
	Resp.	SubscribeResponseStructure
UnsubscribeCurrentStopIndex	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetTripData	Req.	-
	Resp.	CustomerInformationService. GetTripDataResponseStructure
SubscribeTripData	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeTripData	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetVehicleData	Req.	-
	Resp.	CustomerInformationService. GetVehicleDataResponseStructure
SubscribeVehicleData	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeVehicleData	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
RetrievePartialStopSequence	Req.	CustomerInformationService. RetrievePartialStopSequenceRequestStructure
	Resp.	CustomerInformationService. RetrievePartialStopSequenceResponseStructure

Table 10: Description of CustomerInformationService Operations

9.2.2 Data Structure of GetAllData Operation

9.2.2.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.2.2.2 Response

CustomerInformation-Service.GetAllDataResponse				*+Structure*	**Response structure for the GetAllData request**
				choice	One of the following structures
	a	***AllData***	**–1:1**	*+CustomerInformation-Service.AllData*	Detailed response (cf. table below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 11: Description of CustomerInformationService.GetAllDataResponse

CustomerInformationService.AllData			*+Structure*	**Response structure for all data of the CustomerInformationService**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	VehicleRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference to the vehicle-ID
	DefaultLanguage	**1:1**	*IBIS-IP.language*	Definition of the default language
	TripInformation	**1:2**	*+TripInformation*	Trip information (cf. also chapter 10.54)
	CurrentStopIndex	1:1	*IBIS-IP.int*	Index of the current stop
VehicleInformationGroup	***RouteDeviation***	1:1	*RouteDeviationEnumeration*	Information, if there exists a deviation of the planned route (cf. also chapter 12.18)
	DoorState	0:1	*DoorOpenStateEnumeration*	Information about the door state (cf. also chapter 12.7)
	InPanic	0:1	*IBIS-IP.boolean*	Information about the panic button state
	VehicleStopRequested	0:1	*IBIS-IP.boolean*	Stop request status information
	ExitSide	0:1	*ExitSideEnumeration*	Exit side information (cf. also chapter 12.10)
	MovingDirectionForward	0:1	*IBIS-IP.boolean*	Information about the moving direction
	VehicleMode	0:1	*VehicleModeEnumeration*	Vehicle mode information (cf. also chapter 12.24)

Table 12: Description ofCustomerInformationService.AllData

9.2.3 Data Structures of SubscribeAllData Operation

For the subscription the data structures from chapters 10.51 and 10.52 are used.

9.2.4 Data Structures of UnsubscribeAllData Operation

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.2.5 Data Structure of GetCurrentAnnouncement Operation

9.2.5.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.2.5.2 Response

CustomerInformation-Service.GetCurrentAnnouncementResponse				*+Structure*	**Response structure for the current announcement request**
				Choice	One of the following structures
	a	***CurrentAnnouncementData***	**–1:1**	*+CustomerInformation-Service.CurrentAnnouncementData*	Detailed response structure (cf. below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 13: Description of CustomerInformationService.GetCurrentAnnouncementResponse

CustomerInformation-Service.CurrentAnnouncementData			*+Structure*	**Detailed Response structure for the current announcement request**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	CurrentAnnouncement	**1:1**	*+Announcement*	Announcement information (cf. chapter 10.2)

Table 14: Description of CustomerInformationService.CurrentAnnouncementData

9.2.6 Data Structure of SubscribeCurrentAnnouncement Operation

For the subscription the data structures from chapters 10.51 and 10.52 are used.

9.2.7 Data Structure of UnsubscribeCurrentAnnouncement Operation

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.2.8 Data Structure of GetCurrentConnectionInformation Operation

9.2.8.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.2.8.2 Response

CustomerInformation-Service.GetCurrentConnectionResponse				*+Structure*	**Response structure for information at current connection**
				Choice	One of the structures below
	a	***CurrentConnectionData***	**–1:1**	*+CustomerInformation-Service.CurrentConnectionData*	Detailed response structure (cf. below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 15: Description von CustomerInformationService.GetCurrentConnectionResponse

CustomerInformation-Service.CurrentConnectionData			*+Structure*	**Response structure for the request of the current connection**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	CurrentConnection	**1:1**	*+Connection*	Information about the connection (cf. chapter 10.8)

Table 16: Description of CustomerInformationService.CurrentConnectionData

9.2.9 Data Structure of SubscribeCurrentConnectionInformation Operation

For the subscription the data structures from chapters 10.51 and 10.52 are used.

9.2.10 Data Structure of UnsubscribeCurrentConnectionInformation Operation

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.2.11 Data Structure of GetCurrentDisplayContent Operation

9.2.11.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.2.11.2 Response

CustomerInformation-Ser-vice.GetCurrentDisplayContentResponse				*+Structure*	**Structure with the response with the current display content**
				Choice	One of the structures below
	a	***CurrentDisplayCon-tentData***	**–1:1**	*+CustomerInf ormation-Ser-vice.CurrentDi splayCon-tentData*	detailed response structure (cf. below)
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 17: Description of CustomerInformationService.GetCurrentDisplayContentResponse

CustomerInformation-Service.CurrentDisplayContentData			*+Structure*	**Response structure with the current data content for displays**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	CurrentDisplayContent	**1:1**	*+DisplayCont ent*	Data content structure for displays (cf. chapter 10.19)

Table 18: Description of CustomerInformationService.CurrentDisplayContentData

9.2.12 Data Structure of SubscribeCurrentDisplayContent Operation

For the subscription the data structures from chapters 10.51 and 10.52 are used.

9.2.13 Data Structure of UnsubscribeCurrentDisplayContent Operation

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.2.14 Data Structure of GetCurrentStopPoint Operation

9.2.14.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.2.14.2 Response

CustomerInformation-Service.GetCurrentStopPointResponse				*+Structure*	**Response structure with information about the current stop point**
				Choice	One of the structures below
	a	***CurrentStopPoint-Data***	**–1:1**	*+CustomerInf ormation-Ser-vice.CurrentSt opPointData*	detailed response structure (cf. below)
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 19: Description of CustomerInformationService.GetCurrentStopPointResponse

CustomerInformation-Service.CurrentStopPointData			*+Structure*	**Response structure which describes the current stop point data**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	CurrentStopPoint	**1:1**	*+StopInformat ion*	Information to the current stop point (cf. chapter 10.48)

Table 20: Description of CustomerInformationService.CurrentStopPointData

9.2.15 Data Structure of SubscribeCurrentStopPoint Operation

For the subscription the data structures from chapters 10.51 and 10.52 are used.

9.2.16 Data Structure of Operation UnsubscribeCurrentStopPoint

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.2.17 Data Structure of Operation GetCurrentStopIndex

9.2.17.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.2.17.2 Response

CustomerInformation-Service.GetCurrentStopIndexResponse				*+Structure*	**Response structure for the current stop point index**
				Choice	One of the structures below
	a	***CurrentStop-IndexData***	**–1:1**	*+CustomerInformation-Ser-vice.CurrentStopIndexData*	detailed response structure (cf. below)
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 21: Description of CustomerInformationService.GetCurrentStopIndexResponse

CustomerInformation-Service.CurrentStopIndexData			*+Structure*	**Response structure with the data content of the current stop index**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	CurrentStopIndex	**1:1**	*IBIS-IP.int*	Sto index information

Table 22: Description of CustomerInformationService.CurrentStopIndexData

9.2.18 Data Structure of Operation SubscribeCurrentStopIndex

For the subscription the data structures from chapters 10.51 and 10.52 are used.

9.2.19 Data Structure of Operation UnsubscribeCurrentStopIndex

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.2.20 Data Structure of Operation GetTripData

9.2.20.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.2.20.2 Response

CustomerInformation-Service.GetTripDataResponse				*+Structure*	**Response Structure with the current trip data**
				Choice	One of the structures below
	a	***TripData***	**–1:1**	*+CustomerInformation-Ser-vice.TripData*	detailed response structure (cf. below)
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 23: Description of CustomerInformationService.GetTripDataResponse

CustomerInformationService.TripData			*+Structure*	**Response structure with trip data content**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	VehicleRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference at vehicle ID
	DefaultLanguage	**1:1**	*IBIS-IP.language*	Information about the default language
	TripInformation	**1:1**	*+TripInformation*	Trip information (cf. chapter 10.54)
	CurrentStopIndex	**1:1**	*IBIS-IP.int*	Information for the current stop index

Table 24: Description of CustomerInformationService.TripData

9.2.21 Data Structure of SubscribeTripData Operation

For the subscription the data structures from chapters 10.51 and 10.52 are used.

9.2.22 Data Structure of Operation UnsubscribeTripData

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.2.23 Data Structure of Operation GetVehicleData

9.2.23.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.2.23.2 Response

CustomerInformation-Service.GetVehicleDataResponse				*+Structure*	**Response structure with vehicle data**
				Choice	One of the structures below
	a	***VehicleData***	**–1:1**	*+CustomerInformation-Service.VehicleData*	detailed response structure (cf. below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 25: Description of CustomerInformationService.GetVehicleDataResponse

CustomerInformationService.VehicleData			*+Structure*	**Structure with vehicle data content**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	VehicleRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference at a specific vehicle-ID
VehicleInformationGroup	***RouteDeviation***	1:1	*RouteDeviationEnumeration*	Information, if there is a route deviation (cf. chapter 12.18)
	DoorState	0:1	*DoorOpenStateEnumeration*	Information about the door state (cf. chapter 12.7)
	InPanic	0:1	*IBIS-IP.boolean*	In panic status information
	VehicleStopRequested	0:1	*IBIS-IP.boolean*	Stop request status information
	ExitSide	0:1	*ExitSideEnumeration*	Exit side information (cf. chapter 12.10)
	MovingDirectionForward	0:1	*IBIS-IP.boolean*	Information about the moving direction
	VehicleMode	0:1	*VehicleModeEnumeration*	Information about the vehicle mode (cf. chapter 12.24)

Table 26: Description of CustomerInformationService.VehicleData

9.2.24 Data Structure of Operation SubscribeVehicleData

For the subscription the data structures from chapters 10.51 and 10.52 are used.

9.2.25 Data Structure of Operation UnsubscribeVehicleData

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.2.26 Data Structure of Operation RetrievePartialStopSequence

9.2.26.1 Request

CustomerInformation-Service.RetrievePartialStopRequest			*+Structure*	**Request structure for an defined part of the stop point sequence**
	StartingStopIndex	**1:1**	*IBIS-IP.int*	First stop index of the index queue
	NumberOfStopPoints	**1:1**	*IBIS-IP.int*	Requested number of stop points

Table 27: Description of CustomerInformationService.RetrievePartialStopRequest

9.2.26.2 Response

CustomerInformation-Service.RetrievePartialStopSequenceResponse				*+Structure*	**Response structure for a partial stop point sequence request**
				Choice	One of the structures below
	a	***PartialStopSequence-Data***	**–1:1**	*+CustomerInformation-Service.PartialStopSequence-Data*	detailed response structure (cf. below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 28: Description of CustomerInformationService.RetrievePartialStopSequenceResponse

CustomerInformation-Service.PartialStopSequenceData			*+Structure*	**Response structure with the detailed data content for a partial stop index request**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	StopSequence	**1:1**	*+StopSequence*	Information for a defined stop sequence (cf. chapter 10.50)

Table 29: Description of CustomerInformationService.PartialStopSequenceData

9.3 DeviceManagementService

The `DeviceManagementService` is a service which is used to maintain and manage the plugged devices.

As described in chapter 7.2.1 a DeviceManagementService has to run on each device participating in the IBIS-IP network.

The `DeviceManagementService` is designed as an http-service.

9.3.1 Operations of DeviceManagementService

The `DeviceManagementService` Operations are defined in tasks dealing on the one side with the device functions like

- Request static information about the device
- Change the device configuration
- Request the device configuration
- Restart, deactivate and activate the device
- Request the current device status
- Request error codes from the device

and on the other side with functions for Services on the device like

- Static information about the services
- Start, stop and restart services
- Request the state of the services

Operation	**Request/ Response**	**Used Datatype, Data structure**
GetDeviceInformation	Req.	-
	Resp.	DeviceManagementService. GetDeviceInformationResponseStructure
SubscribeDeviceInformation	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeDeviceInformation	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetDeviceConfiguration	Req.	-
	Resp.	DeviceManagementService. GetDeviceConfigurationResponseStructure
SetDeviceConfiguration	Req.	DeviceManagementService. SetDeviceConfigurationRequestStructure

	Resp.	DataAcceptedResponseStructure
SubscribeDeviceConfiguration	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeDeviceConfiguration	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetDeviceStatus	Req.	-
	Resp.	DeviceManagementService. GetDeviceStatusResponseStructure
SubscribeDeviceStatus	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeDeviceStatus	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetDeviceErrorMessages	Req.	-
	Resp.	DeviceManagementService. GetDeviceErrorMessagesResponseStructure
SubscribeDeviceErrorMessages	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeDeviceErrorMessag-es	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
RestartDevice	Req.	-
	Resp.	DataAcceptedResponseStructure
DeactivateDevice	Req.	-
	Resp.	DataAcceptedResponseStructure
ActivateDevice	Req.	-
	Resp.	DataAcceptedResponseStructure
GetServiceInformation	Req.	-
	Resp.	DeviceManagementService. GetServiceInformationResponseStructure
SubscribeServiceInformation	Req.	SubscribeRequestStructure;
	Resp.	SubscribeResponseStructure
UnsubscribeServiceInformation	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetServiceStatus	Req.	-

	Resp.	DeviceManagementService. GetServiceStatusResponseStructure
SubscribeServiceStatus	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeServiceStatus	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
StartService	Req.	DeviceManagementService. StartServiceRequestStructure
	Resp.	DataAcceptedResponseStructure
StopService	Req.	DeviceManagementService. StopServiceRequestStructure
	Resp.	DataAcceptedResponseStructure
RestartService	Req.	DeviceManagementService. RestartServiceRequestStructure
	Resp.	DataAcceptedResponseStructure

Table 30: Description of Operationen des DeviceManagementService

9.3.2 Data Structure of Operation GetDeviceInformation

The operation GetDeviceInformation returns static device information like serial number, manufacturer etc.

9.3.2.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.3.2.2 Response

DeviceManagementService.GetDeviceInformationResponse				*+Structure*	**Request structure with non configuration able device information**
				choice	One of the choices below
	a	***DeviceManagementService.GetDeviceInformationResponseData***	**–1:1**	*+DeviceManagementService.GetDeviceInformationResponseData*	Detailed request structure with the non configurable device parameters (cf. table below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 31: Description of DeviceManagementService.GetDeviceInformationResponse

DeviceManagementService.GetDeviceInformationResponseData			*+Structure*	**Detailed response structure with static device settings**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	DeviceInformation	**1:1**	*+DeviceInformationStructure*	Detailed response structure (cf. chapter 10.14)

Table 32: Description of DeviceManagementService.GetDeviceInformationResponseData

9.3.3 Data Structure of Operation SubscribeDeviceInformation

For the subscription the data structures from chapters 10.51 and 10.52 are used.

9.3.4 Data Structure of OperationUnsubscribeDeviceInformation

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.3.5 Data Structure of Operation GetDeviceConfiguration

The **`GetDeviceConfiguration`** operation enables to set the single variable parameter of a device. This parameter is the device –ID, which is used as a reference to the plugged position inside the vehicle

9.3.5.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.3.5.2 Response

DeviceManagementService.GetDeviceConfigurationResponse				*+Structure*	**Response structure with device configuration data content**
				choice	One of the structures below
	a	***DeviceManagementService.GetDeviceConfigurationResponseData***	**–1:1**	*+DeviceManagementService.GetDeviceConfigurationResponseData*	Detailed response structure with the device configuration (cf. below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 33: Description of DeviceManagementService.GetDeviceConfigurationResponse

DeviceManagementService.GetDeviceConfigurationResponseData			*+Structure*	**Detailed response structure with the device configuration**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	DeviceID	**1:1**	*IBIS-IP.int*	Device-ID (device plug-in position)

Table 34: Description of DeviceManagementService.GetDeviceConfigurationResponseData

9.3.6 Data Structure of Operation SetDeviceConfiguration

With this operation it is possible to set the device-ID. This setting happens just the first time after the device is plugged to the system and is stored in the device configuration file (proprietary). This fall back solution is needed when the device is not able to set this ID based on certain automatic parameters.

9.3.6.1 Request

DeviceManagementService.SetDeviceConfigurationRequest			*+Structure*	**Request structure which enables the setting of the device ID in the device configuration**
	DeviceID	**1:1**	*IBIS-IP.int*	Value of the device-ID

Table 35: Description of DeviceManagementService.SetDeviceConfigurationRequest

9.3.6.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. chapter 10.9) is used.

9.3.7 Data Structure of Operation SubscribeDeviceConfiguration

For the subscription the data structures from chapters 10.51 and 10.52 are used.

9.3.8 Data Structure of Operation UnsubscribeDeviceConfiguration

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.3.9 Data Structure of Operation GetDeviceStatus

9.3.9.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.3.9.2 Response

DeviceManagementService.GetDeviceStatusResponse				*+Structure*	**Response structure with the device status**
				choice	One of the structures below
	a	***DeviceManagementService.GetDeviceStatusResponseData***	**–1:1**	*+DeviceManagementService.GetDeviceStatusResponseData*	Detailed response structure with the device status information (cf. below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 36: Description of DeviceManagementService.GetDeviceStatusResponse

DeviceManagementService.GetDeviceStatusResponseData			*+Structure*	**Detailed response structure with the device status information**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	DeviceState	**1:1**	*+ DeviceStateEnumeration*	Device status (cf. chapter 12.3)

Table 37: Description of DeviceManagementService.GetDeviceStatusResponseData

9.3.10 Data Structure of Operation SubscribeDeviceStatus

For this subscription the data structures from chapters 10.51 and 10.52 are used.

9.3.11 Data Structure of Operation UnsubscribeDeviceStatus

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.3.12 Data Structure of Operation GetDeviceErrorMessages

9.3.12.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.3.12.2 Response

DeviceManagementService.GetDeviceErrorMessagesResponse				*+Structure*	**Response structure for device error messages**
				choice	One of the choices below
	a	***DeviceManagementService.GetDeviceErrorMessagesResponseData***	**–1:1**	*+DeviceManagementService.GetDeviceErrorMessagesResponseData*	Detailed response structure for device errors
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 38: Description of DeviceManagementService.GetDeviceErrorMessagesResponse

DeviceManagementSer-vice.GetDeviceErrorMessagesResponseData			*+Structure*	**Detailed response structure for device errors**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	ErrorMessage	**10:***	*+ Message*	Error message

Table 39: Description of DeviceManagementService.GetDeviceErrorMessagesResponseData

9.3.13 Data Structure of Operation SubscribeDeviceErrorMessages

For this subscription the data structures from chapters 10.51 and 10.52 are used.

9.3.14 Data Structure of Operation UnsubscribeDeviceErrorMessages

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.3.15 Data Structure of Operation RestartDevice

9.3.15.1 Request

With the request **`RestartDevice`** is no data transmitted.

9.3.15.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. chapter 10.9) is used.

9.3.16 Data Structure of Operation DeActivateDevice

9.3.16.1 Request

With the request **`DeActivateDevice`** is no data transmitted.

9.3.16.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. chapter 10.9) is used.

9.3.17 Data Structure of Operation ActivateDevice

9.3.17.1 Request

With the request **`ActivateDevice`** is no data transmitted.

9.3.17.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. chapter 10.9) is used.

9.3.18 Data Structure of Operation GetServiceInformation

9.3.18.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.3.18.2 Response

DeviceManagementService.GetServiceInformationResponse				*+Structure*	**Response structure with information about the available services at the device**
				choice	One of the choices below
	a	***DeviceManagementService.GetServiceInformationResponseData***	**–1:1**	*+DeviceManagementService.GetServiceInformationResponseData*	Detailed response structure with information about the available services (cf. below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 40: Description of DeviceManagementService.GetServiceInformationResponse

DeviceManagementService.GetServiceInformationResponseData			*+Structure*	**Detailed response structure with information about the available services**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	ServiceInformationList	**1:1**	*+ServiceInformationList*	List of available services (cf. chapter 0)

Table 41: Description of DeviceManagementService.GetServiceInformationResponseData

9.3.19 Data Structure of Operation SubscribeServiceInformation

For this subscription the data structures from chapters 10.51 and 10.52 are used.

9.3.20 Data Structure of Operation UnsubscribeServiceInformation

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.3.21 Data Structure of Operation GetServiceStatus

9.3.21.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.3.21.2 Response

DeviceManagementService.GetServiceStatusResponse				*+Structure*	**Response structure with status information about the services located at the device**
				choice	One of the choices below
	a	***DeviceManagementService.GetServiceStatusResponseData***	**–1:1**	*+DeviceManagementService.GetServiceStatusResponseData*	Detailed response structure with the status of services running on the device (cf. below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 42: Description of DeviceManagementService.GetServiceStatusResponse

DeviceManagementService.GetServiceStatusResponseData			*+Structure*	**Detailed response structure with the services located at the device**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	ServiceSpecificationWithStateList	**1:1**	*+ ServiceSpecificationWithStateList*	Service list including the status (cf. chapter 10.44)

Table 43: Description of DeviceManagementService.GetServiceStatusResponseData

9.3.22 Data Structure of Operation SubscribeServiceStatus

For this subscription the data structures from chapters 10.51 and 10.52 are used.

9.3.23 Data Structure of Operation UnsubscribeServiceStatus

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.3.24 Data Structure of Operation StartService

9.3.24.1 Request

DeviceManagementSer-vice.StartServiceRequestStructure			*+Structure*	**Request structure which enables to start one specific (in the ServiceSpecification specified) service**
	ServiceSpecification	**1:1**	*+ServiceSpecification*	Reference on the services to be started (cf. chapter 10.37)

Table 44: Description of DeviceManagementService.StartServiceRequestStructure

9.3.24.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. chapter 10.9) is used.

9.3.25 Data Structure of Operation StopService

9.3.25.1 Request

DeviceManagementSer-vice.StopServiceRequestStructure			*+Structure*	**Request structure which enables to stop one specific (in the ServiceSpecification specified) service.**
	ServiceSpecification	**1:1**	*+ServiceSpecification*	Reference at the stopped services (cf. chapter 10.37)

Table 45: Description of DeviceManagementService.StopServiceRequestStructure

9.3.25.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. chapter 10.9) is used.

9.3.26 Data Structure of Operation RestartService

9.3.26.1 Request

DeviceManagementSer-vice.RestartServiceRequestStructure			*+Structure*	**Request structure which enables to restart one specific (in the ServiceSpecification specified) service.**
	ServiceSpecification	**1:1**	*+ServiceSpecification*	Reference at the restarted services (cf. chapter 10.37)

Table 46: Description of DeviceManagementService.RestartServiceRequestStructure

9.3.26.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. chapter 10.9) is used.

9.4 DistanceLocationService

The `DistanceLocationService` is in IBIS-IP responsible for the interpretation and conversion of odometer pulses from the vehicle interface into covered distance. The service is therefore part of the physical locating functional component. Because of the very short validity of a distance value, this service is provided as a UDP service.

DistanceLocationService.Data			*+Structure*	**Structure for description of the DistanceLocationService information**
	Distance	**1:1**	*IBIS-IP.double*	Covered distance in [m]
	Odometer-Pulses	0:1	*IBIS-IP.int*	Number of odometer pulses

Table 47: Description of DistanceLocationService.Data

9.5 Dienst GNSSLocationService

For the distribution of information which describe the location of the vehicle in coordinates, IBIS-IP provides the `GNSSLocationService` as part of the physical locating functional component. To be open for future development it was intentionally refused to encapsulate the NMEA telegram into XML, instead it was split into values according to their meaning. Since as a source of this data usually a satellite receiver is used which provides the locating information periodically, this service is provided as a UDP service.

GNSSLocationService.Data			*+Structure*	**Structure which describes the information of the GNSSLocationService**
	latitude	**1:1**	*+GNSSCoordinate*	Latitude value in a structure (cf. chapter 10.29)
	longitude	**1:1**	*+GNSSCoordinate*	Longitude value in a structure (cf. chapter 10.29)
	altitude	0:1	*IBIS-IP.double*	Altitude value above sea level
	time	0:1	*IBIS-IP.time*	Time value
	date	0:1	*IBIS-IP.date*	Date value
	SpeedOverGround	0:1	*IBIS-IP.double*	GNSS based speed over ground
	SignalQuality	0:1	*GNSSQualityEnumeration*	Signal quality value (cf. chapter 12.12)
	NumberOfSatellites	0:1	*IBIS-IP.int*	Number of satellites which are used for the GNSS calculation
	HorizontalDilutionOfPrecision	0:1	*IBIS-IP.double*	Value of precision of results in horizontal direction
	VerticalDilutionOfPrecision	0:1	*IBIS-IP.double*	Value of precision of results in vertical direction
	TrackDegreeTrue	0:1	*IBIS-IP.double*	Value of the direction based on the real north pole
	TrackDegreeMagnetic	0:1	*IBIS-IP.double*	Value of the direction based on the magnetic north pole
	GNSSType	**1:1**	*GNSSTypeEnumeration*	Information about the used GNSS system (cf. chapter 12.13)
	GNSSCoordinateSystem	0:1	*GNSSCoordinateSystemEnumeration*	Information about the coordinate system (cf. chapter 12.11)

Table 48: Description of GNSSLocationService.Data

9.6 JourneyInformationService

The `JourneyInformationService` in IBIS-IP is the service providing functionalities of the journey information determination functional component. For this it reads the relevant information from the schedule data that has been transferred from the background systems to the vehicle and provides this information to requesting services. Because the reliable transfer of information is important, this service is provided as an HTTP service. To change for example the current block number or the current trip reference the service provides the **`SetData`** operation.

9.6.1 Operations of JourneyInformationService

Operation	Request/ Response	Used Data type, Data structure
GetAllData	Req.	-
	Resp.	JourneyInformationService. GetAllDataResponseStructure
SubscribeAllData	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeAllData	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetCurrentBlockRef	Req.	-
	Resp.	JourneyInformationService. GetCurrentBlockRefResponseStructure
SubscribeCurrentBlockRef	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeCurrentBlockRef	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
RetrievePartialTripSequence	Req.	JourneyInformationService. RetrievePartialTripSequenceRequestStructure
	Resp.	JourneyInformationService. RetrievePartialTripSequenceResponseStructure
RetrieveSpecificBlockInformation	Req.	JourneyInformationService. RetrieveSpecificBlockInformationRequestStructure
	Resp.	JourneyInformationService. RetrieveSpecificBlockInformationResponseStructure
RetrieveSpecificStopInformation	Req.	JourneyInformationService. RetrieveSpecificStopInformationRequestStructure
	Resp.	JourneyInformationService. RetrieveSpecificStopInformationResponseStructure
RetrieveSpecificTSPPointInformation	Req.	JourneyInformationService. RetrieveSpecificTSPPointInformationRequestStruc-ture
	Resp.	JourneyInformationService. RetrieveSpecificTSPPointInformationResponseStruc-ture
RetrieveSpecificTimingPointInformation	Req.	JourneyInformationService. RetrieveSpecificTimingPointInformationRequestStruc-ture
	Resp.	JourneyInformationService. RetrieveSpecificTimingPointInformationRe-

Operation	Request/ Response	Used Data type, Data structure
		sponseStructure
RetrieveSpecificGNSSPointInformation	Req.	JourneyInformationService. RetrieveSpecificGNSSPointInformationRequestStructure
	Resp.	JourneyInformationService. RetrieveSpecificGNSSPointInformationResponseStructure
RetrieveSpecificBeaconPointInformation	Req.	JourneyInformationService. RetrieveSpecificBeaconPointInformationRequestStructure
	Resp.	JourneyInformationService. RetrieveSpecificBeaconPointInformationResponseStructure
ListAllDisplayContents	Req.	-
	Resp.	JourneyInformationService. ListAllDisplayContentsResponseStructure
ListAllLineInformation	Req.	-
	Resp.	JourneyInformationService. ListAllLineInformationResponseStructure
ListAllDestinationInformation	Req.	-
	Resp.	JourneyInformationService. ListAllDestinationInformationResponseStructure
ListAllViaPoint	Req.	-
	Resp.	JourneyInformationService. ListAllViaPointResponseStructure
ListAllAdditionalDisplayInformation	Req.	-
	Resp.	JourneyInformationService. ListAllAdditionalDisplayInformationResponseStructure
ListAllRoutes	Req.	-
	Resp.	JourneyInformationService. ListAllRoutesResponseStructure
RetrieveAllRoutesPerLine	Req.	JourneyInformationService. RetrieveAllRoutesPerLineRequestStructure
	Resp.	JourneyInformationService. ListAllRoutesResponseStructure
SetBlockNumber	Req.	JourneyInformationService. SetBlockNumberRequestStructure
	Resp.	DataAcceptedResponseStructure
SetTripRef	Req.	JourneyInformationService. SetTripRefRequestStructure

Operation	**Request/ Response**	**Used Data type, Data structure**
	Resp.	DataAcceptedResponseStructure
SetDisplayContent	Req.	JourneyInformationService. SetDisplayContentRequestStructure
	Resp.	DataAcceptedResponseStructure
SetCurrentTripIndex	Req.	JourneyInformationService. SetCurrentTripIndexRequestStructure
	Resp.	DataAcceptedResponseStructure
SetCurrentStopIndex	Req.	JourneyInformationService. SetCurrentStopIndexRequestStructure
	Resp.	DataAcceptedResponseStructure
SetAdditionalAnnouncement	Req.	JourneyInformationService. SetAdditionalAnnouncementRequestStructure
	Resp.	DataAcceptedResponseStructure
SetAdditionalTextMessage	Req.	JourneyInformationService. SetAdditionalTextMessageRequestStructure
	Resp.	DataAcceptedResponseStructure

Table 49: Description of Operationen des JourneyInformationService

9.6.2 Data Structure of Operation GetAllData

9.6.2.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.6.2.2 Response

JourneyInformation-Service.GetAllDataResponse				*+Structure*	**Response structure after requesting information from the JourneyInformationService**
				Choice	One of the structures below
	a	***AllData***	**–1:1**	*+JourneyInformation-Ser-vice.DataContent*	detailed response structure (cf. below)
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 50: Description of JourneyInformationService.GetAllDataResponse

JourneyInformationService.DataContent			*+Structure*	**Detailed response structure with all information of JourneyInformationService**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	BlockRef	0:1	*IBIS-IP.NMTOKEN*	Indication of the block number
	CurrentTripIndex	0:1	*IBIS-IP.int*	Index of the current trip
	TripSequence	**1:***	*+TripSequence*	Indication of the sequence of trips (cf. chapter 10.55)

Table 51: Description of JourneyInformationService.DataContent

9.6.3 Data Structure of Operation SubscribeAllData

For this subscription the data structures from chapters 10.51 and 10.52 are used.

9.6.4 Data Structure of Operation UnsubscribeAllData

To terminate this subscription the structures of chapters 10.57 and 10.58 are used.

9.6.5 Data Structure of Operation GetCurrentBlockRef

9.6.5.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.6.5.2 Response

JourneyInformation-Service.GetCurrentBlockRefResponse				*+Structure*	**Response structure for the current block reference of the JourneyInformationService**
				choice	One of the structures below
	a	***CurrentBlockRefData***	**–1:1**	*+JourneyInformation-Service.CurrentBlockRefData*	Detailed response structure (cf. table below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 52: Description of JourneyInformationService.GetCurrentBlockRefResponse

JourneyInformation-Service.CurrentBlockRefData			*+Structure*	**Response structure for the request of the current block reference of the JourneyInformation-Service**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	CurrentBlockRef	**1:1**	*IBIS-IP.NMTOKEN*	Indication of the current block number

Table 53: Description of JourneyInformationService.CurrentBlockRefData

9.6.6 Data Structure of Operation SubscribeCurrentBlockRef

For this subscription the data structures from chapters 10.51 and 10.52 are used.

9.6.7 Data structure fo Operation UnsubscribeCurrentBlockRef

To terminate this subscription the structures of chapters 10.57 and 10.58 are used.

9.6.8 Data Structure of Operation RetrievePartialTripSequence

9.6.8.1 Request

JourneyInformation-Service.RetrievePartialTripSequenceRequest			*+Structure*	**Request structure for a certain amount of trips in the current block**
	StartingTripIndex	**1:1**	*xs:int*	Index of the first trip that has to be delivered
	NumberOfTrips	**1:1**	*xs:int*	Number of trips that have to be delivered

Table 54: Description of JourneyInformationService.RetrievePartialTripSequenceRequest

9.6.8.2 Response

JourneyInformation-Ser-vice.RetrievePartialTripSequenceResponse				*+Structure*	**Response structure with the answer of *Retrieve-PartialTripSequenceRequest* with a certain amount of trips for the current block**
				choice	One of the structures below
	a	***Partial-TripSequenceData***	**–1:1**	*+JourneyInfor mation-Ser-vice.Partial-TripSequence Data*	Detailed response structure (cf. table below)
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 55: Description of JourneyInformationService.RetrievePartialTripSequenceResponse

JourneyInformation-Service.PartialTripSequenceData			*+Structure*	Response structure for the request of a certain amount of trips for the current block
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	TripSequence	**1:***	*+TripSequenc e*	Sequence of trips (cf. chapter 10.55)

Table 56: Description of JourneyInformationService.PartialTripSequenceData

9.6.9 Data Structure of Operation RetrieveSpecificBlockInformation

9.6.9.1 Request

JourneyInformationSer-vice.RetrieveSpecificBlockInformationReq uest			*+Structure*	**Request structure for information about a specif-ic block**
	BlockRef	**1:1**	*IBIS-IP.NMTOKEN*	Block number for which the information should be returned

Table 57: Description of JourneyInformationService.RetrieveSpecificBlockInformationRequest

9.6.9.2 Response

JourneyInformationSer-vice.RetrieveSpecificBlockInformationResp onse				*+Structure*	**Response structure for the request for a specific block**
				choice	One of the structures below
	a	***SpecificBlockIn-formationData***	**–1:1**	*+JourneyInfor mation-Service.Data*	Detailed response structure with the block infor-mation (cf. chapter 9.6.2.2)
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 58: Description of JourneyInformationService.RetrieveSpecificBlockInformationResponse

For the response to the request for information about a specific block the same data structure as for the request for information about the current block is used (cf. chapter 9.6.2.2).

9.6.10 Data Structure of Operation RetrieveSpecificStopInformation

9.6.10.1 Request

JourneyInformation-Ser-vice.RetrieveSpecificStopInformationRequest			*+Structure*	**Request structure for information about a specific stop**
	StopRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference to the stop point for which the information should be returned

Table 59: Description of JourneyInformationService.RetrieveSpecificStopInformationRequest

9.6.10.2 Response

JourneyInformationSer-vice.RetrieveSpecificStopInformationResponse				*+Structure*	**Response structure for information about a specific stop**
				choice	One of the structures below
	a	***SpecificStopInfor-mationData***	**–1:1**	*+JourneyInfor mation-Ser-vice.SpecificStopInfor-mationData*	Detailed response structure (cf. table below)
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 60: Description of JourneyInformationService.RetrieveSpecificStopInformationResponse

JourneyInformation-Service.SpecificStopInformationData			*+Structure*	**Detailed response structure for a specific stop information**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	JourneyStopInfor-mation	**1:1**	*+JourneyStopI nformation*	Information concerning one specific stop (cf. chapter 10.30)

Table 61: Description of JourneyInformationService.SpecificStopInformationData

9.6.11 Data Structure of Operation RetrieveSpecificTSPPointInformation

9.6.11.1 Request

JourneyInformation-Ser-vice.RetrieveSpecificTSPPointInformationRequest			*+Structure*	**Request structure for information about a specific TSP point**
	TSPPointRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference at the requested TSP point for which information should be returned

Table 62: Description of JourneyInformationService.RetrieveSpecificTSPPointInformationRequest

9.6.11.2 Response

JourneyInformationSer-vice.RetrieveSpecificTSPPointInformationResponse				*+Structure*	**Response structure for information about a specific TSP point**
				choice	One of the structures below
	a	***SpecificTSPPointInformationData***	**–1:1**	*+JourneyInformation-Ser-vice.SpecificTSPPointInfor-mationData*	Detailed response structure (cf. table below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 63: Description of JourneyInformationService.RetrieveSpecificTSPPointInformationResponse

JourneyInformation-Service.SpecificTSPPointInformationData			*+Structure*	Detailed response structure for information about a specific TSP point
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	TSPPoint	**1:1**	*+TSPPoint*	TSP point information (cf. chapter 10.56)

Table 64: Description of JourneyInformationService.SpecificTSPPointInformationData

9.6.12 Data Structure of Operation RetrieveSpecificTimingPointInformation

9.6.12.1 Request

JourneyInformation-Ser-vice.RetrieveSpecificTimingPointInformationRequest			*+Structure*	**Request for information about a specific timing point**
	TimingPointRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference at a requested timing point in the schedule for which information should be returned

Table 65: Description of JourneyInformationService.RetrieveSpecificTimingPointInformationRequest

9.6.12.2 Response

JourneyInformationSer-vice.RetrieveSpecificTimingPointInformationResponse				*+Structure*	**Response structure for information about a specific timing point**
				choice	One of the structures below
	a	***SpecificTimingPointInformationData***	**–1:1**	*+JourneyInformation-Ser-vice.SpecificTimingPointIn-for-mationData*	Detailed request for information (cf. table below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 66: Description of JourneyInformationService.RetrieveSpecificTimingPointInformationResponse

JourneyInformationSer-vice.SpecificTimingPointInformationData			*+Structure*	Detailed request structure for information about a specific timing point
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	TimingPoint	**1:1**	*+TimingPoint*	Timing point information (cf. chapter 10.53)

Table 67: Description of JourneyInformationService.SpecificTimingPointInformationData

9.6.13 Data Structure of Operation RetrieveSpecificGNSSPointInformation

9.6.13.1 Request

JourneyInformation-Ser-vice.RetrieveSpecificGNSSPointInformationRequest			*+Structure*	**Request structure for specific GNSS point data**
	GNSSPointRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference to a specific GNSS point for which information should be returned

Table 68: Description of JourneyInformationService.RetrieveSpecificGNSSPointInformationRequest

9.6.13.2 Response

JourneyInformation-Ser-vice.RetrieveSpecificGNSSPointInformationResponse				*+Structure*	**Response structure for specific GNSS point data**
				choice	One of the structures below
	a	***SpecificGNSSPointInformationData***	**–1:1**	*+JourneyInformation-Ser-vice.SpecificGNSSPointIn-for-mationData*	Detailed response structure for specific GNSS point data (cf. table below)
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 69: Description of JourneyInformationService.RetrieveSpecificGNSSPointInformationResponse

JourneyInformation-Service.SpecificGNSSPointInformationData			*+Structure*	**Detailed response structure for specific GNSS point data**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	GNSSPoint	**1:1**	*+GNSSPoint*	GNSS point information (cf. chapter 10.28)

Table 70: Description of JourneyInformationService.SpecificGNSSPointInformationData

9.6.14 Data Structure of Operation RetrieveSpecificBeaconPointInformation

9.6.14.1 Request

JourneyInformationSer-vice.RetrieveSpecificBeaconPointInformationRequest			*+Structure*	**Request structure for specific beacon point data**
	BeaconRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference to the requested beacon point for which information should be returned

Table 71: Description of JourneyInformationService.RetrieveSpecificBeaconPointInformationRequest

9.6.14.2 Response

JourneyInformationSer-vice.RetrieveSpecificBeaconPointInformationResponse				*+Structure*	**Response structure for information about a specific beacon point**
				choice	One of the structures below
	a	***SpecificBeacon-PointInfor-mationData***	**–1:1**	*+JourneyInfor mation-Ser-vice.SpecificB eaconPointIn-for-mationData*	Detailed response structure (cf. table below)
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 72: Description of JourneyInformationService.RetrieveSpecificBeaconPointInformationResponse

JourneyInformationSer-vice.SpecificBeaconPointInformationData			*+Structure*	**Detailed response structure for specific beacon point data**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	BeaconPoint	**1:1**	*+BeaconPoint*	Information for the beacon point (cf. chapter 10.4)

Table 73: Description of JourneyInformationService.SpecificBeaconPointInformationData

9.6.15 Data Structure of Operation ListAllDisplayContents

9.6.15.1 Request

Because of being a **`List`** operation, there is no request structure for this operation.

9.6.15.2 Response

JourneyInformation-Service.ListAllDisplayContentsResponse				*+Structure*	**Response structure for all current display content**
				choice	One of the structures below
	a	***AllDisplayCon-tentsData***	**–1:1**	*+JourneyInfor mation-Ser-vice.AllDisplay ContentsData*	Detailed response structure (cf. table below)
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 74: Description of JourneyInformationService.ListAllDisplayContentsResponse

JourneyInformation-Service.AllDisplayContentsData			*+Structure*	**Response detailed structure for all current display content**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	DisplayContent	**1:***	*+DisplayCont ent*	Display content (cf. chapter 10.19)

Table 75: Description of JourneyInformationService.AllDisplayContentsData

9.6.16 Data Structure of Operation ListAllLineInformation

9.6.16.1 Request

Because of being a **`List`** operation, there is no request structure for this operation.

9.6.16.2 Response

JourneyInformation-Service.ListAllLineInformationResponse				*+Structure*	**Response structure for all line information**
				choice	One of the structures below
	a	***AllLineInfor-mationData***	**–1:1**	*+JourneyInfor mation-Ser-vice.AllLineInf ormationData*	Detailed response structure (cf. table below)
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 76: Description of JourneyInformationService.ListAllLineInformationResponse

JourneyInformation-Service.AllLineInformationData			*+Structure*	**Detail information structure for all line information**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	LineInformation	**1:***	*+LineInformation*	Line information (cf. chapter 10.31)

Table 77: Description of JourneyInformationService.AllLineInformationData

9.6.17 Data Structure of Operation ListAllDestinationInformation

9.6.17.1 Request

Because of being a **`List`** operation, there is no request structure for this operation.

9.6.17.2 Response

JourneyInformation-Ser-vice.ListAllDestinationInformationRespons e				*+Structure*	**Response structure for all destination texts**
				choice	One of the structures below
	a	***AllDestinationIn-formationData***	**–1:1**	*+JourneyInfor mation-Ser-vice.AllDestin ationInfor-mationData*	Detailed response structure (cf. table below)
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 78: Description of JourneyInformationService.ListAllDestinationInformationResponse

JourneyInformation-Service.AllDestinationInformationData			*+Structure*	**Structure with all destination texts**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	Destination	**1:***	*+Destination*	Information about destination texts (cf. chapter 10.12)

Table 79: Description of JourneyInformationService.AllDestinationInformationData

9.6.18 Data Structure of Operation ListAllViaPoint

9.6.18.1 Request

Because of being a **`List`** operation, there is no request structure for this operation.

9.6.18.2 Response

JourneyInformation-Service.ListAllViaPointResponse				*+Structure*	**Response structure for all via intermediate stops**
				choice	One of the structures below
	a	***AllViapointInfor-mationData***	**–1:1**	*+JourneyInfor mation-Ser-vice.AllViaPoi nData*	Detailed response structure (cf. table below)
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 80: Description of JourneyInformationService.ListAllViaPointResponse

JourneyInformationService.AllViaPointData			*+Structure*	**Detailed information structure about all intermediate stops**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	ViaPoint	**1:***	*+ViaPoint*	Information about via points (cf. chapter 10.60)

Table 81: Description of JourneyInformationService.AllViaPointData

9.6.19 Data Structure of Operation ListAllAdditionalDisplayInformation

9.6.19.1 Request

Because of being a **`List`** operation, there is no request structure for this operation.

9.6.19.2 Response

JourneyInformationSer-vice.ListAllAdditionalDisplayInformationResponse				*+Structure*	**Response structure with all additional display information texts**
				choice	One of the structures below
	a	***AllAdditionalDis-playInformationDa-ta***	**–1:1**	*+JourneyInfor mation-Ser-vice.AllAdditio nalDisplayIn-formationData*	Detailed response structure with display information content (cf. table below)
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 82: Description of JourneyInformationService.ListAllAdditionalDisplayInformationResponse

JourneyInformationSer-vice.AllAdditionalDisplayInformationData			*+Structure*	**Detailed structure with all additional display information texts**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	AdditionalDisplayIn-formation	**1:***	*+International TextType*	Information for additional display texts

Table 83: Description of JourneyInformationService.AllAdditionalDisplayInformationData

9.6.20 Data Structure of Operation ListAllRoutes

9.6.20.1 Request

Because of being a **`List`** operation, there is no request structure for this operation.

9.6.20.2 Response

JourneyInformation-Service.ListAllRoutesResponse				*+Structure*	**Response structure with a listing of all routes**
				choice	One of the structures below
	a	***AllRouteData***	**–1:1**	*+JourneyInformation-Service.AllRouteData*	Detailed response structure (cf. table below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 84: Description of JourneyInformationService.ListAllRoutesResponse

JourneyInformationService.AllRoutesData			*+Structure*	**Detailed response structure with a listing of all routes**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	Route	**1:***	*+TripInformation*	Information for all routes (cf. chapter 10.54)

Table 85: Description of JourneyInformationService.AllRoutesData

9.6.21 Data Structure of Operation RetrieveAllRoutesPerLine

9.6.21.1 Request

JourneyInformation-Service.SetBlockNumberRequest			*+Structure*	**Request structure for all routes for a given line**
	LineRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference to the line for which all routes should be returned

Table 86: Description of JourneyInformationService.SetBlockNumberRequest

9.6.21.2 Response

For the response to the request for all routes of a specific line the same data structure as for the request for all routes is used (cf. chapter 9.6.20.2).

9.6.22 Data Structure of Operation SetBlockNumber

9.6.22.1 Request

JourneyInformation-Service.SetBlockNumberRequest			*+Structure*	**Request structure to set a block number in the JourneyInformationService**
	BlockRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference to the block number to be set

Table 87: Description of JourneyInformationService.SetBlockNumberRequest

9.6.22.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. 10.9) is used.

9.6.23 Data Structure of Operation SetTripRef

9.6.23.1 Request

JourneyInformation-Service.SetTripRefRequest			*+Structure*	**Request structure to set a trip reference in the JourneyInformationService**
	TripRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference at the trip number to be set

Table 88: Description of JourneyInformationService.SetTripRefRequest

9.6.23.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. 10.9) is used.

9.6.24 Data Structure of Operation SetDisplayContent

9.6.24.1 Request

JourneyInformation-Service.SetDisplayContentRequest			*+Structure*	**Request structure to set a display content in the JourneyInformationService**
	DisplayContent	**1:1**	*+DisplayContent*	The display content to be set

Table 89: Description of JourneyInformationService.SetDisplayContentRequest

9.6.24.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. 10.9) is used.

9.6.25 Data Structure of Operation SetCurrentTripIndex

9.6.25.1 Request

JourneyInformation-Service.SetCurrentTripIndexRequest			*+Structure*	**Request structure to set an index for the current trip in the JourneyInformationService**
	CurrentTripIndex	**1:1**	*IBIS-IP.int*	Index of the current trip

Table 90: Description of JourneyInformationService.SetCurrentTripIndexRequest

9.6.25.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. 10.9) is used.

9.6.26 Data Structure of Operation SetCurrentStopIndex

9.6.26.1 Request

JourneyInformation-Service.SetCurrentStopIndexRequest			*+Structure*	**Request structure to set a current stop index in the JourneyInformationService**
	CurrentStopIndex	**1:1**	*IBIS-IP.int*	Index of the current stop

Table 91: Description of JourneyInformationService.SetCurrentStopIndexRequest

9.6.26.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. 10.9) is used.

9.6.27 Data Structure of Operation SetAdditionalAnnouncement

9.6.27.1 Request

JourneyInformation-Service.SetAdditionalAnnouncementRequest			*+Structure*	**Request structure to set an additional announcement in the JourneyInformationService**
	AdditionalAnnouncement	**1:1**	*+AdditionalAnnounce-ment*	Transmission of an additional announcement

Table 92: Description of JourneyInformationService.SetAdditionalAnnouncementRequest

9.6.27.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. 10.9) is used.

9.6.28 Data Structure of Operation SetAdditionalTextMessage

9.6.28.1 Request

JourneyInformation-Service.SetAdditionalTextMessageRequest			*+Structure*	**Request structure for the transmission of an additional text message to the JourneyInformationService**
	AdditionalTextMessage	**1:1**	*IBIS-IP.string*	Additional text message

Table 93: Description of JourneyInformationService.SetAdditionalTextMessageRequest

9.6.28.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. 10.9) is used.

9.7 NetworkLocationService

The `NetworkLocationService` provides information about the current location on the route in the network of public transports. Hence it is the implementation of the functional component network locating. For this, information from the `JourneyInformationService` (the current route) and the services of the functional component physical locating are required.

Because the service provides distances to the next points on the route (and since those are changing frequently) the `NetworkLocationService` is provided as a UDP service.

NetworkLocationService.Data			*+Structure*	**Description structure of NetworkLocationService**
	CurrentTripRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference to the currently valid route
	NextPointRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference to the next point being approached on the route
	DistanceToNextPoint	**1:1**	*IBIS-IP.double*	Distance to the next point being approached on the route
	NextStopPointRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference to the next stop point being approached on the route
	DistanceToNextStopPoint	**1:1**	*IBIS-IP.double*	Distance to the next stop point being approached on the route
	RouteDeviation	0:1	*RouteDeviationEnumeration*	Information if there is a deviation from the planned route (cf. chapter 12.18)
	LocationState	0:1	*LocationStateEnumeration*	Simple classification about the location state on the current route (cf. chapter 12.16)

Table 94: Description of NetworkLocationService.Data

9.8 PassengerCountingService

This service counts the entrancing and escaping passengers at every single door and provides this data to the system. To allow different suppliers at every door this service is a very special actor in the system, because every door provides its own `PassengerCountingService`. The data just change when the vehicle is leaving a stop point, so it is required to use the http for realisation.

9.8.1 Operations of the PassengerCountingService

The *PassengerCountingService* support the following operations (cf. Table 95). The description of the structures is placed afterword in chapters 9.8.2ff.

Operation	**Request/ Response**	**Used data type, Data structure**
GetAllData	Req.	-
	Resp.	PassengerCountingService. GetAllDataResponseStructure
SubscribeAllData	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeAllData	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
RetrieveSpecificDoorData	Req.	PassengerCountingService. RetrieveSpecificDoorDataRequestStructure
	Resp.	PassengerCountingService. RetrieveSpecificDoorDataResponseStructure
SetCounterData	Req.	PassengerCountingService. SetCounterDataRequestStructure
	Resp.	DataAcceptedResponseStructure

Table 95: Description of Operations of PassengerCountingService

9.8.2 Data Structure of Operation GetAllData

9.8.2.1 Request

Because of being a **Get** operation, there is no request structure for this operation.

9.8.2.2 Response

PassengerCountingS-ervice.GetAllDataResponse				*+Structure*	**Response structure of the PassengerCoun-tingService**
				choice	One of the structures below
	a	***Data***	**–1:1**	*+PassengerC ountingS-ervice.AllData*	Detailed response structure (cf. table below)
	b	***ErrorMessage***		*IBIS-IP.string*	Error message

Table 96: Description of PassengerCountingService.GetAllDataResponse

PassengerCountingService.AllData			*+Structure*	**Description structure for data content of the PssengerCountingService**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	CountingData	0:*	*+DoorInformat ion*	Structure of the counting data and further generic information for a specific door (cf. chapter 10.22)

Table 97: Description of PassengerCountingService.AllData

9.8.3 Data Structure of Operation SubscribeAllData

For this subscription the data structures from chapters 10.51 and 10.52 are used.

9.8.4 Data Structure of Operation UnsubscribeAllData

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.8.5 Data Structure of Operation RetrieveSpecificDoorData

To retrieve counting information from a specific door this operation makes is possible.

9.8.5.1 Request

PassengerCountingService.RetrieveSpecificDoorDataRequest			*+Structure*	**Request structure for the specific door information counting data**
	DoorID	**1:1**	*IBIS-IP.NMTOKEN*	ID for door identification

Table 98: Description of PassengerCountingService.RetrieveSpecificDoorDataRequest

9.8.5.2 Response

PassengerCountingService.RetrieveSpecificDoorDataResponse				*+Structure*	**Response structure with counting data of a specific door**
				Choice	One of the structures below
	a	***Data***	**–1:1**	*+PassengerCountingService.AllData*	Detailed structure with counting data of a specific door (cf. table below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 99: Description of PassengerCountingService.RetrieveSpecificDoorDataResponse

PassengerCountingService. SpecificDoorData			*+Structure*	**Detailed structure with counting data of a specific door**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Time stamp of the response
	CountingData	**1:1**	*+DoorInformation*	Detailed structure with counting data and additional information of a specific door (cf. chapter 10.22)

Table 100: Description of PassengerCountingService. SpecificDoorData

9.8.6 Data Structure of Operation SetCounterData

To Set/Reset the counting data of a specific door this operation is needed.

9.8.6.1 Request

PassengerCountingService.SetCounterDataRequest			*+Structure*	**Request structure for setting data at the PassengerCountingService**
	DoorSetList	**1:***	*+DoorCountingList*	List of doors the counter data get set (cf. chapter 10.21)

Table 101: Description of PassengerCountingService.SetCounterDataRequest

9.8.6.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. 10.9) is used.

9.9 SystemDocumentationService

This service documents the system and device status and provides the initial system configuration. For the start-up it is defined that the device starts this service automatically which it is running on. It is required to implement this service as an http-service.

9.9.1 Operations of the SystemDocumentationService

Operation	Request/ Response	Used Data type, Data structure
GetSystemConfiguration	Req.	-
	Resp.	SystemDocumentationService. GetSystemConfigurationResponseStructure
SubscribeSystemConfiguration	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeSystemConfiguration	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
StoreSystemConfiguration	Req.	SystemDocumentationService. StoreSystemConfigurationRequestStructure
	Resp.	DataAcceptedResponseStructure
StoreLogMessages	Req.	SystemDocumentationService. StoreLogMessagesRequestStructure
	Resp.	DataAcceptedResponseStructure
RetrieveLogMessages	Req.	SystemDocumentationService. RetrieveLogMessagesRequestStructure
	Resp.	SystemDocumentationService. RetrieveLogMessagesResponseStructure

Table 102: Description of Operations at the SystemDocumentationService

9.9.2 Data Structure of Operation GetSystemConfiguration

9.9.2.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.9.2.2 Response

SystemDocumentation-Service.GetSystemConfigurationResponse				*+Structure*	**Response structure with system configuration data**
				Choice	One of the structures below
	a	***SystemConfigura-tion***	**–1:1**	*+SystemDocumentation-Ser-vice.SystemConfigura-tionData*	detailed response structure (cf. below)
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 103: Description of SystemDocumentationService.GetSystemConfigurationResponse

SystemDocumentation-Service.SystemConfigurationData			*+Structure*	**Response structure with the system configuration**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	ServiceStartList	**1:1**	*+ServiceStartList*	List with information for all needed services (cf. chapter 10.45)
	DeviceSpecification-List	**1:1**	*+DeviceSpecificationList*	List with information about all available devices (cf. chapter 10.16)
	HertbeatIntervall	0:1	*xs:duration*	Heartbeat value for the cyclic request of the SystemManagementService

Table 104: Description of SystemDocumentationService.SystemConfigurationData

9.9.3 Data Structure of Operation SubscribeSystemConfiguration

For this subscription the data structures from chapters 10.51 and 10.52 are used.

9.9.4 Data Structure of Operation UnsubscribeSystemConfiguration

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.9.5 Data Structure of Operation StoreSystemConfiguration

9.9.5.1 Request

SystemDocumentation-Service.StoreSystemConfigurationRequest			*+Structure*	**Request structure to store a new system configuration**
	ServiceStartList	**1:1**	*+ServiceStartList*	List of services to be started (cf. chapter 10.45)
	DeviceSpecification-List	**1:1**	*+DeviceSpecificationList*	List of available devices (cf. chapter 10.16)
	HertbeatIntervall	0:1	*xs:duration*	Heartbeat value for the cyclic request of the SystemManagementService

Table 105: Description of SystemDocumentationService.StoreSystemConfigurationRequest

9.9.5.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. 10.9) is used.

9.9.6 Data Structure of Operation StoreLogMessages

9.9.6.1 Request

SystemDocumentation-Service.StoreLogMessagesRequest			*+Structure*	**Request structure for transmission of logging messages**
	LogMessage	0:*	*+LogMessage*	Logging message to be stored (cf. chapter 10.32)
	DataVersion	0:*	*+DataVersion*	Information about the data version to be stored (cf. chapter 10.11)

Table 106: Description of SystemDocumentationService.StoreLogMessagesRequest

9.9.6.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. 10.9) is used.

9.9.7 Data Structure of Operation RetrieveLogMessages

9.9.7.1 Request

SystemDocumentation-Service.RetrieveLogMessagesRequest			*+Structure*	**Request structure for logging messages**
	InformationType	**1:1**	*SystemDocu-mentationIn-for-mationEnu-meration*	Information about the logging message type (cf. table below)
	NumberOfEntries	**1:1**	*xs:int*	Number of entities

Table 107: Description of SystemDocumentationService.RetrieveLogMessagesRequest

9.9.7.2 Response

SystemDocumentation-Service.RetrieveLogMessagesResponse				*+Structure*	**Response structure with logging messages data**
				choice	One of the structures below
	a	***LogMessageData***	**–1:1**	*+SystemDocu mentation-Ser-vice.LogMess ageData*	detailed response structure (cf. below)
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 108: Description of SystemDocumentationService.RetrieveLogMessagesResponse

SystemDocumentation-Service.LogMessageData			*+Structure*	**Detailed response structure for logging messag-es**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	Message	**1:***	*+Message*	Message content (cf. chapter10.33)

Table 109: Description of SystemDocumentationService.LogMessageData

9.10 SystemManagementService

The *SystemManagementService* is the representation of the functional component *SystemManagement*. Its major tasks are the following:

- start all in the system configuration listed services
- monitor all devices and service status and
- provide the device and service status to the system.

The *SystemManagementService* is unique in an operating IBIS-IP system. It is automatically starting on exactly one of the devices participating in the IBIS-IP-network. It connects with exactly one *SystemDocumentationServices*.

The operation of two *SystemManagementServices* in parallel is not allowed.

The service is designed as an http-service.

9.10.1 Operations of the SystemManagementService

Operation	**Request/ Response**	**Used data types, Data structures**
GetDeviceStatus	Req.	-

	Resp.	SystemManagementService. GetDeviceStatusResponseStructure
SubscribeDeviceStatus	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeDeviceStatus	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
GetServiceStatus	Req.	-
	Resp.	SystemManagementService. GetServiceStatusResponseStructure
SubscribeServiceStatus	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeServiceStatus		UnsubscribeRequestStructure
		UnsubscribeResponseStructure

Table 110: Description of Operations at the SystemManagementService

9.10.2 Data Structure of Operation GetDeviceStatus

9.10.2.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.10.2.2 Response

SystemManagementService.GetDeviceStatusResponse				*+Structure*	**Response structure for the device state of all devices**
				choice	One of the choices below
	a	***SystemManagementService.GetDeviceStatusResponseData***	**–1:1**	*+ SystemManagementService.GetDeviceStatusResponseData*	Detailed response structure with information about the status of all devices (cf. table below)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 111: Description of SystemManagementService.GetDeviceStatusResponse

SystemManagementService.GetDeviceStatusResponseData			*+Structure*	**Detailed response structure with information about the status of all devices**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	DeviceSpecificationWithStateList	**1:1**	*+DeviceSpecificationWithStateList*	Device list with all device states (vgl. 10.18)

Table 112: Description of SystemManagementService.GetDeviceStatusResponseData

9.10.3 Data Structure of Operation SubscribeDeviceStatus

For this subscription the data structures from chapters 10.51 and 10.52 are used.

9.10.4 Data Structure of Operation UnsubscribeDeviceStatus

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.10.5 Data Structure of Operation GetSystemStatus

9.10.5.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.10.5.2 Response

SystemManagementSer-vice.GetServiceStatusResponse				*+Structure*	**Response structure with the service state of all devices in the system**
				choice	One of the choices below
	a	***SystemManage-mentSer-vice.GetServiceStatusResponseData***	**–1:1**	*+SystemManagementSer-vice.GetServiceSta-tusResponse Data*	Detailed response structure with the device state of all services in the system (cf. table below)
	b	***OperationEr-rorMessage***		*IBIS-IP.string*	Error message

Table 113: Description of SystemManagementService.GetServiceStatusResponse

SystemManagementSer-vice.GetServiceStatusResponseData			*+Structure*	**Detailed response structure with the device state of all services in the system**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	ServiceIdentification-WithStateList	**1:1**	*+ServiceIdentificationWith-StateList*	Service specification and state list (cf. chapter . 10.39)

Table 114: Description of SystemManagementService.GetServiceStatusResponseData

9.10.6 Data Structure of Operation SubscribeSystemStatus

For this subscription the data structures from chapters 10.51 and 10.52 are used.

9.10.7 Data Structure of Operation UnsubscribeSystemStatus

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.11 TicketingService

The TicketingService is a representation of the functional component Ticketing Data Determination.

The service provides all information which are mandatory for selling and validating tickets on board. Since the reliable transfer of data is of importance this service is designed as an http service.

9.11.1 Operations of the TicketingService

Operation	**Request/ Response**	**Used Data types, Data structure**
SetRazzia	Req.	TicketingService. SetRazziaRequestStructure
	Resp.	DataAcceptedResponseStructure
GetCurrentTariffInformation	Req.	-

	Resp.	TicketingService. GetTariffInformationResponseStructure
SubscribeCurrentTariffInformation	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeCurrentTariffInformation	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure
RetrieveTariffInformation	Req.	TicketingService. RetrieveTariffInformationRequestStructure
	Resp.	TicketingService. RetrieveTariffInformationResponseStructure
ValidateTicket	Req.	TicketingService. ValidateTicketRequestStructure
	Resp.	TicketingService. ValidateTicketResponseStructure
GetValidationResult	Req.	-
	Resp.	TicketingService. ValidationResultStructure
SubscribeValidationResult	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeValidationResult	Req.	UnsubscribeRequestStructure
	Resp.	SubscribeResponseStructure

Table 115: Operation of the TicketingService

9.11.2 Data Structure of Operation SetRazzia

This operation allows setting the Razzia status at the ticketing devices to avoid further selling and validation of tickets while a ticket control takes place.

9.11.2.1 Request

TicketingService.SetRazzia			*+Structure*	**Request structure for setting the razzia status at the TicketService**
	TicketInformationRazziaStatus	**1:1**	*TicketRazziaInformationEnumeration*	Information about the razzia status to be set (cf. chapter 12.7)

Table 116: Description of TicketingService.SetRazzia

9.11.2.2 Response

For the acknowledge of the request the DataAcceptedResponseStructure (cf. 10.9) is used.

9.11.3 Data Structure of Operation GetCurrentTariffInformation

This operation provides current tariff information corresponding to the current location inside the public transport network.

9.11.3.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.11.3.2 Response

The data content of the response is referred to the current location.

TicketingService.GetTariffInformationResponse				*+Structure*	**Response structure with current tariff information**
				choice	One of the structures below
	a	***TicketingSevice.GetTariffInformationResponseData***	**–1:1**	*+ TicketingSevice.GetTariffInformationResponseDataStructure*	Detailed response structure (cf. Table 118)
	b	***ErrorMessage***		*IBIS-IP.string*	Error message

Table 117: Description of TicketingService.GetTariffInformationResponse

TicketingSevice.GetTariffInformationResponseDataStructure			*+Structure*	**Detailed structure with the current tariff information of the TicketingService**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	DefaultLanguage	**1:1**	*IBIS-IP.language*	Default language
TarifInformationgroup	***TripRef***	**1:1**	*+IBIS-IP.NMTOKEN*	Trip reference
	Line	**1:1**	*+ LineInfoRmationStructure*	Information about the line (cf. chapter 10.31)
	StopPointTariffInformation	**1:1**	*+ StopPointTariffInformationStructure*	Tariff Information for the specific stop point (cf. chapter 10.49)
	ShortTripStopList	**0:***	*+ShortTripStopListStructure*	Tariff information to the short trip tariff at the specific stop point (cf. chapter 10.46)

Table 118: Description of the TicketingSevice.GetTariffInformationResponseDataStructure

9.11.4 Data Structure of Operation SubscribeCurrentTariffInformation

For this subscription the data structures from chapters 10.51 and 10.52 are used.

9.11.5 Data Structure of Operation UnsubscribeCurrentTariffInformation

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.11.6 Data Structure of Operation RetrieveTariffInformation

The operation **`RetrieveTariffInformation`** allows retrieving the tariff information for a specific location (not the current).

9.11.6.1 Request

TicketingService.RetrieveTariffInformation			*+Structure*	**Request structure for a specific location tariff information**
	TripRef	1:1	*IBIS-IP.NMTOKEN*	Trip reference
	LineRef	1:1	*IBIS-IP.NMTOKEN*	Line reference
	StopRef	1:1	*IBIS-IP.NMTOKEN*	Stop reference

Table 119: Description of TicketingService.RetrieveTariffInformation

9.11.6.2 Response

The response data content of this operation is delivered in the structure from chapter 9.11.3.2.

9.11.7 Data Structure of Operation ValidateTicket

The operation ValidateTicket serves the validation and cancelling of an e-ticket.

9.11.7.1 Request

TicketInformation-Service.Validation.GetDataRequest			*+Structure*	**Request structure with a validation or cancelling request for a ticket**
	CardType	**1:1**	*+CardType*	Information about the card type (cf. chapter 10.7)
	CardApplInformation	0:1	*+CardAppl-Informations*	Information about the number and type of applications running on the card (cf. chapter 10.5)
Card-Ticket-Data-Blocks	***NumberOfCardTicket-DataBlocks***	**1:1**	*IBIS-IP.unsignedInt*	Number of data blocks read
	CardTicketDataBlock	**1:***	*+CardTicket-DataBlocks*	Data blocks read (cf. chapter 10.6)

Table 120: Description of the TicketInformationService.Validation.GetDataRequest

9.11.7.2 Response

TicketingService.ValidateTicketResponse				*+Structure*	**Response structure with tariff information for the ticket validation**
				choice	One of the structures below
	a	***TicketingS-ervice.ValidationResponseData***	**–1:1**	*+ Ticketing-Sevice.Validate-Ticket-ResponseData-Structure*	Detailed validation data (cf. Table 122)
	b	***ErrorMessage***		*IBIS-IP.string*	Error message

Table 121: Description of the TicketingService.ValidateTicketResponse

TicketingService.ValidateTicketResponse-DataStructure			*+Structure*	**Response structure with detailed validation data of the TicketingService**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	GlobalCardStatus	**1:1**	*+GlobalCard-Status*	Global card status (cf. chapter 10.27)
	CardType	**1:1**	*+CardType*	Information about the card type (cf. chapter 10.7)
Card-Application-Validation	***CardApplStatusCode***	**1:1**	*IBIS-IP.unsignedInt*	Status code of the validated application based on the EN 1545
	CardApplicationIn-formation		*+ CardAppl-Informations*	Information about the number and type of applications running on the card (cf. chapter 10.5)
Card-Data-Validation	***CardValidationCode***	**1:1**	*IBIS-IP.unsignedInt*	Result code of the validation based at the VDV-KA
	CardTicketData	**1:1**	*+CardTicket-Data*	Validation ticket data (cf. chapter 10.6)

Table 122: Description of the TicketingService.ValidateTicketResponseDataStructure

9.11.8 Data Structure of Operation GetValidationResult

This operation serves the validation result and provides this to the system.

9.11.8.1 Request

Because of being a **`Get`** operation, there is no request structure for this operation.

9.11.8.2 Response

The data content of the **`GetValidationResult`** is just concerning one validated ticket.

TicketingService.GetValidationResultResponse				*+Structure*	**Response structure for ticket validation**
				choice	One of the structures below
	a	***ValidationResult-Data***	**–1:1**	*+ Ticketing-Sevice.ValidationResult-DataStructure*	Validation result information (cf. Table 124)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 123: Description of the TicketingService.GetValidationResultResponse

TicketingService.GetValidationResultResponseStructure			*+Structure*	**Detailed validation result information**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Response time stamp
	ValidationResult	**1:1**	*TicketValidationEnumeration*	Validation result values (cf. chapter 12.22)

Table 124: Description of the TicketingService.GetValidationResultResponseStructure

9.11.9 Data Structure of Operation SubscribeValidationResult

For this subscription the data structures from chapters 10.51 and 10.52 are used.

9.11.10 Data Structure of Operation UnsubscribeValidationResult

To terminate this subscription the structures of chapters 10.57 and 10.58 are required.

9.12 TimeService

This specific service has no own implementation based on XML as the services described before. For the time synchronisation the Simple Network Time Protocol has to be used. It is based on the RFC 4330 and enhances mechanisms for the synchronisation of server clients based on a server time. The publishing of the necessary multicast address of the SNTP server has to happen via DNS-SD inside the TXT record (`<sntp-server=$IP-Adresse>`). The time zone is also published in the TXT record (`<time-zone=UTC+1>`). Of Course, a further periodic publishing is not allowed.

10 Common data structures

10.1 AdditionalAnnouncement

AdditionalAnnouncement				*+Structure*	**Structure which describes the additional information for an announcement**
		AnnouncementRef	**1:1**	*IBIS-IP.NMTOKEN*	announcement reference
		AnnouncementText	0:*	*+International TextType*	Announcement text
		AnnouncementTTSText	0:*	*+International TextType*	Announcement text for text to speech engines
				choice	One of the choices below
	a	***ImmediateInformation***	**–1:1**	*IBIS-IP.boolean*	Immediate sending of the additional announcement
	b	***PeriodicalInformation***		*IBIS-IP.duration*	Periodical sending of the additional announcement
	c	***InformationAtSpecificPoint***		*+SpecificPoint*	Sending of an announcement at a specific (trip) point (point information cf. chapter 10.37)

Table 125: Description of AdditionalAnnouncement

10.2 Announcement

Announcement			*+Structure*	**Structure with information which is needed for an announcement**
	AnnouncementRef	**1:1**	*IBIS-IP.NMTOKEN*	announcement reference
	AnnouncementText	0:*	*+International TextType*	Announcement text
	AnnouncementTTSText	0:*	*+International TextType*	Announcement text for text to speech engines

Table 126: Description of Announcement

10.3 BayArea

BayArea			*+Structure*	**Structure which describes the bay area (in relation to the stop sign)**
	BeforeBay	0:1	*IBIS-IP.double*	Bay begin, distance to the stop sign in meters in moving dirction
	BehindBay	0:1	*IBIS-IP.double*	Bay ending, distance after the stop sign in metersin moving direction

Table 127: Description of BayArea

10.4 BeaconPoint

BeaconPoint			*+Structure*	**Structure which describes a beacon point**
	PointRef	0:1	*IBIS-IP.NMTOKEN*	Reference at a point
	BeaconCode	**1:1**	*IBIS-IP.NMTOKEN*	Beacon code
	ShortName	0:*	*+International TextType*	Beacon short name
	Description	0:*	*+International TextType*	Description of the beacon

Table 128: Description of BeaconPoint

10.5 CardApplInformation

CardApplInformations			*+Structure*	**Structure for information of applications of a read card**
	CardApplInformation-Length	**1:1**	*IBIS-IP.unsignedInt*	Length of the byte array from *CardApplInformationData*
	CardApplInformation-Data	**1:***	*IBIS-IP.byte*	Data array for application data

Table 129: Description of CardApplInformation

10.6 CardTicketData

CardTicketData			*+Structure*	**information of tariff data on card**
	CardTicketDataID	**1:1**	*IBIS-IP.unsigned-Long*	Card ID
	CardTicketDataLength	**1:1**	*IBIS-IP.unsignedInt*	Length of ticket data
	CardTicketData	**1:***	*IBIS-IP.byte*	Data array for ticket information

Table 130: Description of CardTicketData

10.7 CardType

CardType			*+Structure*	**Structure to describe a card type (ticket)**
	CardSerialNumber	**1:1**	*IBIS-IP.NMTOKEN*	Serial number of the card
	CardTypeID	**1:1**	*IBIS-IP.NMTOKEN*	Type ID of the card
	CardTypeText	0:*	*+International TextType*	Type ID of the card as string/text

Table 131: Description of CardType

10.8 Connection

Connection			*+Structure*	**Structure which describes a connection**
	StopRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference at a stop point which the connection is concerning on
	ConnectionRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference at the connection
	ConnectionType	**1:1**	*ConnectionTypeEnumeration*	Type of connection (cf. chapter 12.2)
	DisplayContent	**1:1**	*+DisplayContent*	Display content of the distributor cf. chapter 10.19)
	Platform	0:1	*IBIS-IP.string*	Information about the platform for the interchange
	ConnectionState	0:1	*ConnectionStateEnumeration*	Description of the connection state in case of a ordered connection (cf. chapter 12.1)
	TransportMode	0:1	*+Vehicle*	Information about the transport mode for the connection (cf. chapter 10.59)
	ExpectedDepartureTime	0:1*	*IBIS-IP.dateTime*	Information about the expected departure

Table 132: Description of Connection

10.9 DataAcceptedResponse

DataAcceptedResponse			*+Structure*	**General response structure for an operation which expects data**
	DataAcceptedResponseData	**1:1**	*+DataAcceptedResponseDataStructure*	Detailed response structure including data (cf. chapter 10.10)
	OperationErrorMessage	**1:1**	*IBIS-IP.string*	Error message

Table 133: Description of DataAcceptedResponse

10.10 DataAcceptedResponseData

DataAcceptedResponseData			*+Structure*	**Detailed response structure including data**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Time stamp of the response
	DataAccepted	**1:1**	*IBIS-IP.boolean*	Data acknowledge
	ErrorCode	0:1	*ErrorCodeEnumeration*	Descriptive value for an error (cf. chapter 12.9)
	ErrorInformation	0:1	*IBIS-IP.string*	Error code information

Table 134: Description of DataAcceptedResponseData

10.11 DataVersion

With the data version different versions of this XML-Scheme are possible in one system

DataVersion			*+Structure*	**Structure with information of the data version**
	DataType	**1:1**	*IBIS-IP.string*	Free text description of the data type
	VersionRef	**1:1**	*IBIS-IP.NMTOKEN*	Version information

Table 135: Description of DataVersion

10.12 DataVersionList

DataVersionList			*+Structure*	**Structure with which several data versions can be listed**
	DataVersion	**1:***	***+DataVersion***	Data Structure for the description of data types (cf. chapter 10.11)

Table 136: Description of DataVersionList

10.13 Destination

Destination			*+Structure*	**Structure with information about the destination**
	DestinationRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference at the display destination text
	DestinationName	0:*	*+International TextType*	Text which is published at the display
	DestinationShortName	0:*	*+International TextType*	Short text which is published at the display

Table 137: Description of Destination

10.14 DeviceInformation

DeviceInformation			*+Structure*	**Structure with non changeable device configuration data**
DeviceInformationGroup	***DeviceName***	**1:1**	*IBIS-IP.string*	Device name
	Manufacturer	**1:1**	*IBIS-IP.string*	Manufacturer of the device
	SerialNumber	**1:1**	*IBIS-IP.NMTOKEN*	Serial number of the device
	DeviceClass	**1:1**	*+DeviceClass Enumeration*	One of the possible device class (cf. 12.3)
	DataVersionList	0:1	*+DataVersion List*	List with the data versions (cf. 10.1210.11)
	WebInterfaceAddress	0:1	*IBIS-IP.anyURI*	URI for a optional web interface for maintenance

Table 138: Description of DeviceInformation

10.15 DeviceSpecification

DeviceSpecification			*+Structure*	**Structure which describes a device**
	DeviceClass	**1:1**	*DeviceClassEnumeration*	One of the available device class (cf. 12.3)
	DeviceID	**1:1**	*IBIS-IP.NMTOKEN*	Device-ID

Table 139: Description of DeviceSpecification

10.16 DeviceSpecificationList

DeviceSpecificationList			*+Structure*	**Structure with the device specification list**
	DeviceSpecification	**1:***	*+DeviceSpecification*	Device information (cf. chapter 10.15)

Table 140: Description of DeviceSpecificationList

10.17 DeviceSpecificationWithState

DeviceSpecificationWithState			*+Structure*	**Structure with the device specification including the current working states**
	DeviceSpecification	**1:1**	*+DeviceSpecification*	Structure which describes a device (10.15)
	DeviceState	**1:1**	*DeviceStateEnumeration*	possible states of the device (cf. 12.4)

Table 141: Description of DeviceSpecificationWithState

10.18 DeviceSpecificationWithStateList

DeviceSpecificationWithStateList			*+Structure*	**List of objects of with device specifications and their states**
	DeviceSpecification-WithState	**1:***	*+DeviceSpecificationWith-State*	Structure with the device specification including the current working states

Table 142: Description of DeviceSpecificationWithStateList

10.19 DisplayContent

DisplayContent			*+Structure*	**Structure with the complete display content**
	DisplayContentRef	0:1	*IBIS-IP.NMTOKEN*	Reference at the display content
	LineInformation	**1:1**	*+LineInformation*	information about the line, which has to be displayed (cf. chapter 10.31)
	Destination	**1:1**	*+Destination*	Information about the destination, which has to be displayed (cf. chapter 10.12)
	ViaPoint	0:*	*+ViaPoint*	Information about the via points, which has to be displayed (cf. chapter10.60)
	AdditionalInformation	0:*	*+InternationalTextType*	Information about the additional information like express bus, additional bus etc. , which has to be displayed
DisplayPolicy	*Priority*	0:1	*IBIS-IP.nonNegativeInteger*	Information about the display priority
	PeriodDuration	0:1	*IBIS-IP.duration*	Information about the period duration
	Duration	0:1	*IBIS-IP.duration*	Duration of a display turn

Table 143: Description of DisplayContent

10.20 DoorCounting

DoorCounting			*+Structure*	**Counting data of a door**
	ObjectClass	**1:1**	*DoorCountingObject-ClassEnumeration*	Value with the detailed description of the counted object (cf. chapter 12.5)
	In	**1:1**	*+IBIS-IP.int*	Number of boarded passengers
	Out	**1:1**	*+IBIS-IP.int*	Number of escaped passengers
	CountQuality	0:1	*DoorCountingQualityEnumeration*	Textstring with information on the quality of counting (cf. 12.6)

Table 144: Description of DoorCounting

10.21 DoorCountingList

DoorCountingList			*+Structure*	**Structure for a list of door with for which values are set**
	DoorID	**1:1**	*IBIS-IP.NMTOKEN*	ID for identification of the door
	CountSet	**1:***	*+DoorCounting*	Structure with counting values (cf. chapter 10.15)

Table 145: Description of DoorCountingList

10.22 DoorInformation

DoorInformation			*+Structure*	**Structure with information about the counting at a specific door**
	DoorID	**1:1**	*IBIS-IP.NMTOKEN*	ID for identification of the door
	Count	**1:***	*+DoorCounting*	structure for the counting data (cf. chapter 10.15)
	State	0:1	*+DoorState*	Structure with door states (cf. chapter 10.25)

Table 146: Description of DoorInformation

10.23 DoorOpenState

DoorOpenState			*+Structure*	**Door state**
	Value	**1:1**	*DoorOpenStateEnumeration*	Description value of the opening state of a door (cf. chapter 12.7)
	ErrorCode	0:1	*ErrorCodeEnumeration*	Descriptive value for an error (cf. chapter 12.9)

Table 147: Description of DoorOpenState

10.24 DoorOperationState

DoorOperationState			*+Structure*	**Door operation state**
	Value	**1:1**	*DoorOperationStateEnumeration*	Description value of the operation state (cf. chapter 12.8)
	ErrorCode	0:1	*ErrorCodeEnumeration*	Descriptive value for an error (cf. chapter 12.9)

Table 148: Description of DoorOperationState

10.25 DoorState

DoorState			*+Structure*	**Structure for description of the door state**
	OpenState	**1:1**	*+DoorOpenState*	**Structure for description of** door opening state (cf. chapter 10.23)
	OperationState	0:1	*+DoorOperationState*	Structure for description of the operation state (cf. chapter 10.24)

Table 149: Description of DoorState

10.26 FareZoneInformation

FareZoneInformation			*+Structure*	**Structure for the description of information for tariffs and fare zones**
Fare-Zone-Infor-mation	***FarezoneID***	**1:1**	*IBIS-IP.NMTOKEN*	Index of afare zone
	FarezoneType	0:1	*+ZoneType*	Information about the fare zone type (cf. chapter 10.61)
	FarezoneLongName	0:*	*+International TextType*	Fare zone long name
	FarezoneShortName	0:*	*+International TextType*	Fare zone short name

Table 150: Description of DoorState

10.27 GlobalCardStatus

GlobalCardStatus			*+Structure*	**Global card status**
	GlobalCardStatusID	**1:1**	*IBIS-IP.unsignedInt*	ID of Card status based on the EN 1545
	GlobalCardStatusText	**0:***	*IBIS-IP.string*	Text of global card status based on the EN 1545

Table 151: Description of GlobalCardStatus

10.28 GNSSPoint

GNSSPoint			*+Structure*	**Structure which describes a point where coordinates are used for locating the point**
	PointRef	0:1	*IBIS-IP.NMTOKEN*	Reference at a GNSS point
	Longitude	**1:1**	*+GNSSCoordinate*	Structure for geographical longitude (cf. chapter 10.29)
	Latitude	**1:1**	*+GNSSCoordinate*	Structure for geographical latitude (cf. chapter 10.29)
	Altitude	0:1	*IBIS-IP.double*	Geographical Altitude

Table 152: Description of GNSSPoint

10.29 GNSSCoordinate

GNSSCoordinate			*+Structure*	**Structure for describing coordinates on the surface**
	Degree	**1:1**	*IBIS-IP.double*	Coordinate in degree
	Direction	**1:1**	*IBIS-IP.string*	geographical direction

Table 153: Description of GNSSCoordinate

10.30 JourneyStopInformation

JourneyStopInformation			*+Structure*	**Structure to describe a stop point by the Journey Information Determination**
	StopRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference at the stop point
	StopName	**1:***	*+InternationalTextType*	Name of stop point
	StopAlternativeName	0:*	*+InternationalTextType*	Alternative name of stop point
	Platform	0:1	*IBIS-IP.string*	Name of the plattform
	DisplayContent	1:*	*+DisplayContent*	Information about display content (cf. chapter 10.19)
	Announcement	0:*	*+Announcement*	Information for announcement (cf. chapter 10.2)
	ArrivalScheduled	0:1	*IBIS-IP.dateTime*	Scheduled arrival
	DepartureScheduled	0:1	*IBIS-IP.dateTime*	Scheduled departure
	Connection	0:*	*+Connection*	Information about the connections (cf. chapter 10.8)
	BayArea	0:1	*+BayArea*	Information about the size of the Bay Area (cf. chapter 10.3)
	GNSSPoint	0:1	*+GNSSPoint*	Information for the Geo-Coordinates of the stop point (cf. chapter 10.28)
	FareZone	0:*	*IBIS-IP.NMTOKEN*	Valid fare zone at the current stop point

Table 154: Description of JourneyStopInformation

10.31 LineInformation

LineInformation			*+Structure*	**Structure for description of the line information**
	LineRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference at the line
	LineName	0:*	*+InternationalTextType*	name of line
	LineShortName	0:*	*+InternationalTextType*	short name of line
	LineNumber	0:1	*IBIS-IP.int*	number of line

Table 155: Description of LineInformation

10.32 LogMessage

LogMessage			*+Structure*	**Structure for logging message**
	MessageProvider	**1:1**	*+DeviceSpecification*	Massage provider (cf. chapter 10.15)
	MessageBody	**1:1**	*+Message*	Message content (cf. chapter 10.33)

Table 156: Description of LogMessage

10.33 Message

Message			*+Structure*	**Structure for describing a message**
	Message-ID	**1:1**	*IBIS-IP.int*	index of message
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	time stamp, when the message was created
	MessageType	**1:1**	*MessageTypeEnumeration*	kind of message (cf. chapter 12.17)
	MessageText	**1:1**	*IBIS-IP.string*	Message text

Table 157: Description of Message

10.34 Point

Point			*+Structure*	**Structure with (logical) point description**
	PointIndex	**1:1**	*IBIS-IP.int*	Point index
	PointType	**1:1**	*+PointType*	Type of the point (cf. chapter 10.36)
	DistanceToPreviousPoint	**1:1**	*IBIS-IP.int*	Distance to the previous point in [m]

Table 158: Description of Point

10.35 PointSequence

PointSequence			*+Structure*	**Structure for describing a sequence of points**
	Point	**2:***	*+Point*	Description of points (cf. chapter 10.34)

Table 159: Description of PointSequence

10.36 PointType

PointType				*+Structure*	**Structure for choosing a specific point type**
				choice	One of the structures below
	a	***StopPoint***	**–1:1**	*+JourneyStopInformation*	Stop point (cf. chapter 10.30)
	b	***BeaconPoint***		*+BeaconPoint*	Beacon point (cf. chapter 10.4)
	c	***GNSSLocationPoint***		*+GNSSPoint*	Point, location described in coordinates (cf. chapter 10.28)
	d	***TimingPoint***		*+TimingPoint*	Point where a schedule comparison should take place (cf. chapter 10.53)
	e	***TSPPoint***		*+TSPPoint*	Point for traffic light priorisation (cf. chapter 10.56)

Table 160: Description of PointType

10.37 ServiceIdentification

The ServiceIdentification structure allows identifying a service in the system. Contrary to the ServiceSpecification (cf. 10.37) this structure is including the information about the device where the service is running.

ServiceIdentification			*+Structure*	**Structure for the unique identification of a service in the system**
	ServiceName	**1:1**	*+ServiceSpecification*	Structure for the service description
	Device	**1:1**	*+DeviceSpecification*	Structure for device description

Table 161: Description of ServiceIdentification

10.38 ServiceIdentificationWithState

ServiceIdentificationWithState			*+Structure*	**Structure for unique identification of a service in the whole system including its state**
	ServiceIdentification	**1:1**	*+ServiceIdentification*	Structure for unique identification of a service in the whole system (cf. 10.37)
	ServiceState	**1:1**	*ServiceStateEnumeration*	Information about the state of the service

Table 162: Description of ServiceIdentificationWithState

10.39 ServiceIdentificationWithStateList

ServiceIdentificationWithStateList			*+Structure*	**Structure with a list of all unique services and their state in the system**
	ServiceSpecification-WithState	**1:***	*+ServiceSpecificationWith-State*	Structure for the unique identification of a service including its state(cf. 10.43)

Table 163: Description of ServiceIdentificationWithStateList

10.40 ServiceInformation

ServiceInformation			*+Structure*	**Structure for description of the services which are available on a device**
	Service	**1:1**	*+ServiceSpecification*	Structure for description of a service (cf. chapter 10.42)
	Autostart	**1:1**	*IBIS-IP.boolean*	Information whether a service has to be started automatically by the DeviceManagement (and not by the SystemManagementService), especially relevant for the vehicle operation functionalities

Table 164: Description of ServiceInformation

10.41 ServiceInformationList

ServiceInformationList			*+Structure*	**Structure for describing a list of services which are available on a device**
	ServiceInformation	**1:***	*+ServiceInformation*	Structure for describing available services (cf. chapter 10.40)

Table 165: Description of ServiceIdentificationWithStateList

10.42 ServiceSpecification

The ServiceSpecification describes a service at a device based on its ServiceName and the protocol version. For a system wide identification it is necessary to know the device where the service is running (ServiceIdentification cf. 10.37).

ServiceSpecification			*+Structure*	**Structure for the unique service identification on a device**
	ServiceName	**1:1**	*ServiceNameEnumeration*	A possible service (cf. 12.19)
	IBIS-IP-Version	**1:1**	*IBIS-IP.NMTOKEN*	Version information of the used protocol (this is especially necessary for the SystemManagementService to know which service (version) has to be started)

Table 166: Description of ServiceSpecification

10.43 ServiceSpecificationWithState

ServiceSpecificationWithState			*+Structure*	**Structure for the unique identification of a service at the device including its status**
	ServiceSpecification	**1:1**	*+ServiceSpecification*	Structure which describes a service (cf. 10.40)
	ServiceState	**1:1**	*ServiceStateEnumeration*	Information about the operation sate of the service

Table 167: Description of ServiceSpecificationWithState

10.44 ServiceSpecificationWithStateList

ServiceSpecificationWithStateList			*+Structure*	**Structure with a list of the service specifications including the operation states**
	ServiceSpecificationWithState	**1:***	*+ServiceSpecificationWithState*	Description structure of a service including the operation state (cf. 10.43)

Table 168: Description of ServiceSpecificationWithStateList

10.45 ServiceStartList

ServiceStartList			*+Structure*	**Structure with services that are available at one device**
	ServiceIdentification	**1:***	*+ServiceIdentification*	Structure with all available services (cf. 10.37)

Table 169: Description of ServiceStartList

10.46 ShortTripStopList

ShortTripStopList			*+Structure*	**Structure with a list of all possible short trips**
	ShortTripStopList	**1:***	*+StopPointTariffInformation*	Structure which describes one short trip (cf. chapter 10.49)

Table 170: Description of ShortTripStopList

10.47 SpecificPoint

SpecificPoint			*+Structure*	**Structure with a specific point**
	PointRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference at a point
	DistanceToPreviousPoint	**1:1**	*IBIS-IP.double*	Distance to the previous point in [m]

Table 171: Description of SpecificPoint

10.48 StopInformation

StopInformation			*+Structure*	**Structure for description of a stop point**
	StopIndex	**1:1**	*IBIS-IP.int*	Index of this stop point in a list of stop point
	StopRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference at a stop point
	StopName	**1:***	*+InternationalTextType*	name of stop point
	StopAlternativeName	0:*	*+InternationalTextType*	alternative name of stop point
	Platform	0:1	*IBIS-IP.string*	Name of the platform
	DisplayContent	**1:***	*+DisplayContent*	Information about the display content (cf. chapter 10.19)
	StopAnnouncement	0:*	*+Announcement*	Information about the announcement (cf. chapter 10.2)
	ArrivalScheduled	0:1	*IBIS-IP.dateTime*	Scheduled arrival
	DepartureScheduled	0:1	*IBIS-IP.dateTime*	Scheduled departure
	RecordedArrivalTime	0:1	*IBIS-IP.dateTime*	Recorded arrival time
	DistanceToNextStop	0:1	*IBIS-IP.int*	Distance to the next stop point
	Connection	0:*	*+Connection*	Information about the connections (cf. chapter 10.8)
	FareZone	0:*	*IBIS-IP.NMTOKEN*	Valid fare zone at this stop point

Table 172: Description of StopInformation

10.49 StopPointTariffInformation

StopPointTariffInformationStructure			*+Structure*	**Structure with tariff information for astop point**
Stop-Point-Tariff-Information	***JourneyStopInformation***	**1:1**	*+Journey-StopInformationStructure*	Information about the requested stop point (cf. chapter 10.30)
	FareZoneInformation	**1:1**	*+FareZone-Information-Structure*	Information about the fare zone for this stop point (cf. chapter 10.26)

Table 173: Description of StopInformation

10.50 StopSequence

StopSequence			*+Structure*	**Structure for describing a sequence of stop points**
	StopPoint	**2:***	*+StopInformation*	Stop point information (cf. chapter 10.48)

Table 174: Description of StopSequence

10.51 SubscribeRequest

SubscribeRequest			*+Structure*	**Structure with a subscription request**
	Client-IP-Address	**1:1**	*IBIS-IP.string*	IP address of the client for which subscription
	ReplyPort	0:1	*IBIS-IP.int*	Reply port for the subscription
	Reply-Path	0:1	*IBIS-IP.string*	Reply path for the subscriptions

Table 175: Description of SubscribeRequest

10.52 SubscribeResponse

SubscribeResponse				*+Structure*	**Structure for the subscription response**
				Choice	One of the structures below
	a	***Active***	**–1:1**	*IBIS-IP.boolean*	Information about the subscription acknowledgement
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 176: Description of SubscribeResponse

10.53 TimingPoint

TimingPoint			*+Structure*	**Structure for describing a point, where a schedule comparison should take place**
	TimingPointRef	0:1	*IBIS-IP.NMTOKEN*	Reference at a point
	ScheduleTime	**1:1**	*IBIS-IP.dateTime*	Scheduled departure time
	GNSSPoint	**1:1**	*+GNSSPoint*	GNSS information (cf. chapter 10.28)

Table 177: Description of TimingPoint

10.54 TripInformation

TripInformation			*+Structure*	**Structure with trip information**
	TripRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference at the trip ID
	StopSequence	**1:1**	*+StopSequence*	Description of a stop sequence (cf. chapter 10.50)
	LocationState	0:1	*LocationStateEnumeration*	Roughly information for the current position between two stop point (cf. chapter 12.16)
	TimetableDelay	0:1	*IBIS-IP.int*	Timetable delay in [min]
	AdditionalTextMessage	0:1	*IBIS-IP.string*	Additional text information
	AdditionalAnnouncement	0:*	*+AdditionalAnnouncement*	Additional announcement (cf. chapter 10.1)

Table 178: Description of TripInformation

10.55 TripSequence

TripSequence			*+Structure*	**Structure with a trip sequence**
	TripRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference at the trip ID
	TripIndex	0:1	*IBIS-IP.int*	Index at the current trip
	TripStart	0:1	*IBIS-IP.time*	Scheduled trip start
	CurrentStopIndex	0:1	*IBIS-IP.int*	Information about the index of the current stop point
	JourneyMode	0:1	*JourneyModeEnumeration*	Information about the mode of the journey (cf. chapter 12.15)
	PointSequence	**1:1**	*+PointSequence*	Description of a sequence of points (cf. chapter 10.35)

Table 179: Description of TripSequence

10.56 TSPPoint

TSPPoint			*+Structure*	**Structure with description of a point for traffic light priorisation**
	TSPPointRef	0:1	*IBIS-IP.NMTOKEN*	Reference at a TSP point
	TSPCode	**1:1**	*IBIS-IP.NMTOKEN*	TSP content
	ShortName	0:*	*+International TextType*	TSP short name
	Description	0:*	*+International TextType*	TSP description

Table 180: Description of TSPPoint

10.57 UnsubscribeRequest

UnsubscribeRequest			*+Structure*	**Structure for the request of termination of a subscription**
	Client-IP-Address	**1:1**	*IBIS-IP.string*	Information about the IP address where the subscription has to be terminated
	ReplyPort	0:1	*IBIS-IP.int*	Information about the reply port where the subscription has to be terminated
	Reply-Path	0:1	*IBIS-IP.string*	Information about the reply path where the subscription has to be terminated

Table 181: Description of UnsubscribeRequest

10.58 UnsubscribeResponse

UnsubscribeResponse				*+Structure*	**Structure for the response to a request of termination of a subscription**
				Choice	One of the structures below
	a	***Active***	**–1:1**	*IBIS-IP.boolean*	Information about the termination
	b	***OperationErrorMessage***		*IBIS-IP.string*	Error message

Table 182: Description of UnsubscribeResponse

10.59 Vehicle

Vehicle			*+Structure*	**Structure with information about the vehicle**
	VehicleTypeRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference at a vehicle type
	Name	0:*	*+International TextType*	Vehicle name

Table 183: Description of Vehicle

10.60 ViaPoint

ViaPoint			*+Structure*	**Structure which describes a via point**
	ViaPointRef	**1:1**	*IBIS-IP.NMTOKEN*	Reference at a via stop point
	PlaceRef	0:1	*IBIS-IP.NMTOKEN*	Reference at the associated stop place
	PlaceName	0:*	*+International TextType*	name of the via point
	PlaceShortName	0:*	*+International TextType*	short name of the via point
	ViaPointDisplayPriority	0:1	*IBIS-IP.int*	Information about the display priority of the via point

Table 184: Description of ViaPoint

10.61 ZoneType

ZoneType			*+Structure*	**Structure for description of a zone type**
	FarezoneTypeID	**1:1**	*IBIS-IP.NMTOKEN*	Index at the fare zone type
	FarezoneTypeName	0:*	*+International TextType*	Fare zone type name

Table 185: Description of ZoneType

11 Special IBIS-IP-Data types

11.1 IBIS-IP.anyURI

IBIS-IP.anyURI			*+Structure*	**IBIS-IP-Structure for an address**
	Value	**1:1**	*xs:anyURI*	value
	ErrorCode	0:1	*Error-CodeEnumer-ation*	Descriptive value for an error (cf. chapter 12.9)

Table 186: Description of IBIS-IP.anyURI

11.2 IBIS-IP.boolean

IBIS-IP.boolean			*+Structure*	**IBIS-IP-Structure for description of an Boolean value**
	Value	**1:1**	*xs:boolean*	value
	ErrorCode	0:1	*Error-CodeEnumer-ation*	Descriptive value for an error (cf. chapter 12.9)

Table 187: Description of IBIS-IP.boolean

11.3 IBIS-IP.byte

IBIS-IP.byte			*+Structure*	**IBIS-IP-Structure for description of an byte**
	Value	**1:1**	*xs:byte*	value
	ErrorCode	0:1	*Error-CodeEnumer-ation*	Descriptive value for an error (cf. chapter 12.9)

Table 188: Description of IBIS-IP.byte

11.4 IBIS-IP.date

IBIS-IP.date			*+Structure*	**IBIS-IP-Structure for description of a date**
	Value	**1:1**	*xs:date*	value
	ErrorCode	0:1	*Error-CodeEnumer-ation*	Descriptive value for an error (cf. chapter 12.9)

Table 189: Description of IBIS-IP.date

11.5 IBIS-IP.dateTime

IBIS-IP.dateTime			*+Structure*	**IBIS-IP-Structure with date and time values**
	Value	**1:1**	*xs:dateTime*	value
	ErrorCode	0:1	*Error-CodeEnumer-ation*	Descriptive value for an error (cf. chapter 12.9)

Table 190: Description of IBIS-IP.dateTime

11.6 IBIS-IP.double

IBIS-IP.double			*+Structure*	**IBIS-IP-Structure with a “double” value**
	Value	**1:1**	*xs:double*	Value
	ErrorCode	0:1	*Error-CodeEnumer-ation*	Descriptive value for an error (cf. chapter 12.9)

Table 191: Description of IBIS-IP.double

11.7 IBIS-IP.duration

IBIS-IP.duration			*+Structure*	**IBIS-IP-Structure for duration values**
	Value	**1:1**	*xs:duration*	Value
	ErrorCode	0:1	*Error-CodeEnumer-ation*	Descriptive value for an error (cf. chapter 12.9)

Table 192: Description of IBIS-IP.duration

11.8 IBIS-IP.int

IBIS-IP.int			*+Structure*	**IBIS-IP-Structure for description of an integer**
	Value	**1:1**	*xs:int*	Value
	ErrorCode	0:1	*Error-CodeEnumer-ation*	Descriptive value for an error (cf. chapter 12.9)

Table 193: Description of IBIS-IP.int

11.9 IBIS-IP.language

IBIS-IP.language			*+Structure*	**IBIS-IP-Structure for a language**
	Value	**1:1**	*xs:language*	Language description
	ErrorCode	0:1	*Error-CodeEnumer-ation*	Descriptive value for an error (cf. chapter 12.9)

Table 194: Description of IBIS-IP.language

11.10 IBIS-IP.NMTOKEN

IBIS-IP.NMTOKEN			*+Structure*	**IBIS-IP-Structure for an index value**
	Value	**1:1**	*xs:NMTOKEN*	Value
	ErrorCode	0:1	*Error-CodeEnumer-ation*	Descriptive value for an error (cf. chapter 12.9)

Table 195: Description of IBIS-IP.NMTOKEN

11.11 IBIS-IP.nonNegativeInteger

IBIS-IP.nonNegativeInteger			*+Structure*	**IBIS-IP-Structure for a non negative integer**
	Value	**1:1**	*xs:nonNegativ eInteger*	Value
	ErrorCode	0:1	*Error-CodeEnumer-ation*	Descriptive value for an error (cf. chapter 12.9)

Table 196: Description of IBIS-IP.nonNegativeInteger

11.12 IBIS-IP.normalizedString

IBIS-IP.normalizedString			*+Structure*	**IBIS-IP-Structure for a normalized string**
	Value	**1:1**	*xs:normalized String*	value
	ErrorCode	0:1	*Error-CodeEnumer-ation*	Descriptive value for an error (cf. chapter 12.9)

Table 197: Description of IBIS-IP.normalizedString

11.13 IBIS-IP.string

IBIS-IP.string			*+Structure*	**IBIS-IP-Structure for a string**
	Value	**1:1**	*xs:string*	Value
	ErrorCode	0:1	*Error-CodeEnumer-ation*	Descriptive value for an error (cf. chapter 12.9)

Table 198: Description of IBIS-IP.string

11.14 IBIS-IP.time

IBIS-IP.time			*+Structure*	**IBIS-IP-Structure for the time**
	Value	**1:1**	*xs:time*	Value
	ErrorCode	0:1	*Error-CodeEnumer-ation*	Descriptive value for an error (cf. chapter 12.9)

Table 199: Description of IBIS-IP.time

11.15 IBIS-IP.unsignedInt

IBIS-IP.unsignedInt			*+Structure*	**IBIS-IP-Structure for unsigned integer**
	Value	**1:1**	*xs:unsignedInt*	unsigned integer value
	ErrorCode	0:*	*Error-CodeEnumer-ation*	Descriptive value for an error (cf. chapter 12.9)

Table 200: Description of IBIS-IP.unsignedInt

11.16 IBIS-IP.unsignedLong

IBIS-IP.unsignedLong			*+Structure*	**IBIS-IP-Structure for unsigned long**
	Value	**1:1**	*xs:unsignedLong*	Insigned Long value
	ErrorCode	0:1	*Error-CodeEnumer-ation*	Descriptive value for an error (cf. chapter 12.9)

Table 201: Description of IBIS-IP.unsignedLong

11.17 InternationalTextType

InternationalTextType			*+Structure*	**IBIS-IP-Structure for description of in international text**
	Value	**1:1**	*IBIS-IP.string*	Value
	Language	**1:1**	*IBIS-IP.language*	Language
	ErrorCode	0:1	*Error-CodeEnumer-ation*	Descriptive value for an error (cf. chapter 12.9)

Table 202: Description of InternationalTextType

12 Common used enumerations

This chapter describes the in IBIS-IP defined enumerations that are context based used.

12.1 ConnectionStateEnumeration

Enumeration Name	**Possible Values**	**Description**
ConnectionStateEnumeration	ConnectionBroken ConnectionOK NoInformationAvailable	Information about the status of the connection

Table 203: Description of ConnectionStateEnumeration

12.2 ConnectionTypeEnumeration

Enumeration Name	**Possible Values**	**Description**
ConnectionTypeEnumeration	Interchange ProtectedConnection	Information about the type of the connection

Table 204: Description of ConnectionTypeEnumeration

12.3 DeviceClassEnumeration

Enumeration Name	Possible Values	Description
DeviceClassEnumeration	OnBoardUnit SideDisplay FrontDisplay InteriorDisplay Validator TicketVendingMachine AnnouncementSystem MMI VideoSystem APC MobileInterface Other TestDevice	Information about the device class (cf. chapter 3.4)

Table 205: Description of DeviceClassEnumeration

12.4 DeviceStateEnumeration

Enumeration Name	Possible Values	Description
DeviceStateEnumeration	defective notavailable running	Information about the device state (cf. chapter 6.3.1)

Table 206: Description of DeviceStateEnumeration

12.5 DoorCountingObjectClassEnumeration

Enumeration Name	Possible Values	Description
DoorCountingObjectClassEnumeration	Adult Bike Child Pram Wheelchair Unidentified Others	Information about the counted objects at the counting of passengers

Table 207: Description of DoorCountingObjectClassEnumeration

12.6 DoorCountingQualityEnumeration

Enumeration Name	Possible Values	Description
DoorCountingQualityEnumeration	Defect Other Regular Sabotage	Information about the counting quality

Table 208: Description of DoorCountingQualityEnumeration

12.7 DoorOpenStateEnumeration

Enumeration Name	Possible Values	Description
DoorOpenStateEnumeration	DoorsOpen AllDoorsClosed SingleDoorOpen SingleDoorClosed	Information about the opening state of a door

Table 209: Description of DoorOpenStateEnumeration

12.8 DoorOperationStateEnumeration

Enumeration Name	Possible Values	Description
DoorOperationStateEnumeration	Locked Normal EmergencyRelease	Information about the dorr operation state of a door

Table 210: Description of DoorOperationStateEnumeration

12.9 ErrorCodeEnumeration

Enumeration Name	Possible Values	Description
ErrorCodeEnumeration	DataEstimated FaultData NoScheduleDataAvailable DeviceMissing NoServiceResponse ImportantDataNotAvailable DataNotValid	Descriptive Information about the error reason (cf. chapter 6.3.2)

Table 211: Description of ErrorCodeEnumeration

12.10 ExitSideEnumeration

Enumeration Name	Possible Values	Description
ExitSideEnumeration	both left right unknown	Information about the exit side

Table 212: Description of ExitSideEnumeration

12.11 GNSSCoordinateSystemEnumeration

Enumeration Name	Possible Values	Description
GNSSCoordinateSystemsEnumeration	CH1903 ETSR89 IERS NAD27 NAD83 WGS84 WGS72 SGS85 P90	Information about the coordinate system used by the GNSS system

Table 213: Description of GNSSCoordinateSystemsEnumeration

12.12 GNSSQualityEnumeration

Enumeration Name	Possible Values	Description
GNSSQualityEnumeration	dGPS Estimated GPS NotValid Unknown	Infromation about the GNSS quality

Table 214: Description of GNSSQualityEnumeration

12.13 GNSSTypeEnumeration

Enumeration Name	Possible Values	Description
GNSSTypeEnumeration	GPS Glonass Galileo Beidou IRNSS Other DeadReckoning MixedGNSSTypes	Information about the GNSS type

Table 215: Description of GNSSTypeEnumeration

12.14 IBIS-IP-VersionEnumeration

Enumeration Name	Possible Values	Description
IBIS-IP-VersionEnumeration	1.0	Version of IBIS-IP

Table 216: Description of IBIS-IP-VersionEnumeration

12.15 JourneyModeEnumeration

Enumeration Name	Possible Values	Description of
JourneyModeEnumeration	NoTrip AdditionalTrip ServiceTrip	Information about the journey mode

Table 217: Description of JourneyModeEnumeration

12.16 LocationStateEnumeration

Enumeration Name	Possible Values	Description
LocationStateEnumeration	AfterStop AtStop BetweenStop BeforeStop	Information about the location state relative to the subsequent stop point

Table 218: Description of LocationStateEnumeration

12.17 MessageTypeEnumeration

Enumeration Name	Possible Values	Description
MessageTypeEnumeration	Status Warning Error	Information about a message type

Table 219: Description of MessageTypeEnumeration

12.18 RouteDeviationEnumeration

Enumeration Name	Possible Values	Description
RouteDeviationEnumeration	onroute offroute unknown	Information about the route deviation

Table 220: Description of RouteDeviationEnumeration

12.19 ServiceNameEnumeration

Enumeration Name	Possible Values	Description
ServiceNameEnumeration	BeaconLocationService CustomerInformationService DeviceManagementService DistanceLocationService GNSSLocationService JourneyInformationService NetworkLocationService PassengerCountingService SystemDocumentationService SystemManagementService TicketingService TimeService TestService	Information about the service name (cf. chapter 5.2)

Table 221: Description of ServiceNameEnumeration

12.20 ServiceStateEnumeration

Enumeration Name	Possible Values	Description
ServiceStateEnumeration	defective notrunning running starting standby	Information about the service status

Table 222: Description of ServiceStateEnumeration

12.21 SystemDocumentationInformationEnumeration

Enumeration Name	Possible Values	Description
SystemDocumentationInformationEnumeration	ErrorMessage StatusMessage WarningMessage All	Information about the message type

Table 223: Description of SystemDocumentationInformationEnumeration

12.22 TicketRazziaInformationEnumeration

Enumeration Name	Possible Values	Description
TicketRazziaInformationEnumeration	razzia norazzia	Information whether a razzia takes place

Table 224: Description of TicketRazziaInformationEnumeration

12.23 TicketValidationEnumeration

Enumeration Name	Possible Values	Description
TicketValidationEnumeration	Valid notvalid NoCard	Validation result

Table 225: Description of TicketValidationEnumeration

12.24 VehicleModeEnumeration

Enumeration Name	Possible Values	Description
VehicleModeEnumeration	Air bus coach ferry metro rail tram underground	Vehicle mode information

Table 226: Description of VehicleModeEnumeration

13 Referenzen

[1] VDV-Schrift 300: „Integriertes Bordinformationssystem IBIS“

[2] VDV-Mitteilung 3001: „Kommunikation im ÖV (IP-KOM-ÖV) - Technische Anforderungen für Anwendungen im Integrierten Bordinformationssystem (IBIS)“

[3] VDV-Schrift 453 „VDV-Ist-Datenschnittstellen“

[4] VDV-Schrift 454 „VDV-Ist-Datenschnittstellen“

[5] EN 15531 „Service Interface for Real Time Information„

[6] Für Windows- und MAC-OS-Betriebssysteme gibt es die Bibliotheken von Bonjour, unter Linux die Bibliotheken von Avahi, beide Bibliotheken stehen im Quelltext unter einer Opensource-Lizenz zur Verfügung.

[7] EN 1545 „Identification card systems.“

[8] 2927 „MIME Directory Profile for LDAP Schema“

[9] 6365 „Terminology Used in Internationalization in the IETF“

[10] 2782 „A DNS RR for specifying the location of services (DNS SRV)“

[11] 1035 „Domain Names – Iimplementation an Specification“

[12] 1464 „Using the Domain Name System to Store Arbitrary String Attributes“

[13] 4330 „Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI“

Download of the XSD files under www.vdv.de/ip-kom-oev.aspx

14 Begriffe

Die bereits im Teil 1 definierten Begriffe werden an dieser Stelle nicht wiederholt.

Begriff	**Beschreibung**
AnnouncementSystem	Describes the device class of electro-acoustic systems incl. passenger communication unit
Application	The software on the device is referred to as an application. However, this proprietary software uses specified interfaces..
BeaconLocationService	An IBIS-IP service, which transmits information of the location beacons.
CustomerInformationService	As a central point of information for all aspects of passenger information in IBIS-IP this service is responsible for a consistent delivery of all data.
Device class	For the naming of all connected devices in the network device-classes are specified for all currently known devices
DeviceManagementService	Der auf jedem Gerät in IBIS-IP vorherrschender Dienst stellt Informationen über das Gerät und die laufenden Dienste bereit.
DeviceManagementService	The on every device in IBIS-IP existing service provides information about the device and the running services.
DistanceLocationService	The evaluation of the odometer pulses in IBIS IP is done by this service.
FrontDisplay	Corresponds to the device class FrontDisplay, a display

	on front of a bus indicating its destination.
GNSS coordinates	Due to a satellite-based positioning (e.g. GPS, Galileo) obtained coordinates of a point.
GNSSLocationService	IBIS-IP service which provides the positioning information of a vehicle on base of NMEA-telegrams
HTTP-GET	Request-Function of HTTP, with this function no data is sent
HTTP-POST	Request-Function of HTTP, with this function data could be sent
HTTP-Protocol	Communication protocol in IBIS-IP which is used for event triggered changing data
HTTP-Protocol-Stack	Term for the protocol unit (with TCP and HTTP) for a secure data transmission
InteriorDisplay	Device class for the description of displays inside vehicles independent of the shown contents.
JourneyInformationService	This service provides scheduling data to requesting services/devices.
MobileInterface	Describes the device class of the interface where the passengers with their mobile device can receive information from the vehicle.
Multicast	Communication method for addressing many dedicated receiver.
Multicast group	Describes the dedicated receiver of a multicast mes-

	sage.
NetworkLocationService	This service provides information of the current position on the planned vehicle journey in the network of public transport.
Odometer	A wheel-based rangefinder which produces a certain number of pulses per driven meter.
OnBoardUnit	Device class for the description of the central unit (refers to the master device in IBIS-systems).
Other (Deviceclass)	With this deviceclass devices which are not classified in this standard could be integrated .
PassengerCountigService	This service provides the data of passenger counting on each door in an IBIS-IP-system.
Port	Is an “access” of the protocol, which allows the application a unique source identification
Request/Response	Describes the communication process where a request of communication partner A is answered by a response of communication partner B
SideDisplay	Describes the device class of the lateral outer display on a vehicle.
SRV-Records	Due to RFC 2782 with this record additional information (e.g. device name) to a DNS registration could be provided.
Subscription	With this method, the re-

	questing party is automatically provided with current information.
SystemDocumentationService	This service in IBIS-IP is responsible for documentation of system messages and provision of the system configuration.
SystemManagementService	The main task of the SystemManagementService in IBIS-IP is to inform about the current state of devices and services and to coordinate the system.
TestDevice	Placeholder for the device class of test devices which are connected in case of verification of the system.
TicketingService	This service provides ticketing functions in the IBIS-IP system.
TicketVendingMachine	Deviceclass which describes vending machines for tickets onboard vehicles. They could be managed by the driver as well be autonomous.
Timeserver	Is a key component within IBIS-IP, which provides the system time
TimeService	The TimeService provides via SNTP the current time, date, and time zone.
TransModel	Reference data model based on EN12896.
TXT-Records	According to RFC 1464 this could be used for adding additional information (e.g. devicename) to a DNS-service.

UDP-Protocol	IP-Protocol which in IBIS-IP is used fort he transmission of cyclic changing information.
UML-Convention	Format specifications from UML programming language
USB-Stick	Mobile Data Storage Device
Validator	Deviceclass which is used for stamping validators as well as for the electronic validation of E-tickets.
VideoSystem	Deviceclass which is used for configuration and data transmission of CCTV-systems
Zero Conf	Method and Summary of special technologies for automated identification and configuration of devices in an IP-network

15 Abbreviations

Abbreviations listed in part 1 are not repeated here.

Abbreviation	Description
APC	Automatic Passenger Counting system) Used to describe the device class apc within the vehicle. The technical implementation of the counting is irrelevant.
DNS	Domain Name Server
DNS-SD	Domain Name Server Service Discovery
Eeprom	Electrically Erasable Programmable Read-Only Memory
EKAP	German: Echtzeit Kommunikations- und Auskunftsplattform. Realtime communication and information platform , cf. VDV 431-1.
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IETF	Internet Engineering Task Force
RFC	Request for Comment, technical document of the Internet Society
SNTP	Simple Network Time Protocol
XSD	XML Schema Definition
DELFI	German: Durchgängige Elektronische Fahrplaninformation Fully electronic timetable information system
UML	Unified Modeling Language

16 List of figures

17 List of tables

Verband Deutscher Verkehrsunternehmen e. V. (VDV)
Kamekestraße 37-39 · 50672 Köln
T 0221 57979-0 · F 0221 57979-8000
info@vdv.de · www.vdv.de