VDV Die Verkehrsunternehmen

VDV-Schrift 301-2

07/2016

Internetprotokoll basiertes integriertes Bordinformationssystem IBIS-IP, Teil 2: Schnittstellenspezifikation

Basisdienste V1.0

Gesamtbearbeitung
Ausschuss für Telematik und Informationssysteme (ATI)

Gefördert durch:

Bundesministerium für Wirtschaft und Energie

aufgrund eines Beschlusses des Deutschen Bundestages

Das dieser VDV-Schrift zugrundeliegende Vorhaben IP-KOM-ÖV wurde mit Mitteln des Bundesministeriums für Wirtschaft und Energie unter dem Förderkennzeichen 19P10003 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

VDV-Schrift 301-2 07/2016

Internetprotokoll basiertes integriertes Bordinformationssystem IBIS-IP, Teil 2: Schnittstellenspezifikation

Basisdienste V1.0

Sachbearbeitung
Unterausschuss für Telematik
(UA Telematik)

Autorenverzeichnis
Dipl.-Ing. Dirk Weißer, INIT, Karlsruhe
Dr. Torsten Franke, IVU, Aachen
Dr. Holger Bandelin, Scheidt & Bachmann, Mönchengladbach
Dipl.-Ing. Berthold Radermacher, VDV, Köln
Dipl.-Ing. (FH) Andreas Wehrmann, VDV, Köln
Dipl.-Ing. ETH Walter Meier-Leu, we, Schaffhausen
Dipl.-Ing. René Fischli, Trapeze, Neuhausen

© Verband Deutscher Verkehrsunternehmen e. V. Köln 2015 | Alle Rechte, einschließlich des Nachdrucks von Auszügen, der fotomechanischen oder datenverarbeitungstechnischen Wiedergabe und der Übersetzung, vorbehalten.

Vorwort

Auf Initiative des VDV und gefördert durch das BMWi begann im September 2010 das Forschungs- und Standardisierungsprojekt *Internet Protokoll basierte Kommunikationsdienste im öffentlichen Verkehr (IP-KOM-ÖV)*.

Das Projekt wird von 14 Partnern aus Industrie, Universitäten und Verkehrsunternehmen getragen. Es dient der Erarbeitung moderner Kommunikationskonzepte für die umfassende und kontinuierliche Fahrgastinformation.

Eine umfassende Fahrgastinformation stellt heutzutage ein entscheidendes Wettbewerbsmerkmal im öffentlichen Personenverkehr dar, nicht nur im Vergleich mit anderen Verkehrsunternehmen sondern auch im Vergleich zum Individualverkehr.

Bereits heute ist es üblich, dass Verkehrsunternehmen ihre Fahrgäste nicht nur über die geplanten Fahrten informieren, sondern auch Echtzeitinformationen z. B. zu Verspätungen, Ereignisse oder Fahrtzieländerungen bereitstellen. Diese Informationen werden zum einen über öffentliche Anzeiger bzw. Ansagen in Fahrzeugen oder an Haltestellen allen dort befindlichen Personen zur Verfügung gestellt. Zum anderen lassen sich solche Informationen mit speziellen Applikationen oder über Web-Angebote von Einzelpersonen individuell abfragen.

Bislang ist es aber nicht möglich, Fahrgäste im öffentlichen Verkehr mit Informationen zu ihrer persönlich relevanten Fahrt zu versorgen, den Fahrgast also auch im Störungsfall mit Hilfe des öffentlichen Verkehrs auf dem schnellsten Weg zu seinem Ziel zu führen.

Die weit verbreiteten Smartphones und Tablets bieten hierfür vielfältige Möglichkeiten und ermöglichen eine hohe Akzeptanz der Benutzer. Die Informationsübertragung erfolgt dabei IP-basiert und sollte bevorzugt zwischen einem zentralen Informations-Server und dem Kundenendgerät erfolgen. Für den Fall, dass der zentrale Datenserver nicht erreichbar ist, resp. das Fahrzeug nicht mit einem solchen verbunden ist, sollte auch eine Kommunikation direkt zwischen Kundenendgerät und Fahrzeug möglich sein.

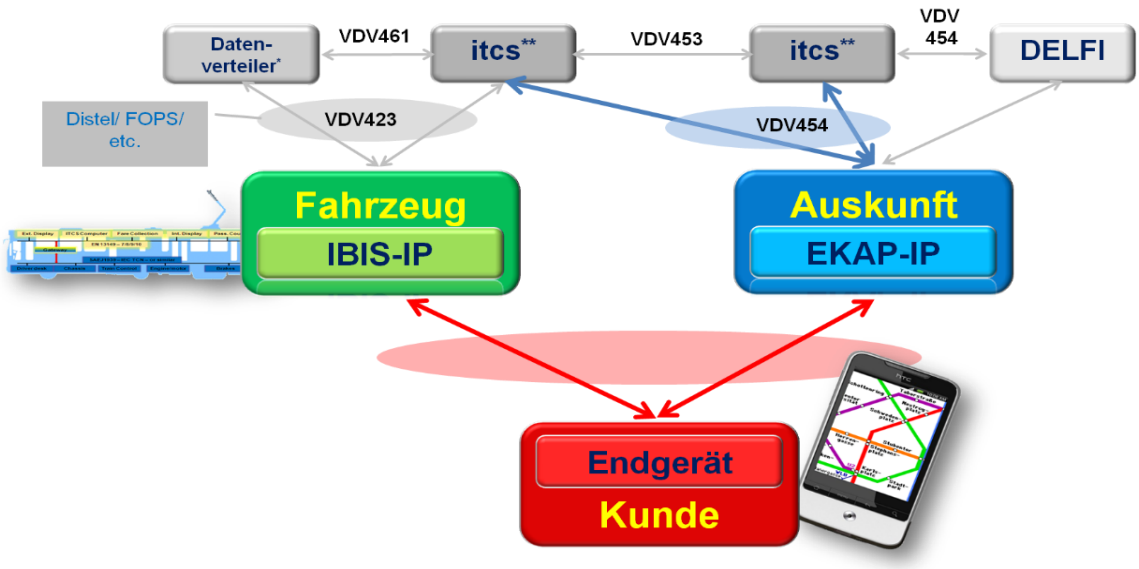

Abbildung 1: Umfeld und Schwerpunkte im Projekt IP-KOM-ÖV

Das Forschungs- und Standardisierungsprojekt IP-KOM-ÖV arbeitet deshalb an drei Schwerpunkten (vgl. Abbildung 1).

Erster Schwerpunkt (grün in Abbildung 1) ist die Spezifikation eines performanten IP-basierten Kommunikationsprotokolls im Fahrzeug (IBIS-IP). Dabei geht es zum einen darum, den gewachsenen Bedürfnissen der fahrzeuginternen Fahrgastinformation und Kommunikation gerecht zu werden. Hierzu wird der in den achtziger Jahren entwickelte IBIS-Wagenbus aus der VDV-Schrift 300 auf eine moderne IP-Informationsarchitektur umgesetzt. Zum anderen geht es um die Definition einer IP-basierten Schnittstelle zur Übertragung der Informationen vom Fahrzeug zum mobilen Kundenendgerät. Die vorliegende Schrift ist das Resultat dieses Teils des Forschungsprojektes.

Zweiter Schwerpunkt (rot in Abbildung 1) ist die Definition der notwendigen Schnittstellen um zukünftigen Applikationen für individuelle Fahrgastinformation unter Verwendung mobiler Geräte des Fahrgasts (Smartphones, Tablet-PC u. ä.) standardisiert zur Verfügung zu stellen. Hierzu wurden im ersten Schritt die Bedürfnisse von Fahrgästen zu individuellen Informationen ermittelt. Im zweiten Schritt wurden einheitliche Schnittstellen zwischen der Echtzeit-Kommunikations- und Auskunftsplattform (EKAP vgl. dritten Schwerpunkt) und den mobilen Kundenendgeräten bzw. zwischen der EKAP und den Hintergrundsystemen entwickelt. Hierbei wurden ausschließlich die Datenmodellierungen und Architekturen erforscht und spezifiziert. Die Entwicklung einer Applikation für mobile Endgeräte war ausdrücklich nicht vorgesehen und wurde lediglich für Tests in einfachster Form realisiert. Die Resultate dieses Teils des Forschungsprojektes sind in der VDV-Schrift 430 dokumentiert.

Dritter Schwerpunkt (blau in Abbildung 1) ist die Definition und Schaffung einer EKAP Die EKAP bündelt Informationen von itcs- und anderen Auskunfts- und Informationssystemen und stellt die Vielzahl an Informationen über geeignete Schnittstellen den Applikationen auf den Kundenendgeräten standardisiert zur Verfügung. Diese Plattform ermöglicht es den Herstellern von Apps, Kunden dynamisch mit individuellen Störungsmeldungen versorgen zu können. Die Resultate dieses Teils des Forschungsprojektes sind ebenfalls in der VDV-Schrift 431 dokumentiert.

Darüber hinaus wird die Praxistauglichkeit dieses neuen Standards in Labor- und Feldtests verifiziert.

Im Zuge der Weiterentwicklung der VDV-301 hat sich gezeigt, dass die Integration aller Dienste in einer einzigen Schrift 301-2 Version 1.0 umständlich in der Handhabung ist. Daher wurden die einzelnen Dienste in einzelne Schriften separiert.

Die Unterteilung der Dienste in einzelne Dokumente ist begründet in der einfacheren Anpassung und Freigabe. Zurzeit sind viele Hersteller daran, die Norm in die Umsetzung zu bringen. Daher ist ein grösserer Anpassungsbedarf vorhanden. Auch können neue Dienste unabhängig von den bestehenden angepasst werden.

Die VDV-301-Schriften bestehen aus grundlegenden Dokumenten wie die VDV-301-1 und VDV-301-2 sowie die einzelnen Beschreibungen der Dienste (VDV-301-2-x).

In der VDV-301-2 werden die technischen Grundlagen wie auch die Basisdienste, welche die Grundlagen eines IBIS-IP-Systems bilden, beschrieben. Unter den Basisdiensten werden die Dienste DeviceManagementService, SystemManagementService und SystemDocumentationService verstanden.

Inhaltsverzeichnis

Vorwort			**6**
Versionshistorie			**14**
Abkürzungen			**15**
1	**Einführung in IBIS-IP**		**16**
1.1	Anforderungen an Geräte in IBIS-IP	16	
1.2	Konfigurationsparameter	16	
1.3	Einbaukennung	17	
1.4	Geräteklassen	17	
1.5	Notationen	18	
1.6	Version des IBIS-IP	19	
2	**Verwendete Kommunikationsprotokolle**		**20**
2.1	Adressierung	20	
2.1.1	IP-Adressen	20	
2.1.2	Subnetzmasken/Gateways	20	
2.2	Konfigurationsparameter für TCP und UDP	20	
2.3	Gültigkeitsdauer von Informationen	20	
2.3.1	Zyklische Informationen	21	
2.3.2	Ereignisgesteuerte Informationen	21	
2.4	Verwendung des UDP- und HTTP-Protokolls	21	
2.5	Weitere Protokolle	22	
3	**Veröffentlichung und Kommunikation der Dienste**		**23**
3.1	Von Fachkomponenten zu Diensten	23	
3.2	Spezifizierte Dienste	25	
3.3	Veröffentlichung via DNS-SD	25	
3.3.1	Nutzung des SRV-Records	25	
3.3.2	Nutzung des TXT-Records	27	
3.4	Veröffentlichung von UDP-Diensten	27	
3.5	Veröffentlichung von HTTP-Diensten	28	
4	**Konventionen für Dienste und Datenstrukturen**		**29**
4.1	Konventionen für HTTP-Dienste	29	
4.1.1	Verwendung von HTTP-POST und HTTP-GET	29	
4.1.1.1	Operationsaufrufe mit HTTP-POST	30	
4.1.1.2	Operationsaufrufe mit HTTP-GET	30	
4.1.2	Namenskonventionen für Operationen	30	

4.1.2.1	Rein datenorientierte Operationen	31
4.1.2.1.1	Einstellen von Werten bei einem Dienst	32
4.1.2.1.2	Einmalige Abfrage von aktuell gültigen Daten bzw. Datenstrukturen	32
4.1.2.1.3	Abonnieren auf Daten bzw. Datenstrukturen	32
4.1.2.1.4	Beenden eines Abonnements auf Daten bzw. Datenstrukturen	33
4.1.2.1.5	Operationen zur Abfrage von spezifischen Informationen	33
4.1.2.1.6	Operationen zur Abfrage von Datenlisten aus der Grunddatenversorgung	34
4.1.2.2	Steuerungsoperationen	34
4.1.2.2.1	Starten	34
4.1.2.2.2	Neustart auslösen	34
4.1.2.2.3	Stoppen	35
4.1.2.2.4	Deaktivieren eines Gerätes	35
4.1.2.2.5	Aktivieren eines Gerätes	35
4.1.2.3	Operationen zur Gültigkeitsprüfung	35
4.1.3	Konvention zu Get/Subscribe/Unsubscribe	35
4.1.4	Unterscheidung zwischen Get und Retrieve	36
4.1.5	Konvention zu Deactivate/Activate	36
4.2	Konventionen für UDP-Dienste	36
4.3	Konventionen für besonderes Dienstverhalten	36
4.3.1	Konventionen für Zustände	36
4.3.2	Konventionen für Fehlermeldungen	37
4.4	Konventionen für die Datenstrukturierung	38
5	**Prinzipien in IBIS-IP**	**39**
5.1	Grundlagen	39
5.1.1	Systemkonfiguration muss vorliegen	39
5.1.2	Nur ein Dienst pro Fachlichkeit	39
5.1.3	Kenntnis der erforderlichen Dienste	40
5.1.4	Identifikation von Diensten und Geräten in IBIS-IP	41
5.1.4.1	Geräte	41
5.1.4.1.1	Fachliche Identifikation	41
5.1.4.1.2	Technische Identifikation	42
5.1.4.2	Dienste	42
5.1.4.2.1	Fachliche Identifikation	42
5.1.4.2.2	Technische Identifikation	42
5.2	Das Konzept des Systemstarts	43
5.2.1	Stufe 1: Start systemrelevanter Fahrzeugbetriebsfunktionen	43
5.2.1.1	Start der Dienste der Fahrzeugbetriebsfunktionen	43
5.2.1.2	Veröffentlichung der Dienste der Fahrzeugbetriebsfunktionen	44
5.2.1.3	Verbindungsaufbau mit den veröffentlichten Diensten der Fahrzeugbetriebsfunktionen und Ausführung fachlicher Aufgaben	45

5.2.1.3.1	Abfrage der Systemkonfiguration	45
5.2.1.3.2	Abfrage der DeviceManagementServices	45
5.2.1.3.3	Zuordnung technischer und fachlicher Verbindungsdaten	45
5.2.1.3.4	Startbefehle an die DeviceManagementServices	45
5.2.2	Stufe 2: Start der fachlichen Dienste	46
5.2.2.1	Start weiterer Dienste auf den Geräten	46
5.2.2.2	Veröffentlichung weiterer Dienste	46
5.2.2.3	Verbindungsaufbau mit den veröffentlichten Diensten und Ausführung fachlicher Aufgaben	46
5.3	Beispiel eines Systemstarts in IBIS-IP	47
5.3.1	Stufe 1: Start systemrelevanter Fahrzeugbetriebsfunktionen	47
5.3.1.1	Start der Dienste der Fahrzeugbetriebsfunktionen	47
5.3.1.2	Veröffentlichung der Dienste der Fahrzeugbetriebsfunktionen	48
5.3.1.3	Verbindungsaufbau mit veröffentlichten Diensten der Fahrzeugbetriebsfunktionen und Ausführung fachlicher Aufgaben	48
5.3.2	Stufe 2: Start der fachlichen Dienste	48
5.3.2.1	Start weiterer Dienste	48
5.3.2.2	Veröffentlichung weiterer Dienste	48
5.3.2.3	Verbindungsaufbau mit den veröffentlichten Diensten und Ausführung fachlicher Aufgaben	48
6	**Strukturierung der Informationsinhalte**	**50**
6.1	Notation der XML-Elemente und -Strukturen	50
6.1.1	Darstellung von XML-Elementen im Text	50
6.1.2	Tabellennotation für Operationen	50
6.1.3	Tabellennotation von XML-Strukturen	51
6.1.3.1	Gruppierung	52
6.1.3.2	Elementname	52
6.1.3.3	Multiplizität & Choice (Min:Max)	52
6.1.3.4	Datentyp	52
6.1.3.5	Erläuterung	53
7	**Beschreibung einzelner Fachdienste**	**54**
7.1	Dienst DeviceManagementService	54
7.1.1	Operationen des DeviceManagementService	54
7.1.2	Datenstrukturen der Operation GetDeviceInformation	57
7.1.2.1	Request	57
7.1.2.2	Response	57
7.1.3	Datenstrukturen der OperationUnsubscribeDeviceInformation	57
7.1.4	Datenstrukturen der Operation GetDeviceConfiguration	57
7.1.4.1	Request	57
7.1.4.2	Response	58

7.1.5	Datenstrukturen der Operation SetDeviceConfiguration	58
7.1.5.1	Request	58
7.1.5.2	Response	58
7.1.6	Datenstrukturen der Operation SubscribeDeviceConfiguration	58
7.1.7	Datenstrukturen der Operation UnsubscribeDeviceConfiguration	58
7.1.8	Datenstrukturen der Operation GetDeviceStatus	59
7.1.8.1	Request	59
7.1.8.2	Response	59
7.1.9	Datenstrukturen der Operation SubscribeDeviceStatus	59
7.1.10	Datenstrukturen der Operation UnsubscribeDeviceStatus	59
7.1.11	Datenstrukturen der Operation GetDeviceErrorMessages	59
7.1.11.1	Request	59
7.1.11.2	Response	60
7.1.12	Datenstrukturen der Operation SubscribeDeviceErrorMessages	60
7.1.13	Datenstrukturen der Operation UnsubscribeDeviceErrorMessages	60
7.1.14	Datenstrukturen der Operation RestartDevice	60
7.1.14.1	Request	60
7.1.14.2	Response	60
7.1.15	Datenstrukturen der Operation DeactivateDevice	60
7.1.15.1	Request	60
7.1.15.2	Response	60
7.1.16	Datenstrukturen der Operation ActivateDevice	61
7.1.16.1	Request	61
7.1.16.2	Response	61
7.1.17	Datenstrukturen der Operation GetServiceInformation	61
7.1.17.1	Request	61
7.1.17.2	Response	61
7.1.18	Datenstrukturen der Operation SubscribeServiceInformation	61
7.1.19	Datenstrukturen der Operation UnsubscribeServiceInformation	61
7.1.20	Datenstrukturen der Operation GetServiceStatus	62
7.1.20.1	Request	62
7.1.20.2	Response	62
7.1.21	Datenstrukturen der Operation SubscribeServiceStatus	62
7.1.22	Datenstrukturen der Operation UnsubscribeServiceStatus	62
7.1.23	Datenstrukturen der Operation StartService	62
7.1.23.1	Request	62
7.1.23.2	Response	63
7.1.24	Datenstrukturen der Operation StopService	63
7.1.24.1	Request	63
7.1.24.2	Response	63
7.1.25	Datenstrukturen der Operation RestartService	63

7.1.25.1	Request	63
7.1.25.2	Response	63
7.2	Dienst SystemDocumentationService	63
7.2.1	Operationen des SystemDocumentationService	64
7.2.2	Datenstruktur der Operation GetSystemConfiguration	64
7.2.2.1	Request	64
7.2.2.2	Response	64
7.2.3	Datenstruktur der Operation SubscribeSystemConfiguration	65
7.2.4	Datenstruktur der Operation UnsubscribeSystemConfiguration	65
7.2.5	Datenstruktur der Operation StoreSystemConfiguration	65
7.2.5.1	Request	65
7.2.5.2	Response	65
7.2.6	Datenstruktur der Operation StoreLogMessages	65
7.2.6.1	Request	65
7.2.6.2	Response	66
7.2.7	Datenstruktur der Operation RetrieveLogMessages	66
7.2.7.1	Request	66
7.2.7.2	Response	66
7.3	Dienst SystemManagementService	66
7.3.1	Operationen des SystemManagementService	67
7.3.2	Datenstrukturen der Operation GetDeviceStatus	67
7.3.2.1	Request	67
7.3.2.2	Response	68
7.3.3	Datenstrukturen der Operation SubscribeDeviceStatus	68
7.3.4	Datenstrukturen der Operation UnsubscribeDeviceStatus	68
7.3.5	Datenstrukturen der Operation GetServiceStatus	68
7.3.5.1	Request	68
7.3.5.2	Response	68
7.3.6	Datenstrukturen der Operation SubscribeSystemStatus	69

8	**Besondere IBIS-IP-Datentypen**	**70**
8.1	IBIS-IP.anyURI	70
8.2	IBIS-IP.boolean	70
8.3	IBIS-IP.byte	70
8.4	IBIS-IP.date	70
8.5	IBIS-IP.dateTime	71
8.6	IBIS-IP.double	71
8.7	IBIS-IP.duration	71
8.8	IBIS-IP.int	71
8.9	IBIS-IP.language	71
8.10	IBIS-IP.NMTOKEN	72

8.11	IBIS-IP.nonNegativeInteger	72
8.12	IBIS-IP.normalizedString	72
8.13	IBIS-IP.string	72
8.14	IBIS-IP.time	73
8.15	IBIS-IP.unsignedInt	73
8.16	IBIS-IP.unsignedLong	73
8.17	InternationalTextType	73
9	**Gemeinsam genutzte Aufzählungstypen**	**74**
9.1	ConnectionStateEnumeration	74
9.2	ConnectionTypeEnumeration	74
9.3	DeviceClassEnumeration	74
9.4	DeviceStateEnumeration	75
9.5	ErrorCodeEnumeration	75
9.6	IBIS-IP-VersionEnumeration	75
9.7	MessageTypeEnumeration	75
9.8	ServiceNameEnumeration	75
9.9	ServiceStateEnumeration	76
9.10	SystemDocumentationInformationEnumeration	76
10	**Begriffe**	**77**
Regelwerke – Normen und Empfehlungen		**80**
Bildverzeichnis		**81**
Tabellenverzeichnis		**82**
Impressum		**85**

Versionshistorie

Datum	Verfasser	Änderung

Abkürzungen

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

Abkürzung	Beschreibung
APC	Fahrgastzählsystem (automatic passenger counting system) Beschreibt die Geräteklasse des Fahrgastzählsystems innerhalb des Fahrzeuges. Die technische Umsetzung der Zählung wird nicht unterschieden.
DNS	Domain Name Server
DNS-SD	Domain Name Server Service Discovery
Eeprom	Electrically Erasable Programmable Read-Only Memory
EKAP	Echtzeit Kommunikations- und Auskunftsplattform. Siehe VDV 431-1.
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IETF	Internet Engineering Task Force
RFC	„Reference“ Technisches Degenblatt der Internet Society
SNTP	Simple Network Time Protocol
XSD	XML Schema Definition
DELFI	Durchgängige Elektronische Fahrplaninformation
UML	Unified Modeling Language

1 Einführung in IBIS-IP

Zu Beginn dieser Schrift werden einige einführende Definitionen zur Umsetzung beschrieben. Die Kenntnis des Teil 1 (VDV- Schrift 301-1) wird hierbei vorausgesetzt.

1.1 Anforderungen an Geräte in IBIS-IP

In den nachfolgenden Abschnitten werden die Anforderungen an Geräte und Dienste beschrieben, die sich aus dem im Teil 1 (VDV- Schrift 301-1) skizzierten System ergeben.

Demnach muss ein Gerät folgende Mindestanforderungen erfüllen, um am IBIS-IP teilnehmen zu können:

— eine Ethernet-Schnittstelle

— Erkennung der Einbauposition

— eine TCP/IP- bzw. UDP/IP-Stack Implementierung

— Fähigkeit zur Verarbeitung von HTTP- Protokollen

— Speicher für die gerätespezifische Konfigurationsdaten

— Implementierung der DNS-SD Funktionalität

— Rückwirkungsfreiheit zu angeschlossenen System gewährleisten (falls Verbindung vorhanden)

— Bereitstellen eines *DeviceManagementService* (vgl. Kapitel 3.2)

— Optional: Bereitstellen einer webbasierten herstellerspezifischen Wartungsschnittstelle, die per URL (im *DeviceManagementService*) angesprochen werden kann.

1.2 Konfigurationsparameter

In einem IBIS-IP-System ist ein großer Teil des Systemverhaltens konfigurierbar. So lässt sich durch die Systemkonfiguration festlegen, welche Dienste auf welchen Geräten in welcher IBIS-IP-Version laufen.

Eine IBIS-IP-Systemkonfiguration enthält also Angaben dazu,

— welche Geräte zum System gehören (bestehend aus Geräteklasse (vgl. Kapitel 1.4) und Einbaukennung (vgl. Kapitel 1.3),

— welche Dienste in welcher IBIS-IP-Version zum System gehören und

— auf welchem Gerät welcher Dienst läuft.

Fahrzeugspezifische Angaben (Fahrzeugnummer, Funkadresse und gerätespezifische Konfigurationsparameter) spielen dagegen für das IBIS-IP-System keine Rolle.

1.3 Einbaukennung

Die Ermittlung der Einbaukennung kann durch Auslesen einer Information erfolgen, die am Ort des Geräte-Einbaus hinterlegt ist. Auf welchem technischen Weg dies erfolgt (Steckercodierung, Eeprom, USB-Stick o.ä.), spielt dabei keine Rolle. Der `DeviceManagementService` eines Gerätes kann dann die Information über die Einbaukennung anderen Diensten bekanntmachen und ermöglicht so eine Zuordnung einer technischen Gerätekennung (IP-Adresse, DNS-Name) zu einer fachlichen Gerätekennung (aus Geräteklasse und Einbauposition).

1.4 Geräteklassen

Zur eindeutigen Identifikation der angeschlossenen Geräte im IBIS-IP-System wurden Geräteklassen spezifiziert, die alle derzeit denkbaren Geräte abdecken. Dabei wurde die Klassifizierung bereits in englischer Sprache vorgenommen, da diese Begriffe als „enumeration“ im englisch gehaltenen XML (vgl. Kapitel 7ff) verbreitet werden.

Die in IBIS-IP verwendeten Geräteklassen sind wie folgt definiert:

Geräteklasse	**Gerät**	**Beschreibung**
OnBordUnit	Bordrechner	Entspricht dem Zentralgerät gemäß VDV 300.
Validator	Entwerter	Stempelautomat oder elektronischer Entwerter für E-Tickets.
SideDisplay	Seiten Anzeiger	Beschreibt die seitliche Aussenanzeige an einem Fahrzeug.
FrontDisplay	Front Anzeiger	Entspricht der Fahrzeugzielanzeige.
InteriorDisplay	Innen Displays	Beschreibt die Anzeiger im Innenraum des Fahrzeuges, dies können Fahrgastinformationsanzeiger aber auch digitale Werbeanzeiger sein.
TicketVendingMachine	Ticket Automat	Beschreibt Fahrscheinverkaufsgeräte innerhalb des Fahrzeuges. Dies können fahrerbediente aber auch durch Fahrgäste bediente Automaten sein.
AnnouncementSystem	ELA	Beschreibt die Elektroakustische-Anlage in Fahrzeugen
MMI	Fahrer (Bedien-) Display	Entspricht der Fahrerbedieneinheit mit Anzeigemöglichkeit (z. B. Touch-Display).
VideoSystem	Videoüberwachungssystem	Beschreibt das Videoüberwachungssystem, das über IBIS-IP gesteuert wird bzw. Videodaten überträgt.

Geräteklasse	**Gerät**	**Beschreibung**
APC	Fahrgastzählsystem	Beschreibt das Fahrgastzählsystem innerhalb des Fahrzeuges. Die technische Umsetzung der Zählung wird nicht unterschieden.
MobileInterface	Kundenkommunikations- Schnittstelle	Beschreibt die Schnittstelle, an der die Fahrgäste mit ihrem mobilen Endgerät Informationen vom Fahrzeug abholen können.
TestDevice	Test Geräte	Platzhalter für Testgeräte, die im Falle einer Verifikation des Systems angeschlossen werden.
Other	Sonstige Geräte	Hierunter lassen sich Geräte anbinden, die noch nicht in diesem Standard klassifiziert wurden.

Tabelle 1 IBIS-IP Geräteklassen Klassifikation (Sortierung beliebig)

Die Geräteklasse sorgt zusammen mit der Einbaukennung für eine eindeutige fachliche Identifikation eines Geräts in einem IBIS-IP-System (vgl. Kapitel 5.1.4.1)

1.5 Notationen

Im Sinne einer besseren Lesbarkeit werden im weiteren Verlauf des Dokuments folgende Schriftarten mit spezieller Bedeutung verwendet.

Operationen:

Für Operationen wird folgendes Format verwendet:

Operation in Courier New, Schriftgrad 11 und Fett

Dienste:

Für Dienste wird folgendes Format verwendet:

Dienst in Courier New, Schriftgrad 11 und Kursiv

Syntax:

Eine Syntax wird mit folgendem Format dargestellt:

<Syntax in Courier New, Schriftgrad 11 und mit <> eingefasst>

Pfad:

Ein Pfad wird im folgenden Format dargestellt:

```
Pfad in Courier New, Schriftgrad 11 ohne vorausgehende und
                        folgende /
```

1.6 Version des IBIS-IP

Der IBIS-IP Standard wird mit der VDV-Schrift 301-2 zum ersten Mal beschrieben. Herausgelöst aus dem Forschungs- und Standardisierungsprojekt IP-KOM-ÖV wurde diese Version 1.0 (vgl. Kapitel 7ff) spezifiziert. Um Migrationskonfigurationen verschiedener Versionsstände in einem System zu ermöglichen, wird diese Versionsinformation beim Datenaustausch verwendet. Siehe für weitere Informationen Kapitel 3, 3.3.2 oder das Beispiel in 5.1.4.2.2.

2 Verwendete Kommunikationsprotokolle

In diesem Kapitel wird beschrieben, unter welchen Gesichtspunkten in IBIS-IP entsprechende Kommunikationsprotokolle zum Einsatz kommen. Dazu folgt zunächst eine Grundlagendefinition der unteren Vermittlungsprotokolle. Anschließend folgt eine kurze Betrachtung des fachlichen Kontexts und der daraus folgenden Festlegung des sich in der Applikationsebene befindlichen Kommunikationsprotokolls. Zuletzt wird der Stand von IBIS-IP bzgl. anderer Kommunikationsprotokolle beschrieben.

2.1 Adressierung

2.1.1 IP-Adressen

Die IP-Adressen werden in IBIS-IP dezentral mithilfe eines Teiles des „Zero Conf“ (vgl. RFC 2927 zur automatischen Adressvergabe) vergeben. Eventuelle feste Adressbereiche müssten projektspezifisch festgelegt werden. In Bezug auf Adressbereiche macht die RFC bereits Vorgaben (169.254.xxx.xxx), welche für ein interoperables Netzwerk beachtet werden müssen. Wichtig ist allerdings einzig, dass diese Adressbereiche konsistent zwischen allen Teilnehmern sein müssen.

2.1.2 Subnetzmasken/Gateways

Es ist projektspezifisch festzulegen, welche Adressen verwendet werden sollen.

2.2 Konfigurationsparameter für TCP und UDP

Eine spezielle Anpassung der Kommunikationsparameter der TCP- und UDP-Protokollen muss für IBIS-IP nicht erfolgen. Eine Festlegung der Ports ist insoweit nicht notwendig, als diese Informationen über DNS-SD (vgl. Kapitel 3.3.1) mitgeteilt werden. So kann jede Applikation bzw. jeder Gerätehersteller die Ports frei wählen. Es sollte im Sinne der Standardkonformität auf Ports im Bereich 0-1024 verzichtet werden (vgl. RFC 6335).

2.3 Gültigkeitsdauer von Informationen

Wie in der VDV-Schrift 301-1 beschrieben, lassen sich die Informationen anhand ihrer Gültigkeitsdauer im Wesentlichen in zwei Klassen einteilen. Es gibt

— zyklische Informationen und

— ereignisgesteuerte Informationen.

Wird bei der Umsetzung einer Fachkomponente in Dienste (vgl. Kapitel 3) festgestellt, dass diese Fachkomponente sowohl zyklische als auch ereignisgesteuerte Information bereitstellt, so sind hierfür jeweils unterschiedliche Dienste zu erstellen.

2.3.1 Zyklische Informationen

Es gibt Informationen,

— die sich zyklisch in kurzen Zeitabständen (~sekündlich) ändern bzw. ändern können und

— bei denen ein schneller und gleichzeitiger Transport der Information zu den Konsumenten wichtiger ist als eine gesicherte Informationsübertragung.

Dabei handelt es sich in der Regel um Informationen, die von Geräten oder Schnittstellen zyklisch in kurzen Zeitabständen zur Verfügung gestellt werden. Dazu gehören u. a.

— Uhrzeit (sekündlich)

— Odometer (<=sekündlich)

— GNSS-Koordinaten (<=sekündlich)

Zyklische Informationen ändern sich i. d. R. ohne eine Änderung des betrieblichen oder fachlichen Zustands eines Fahrzeugs zu bewirken. Sie werden i. d. R. von den Diensten der Basisfunktionen bereitgestellt.

Beispiel*:*
In einem Fahrzeug, das sich in Schrittgeschwindigkeit bewegt, ändern sich sekündlich die Odometerwerte, die GPS-Koordinaten und die Uhrzeit. Dennoch kommt es nur gelegentlich zu einer Änderung des betrieblichen oder fachlichen Zustands des Fahrzeugs, z. B. wenn eine neue Haltestelle erreicht wird.

2.3.2 Ereignisgesteuerte Informationen

Es gibt Informationen,

— die sich seltener als sekündlich und ereignisgesteuert ändern,

— die i. d. R. etwas über den fachlichen oder betrieblichen Zustand eines Fahrzeugs aussagen,

— bei denen sichergestellt sein muss, dass auch alle Konsumenten von dieser Änderung zuverlässig erfahren.

Beispiel*:*
Bei Erreichen einer neuen Haltestelle ändert sich der fachliche Zustand eines Fahrzeugs. Es soll sichergestellt sein, dass alle Dienste und Applikationen informiert werden, für die diese Änderung relevant ist, wie die Fahrgast-Innenanzeiger, die Fahrgast-Ansage, das Fahrer-MMI, die mobilen Kundenendgeräte, die Automatische Fahrgastzählung, das itcs, etc.

2.4 Verwendung des UDP- und HTTP-Protokolls

Um den unterschiedlichen fachlichen Kontexten von zyklischen und ereignisgesteuerten Informationen Rechnung zu tragen, werden für IBIS-IP die folgenden Festlegungen getroffen:

— Für die Übertragung von sich zyklisch ändernden Daten wird das UDP-Protokoll verwendet. Die Daten werden dazu in geeigneten zeitlichen Abständen an eine Multicast-Adresse gesendet.
Der entsprechende Dienst wird dann als UDP-Dienst bezeichnet.

Die Wahl der geeigneten zeitlichen Abstände richtet sich nach den fachlichen Anforderungen und den technischen Möglichkeiten der informationsliefernden Schnittstellen. Aufgrund der derzeitigen technischen Gegebenheiten ist eine Zykluszeit unter 1s für UDP-Dienste zu vermeiden.

— Für die Übertragung von Daten, die sich ereignisgesteuert ändern, wird das HTTP-Protokoll verwendet.
Der entsprechende Dienst wird dann als HTTP-Dienst bezeichnet.
Aufgrund dieser Entscheidung muss jedes Gerät, das einen HTTP-Dienst anbietet, die Funktionalität eines eigenen HTTP-Servers haben. Mit HTTP-Diensten steht für die Kommunikation zwischen zwei Teilnehmern eine Punkt-zu-Punkt-Verbindung zur Verfügung. Das Zustandekommen und die sichere Übertragung der Information werden hierbei durch den HTTP-Protokoll-Stack gewährleistet.

2.5 Weitere Protokolle

Für den Zeitserver im System wird das etablierte Standardprotokoll zur Zeitsynchronisation „SNTP“ verwendetet. Das Simple Network Time Protocol (SNTP) erlaubt es, auf einfache Art und Weise mithilfe von UDP die aktuelle Uhrzeit in einem Netzwerk zu verteilen. Weitere Informationen zur Umsetzung siehe VDV 301-2-11.

Bei der Umsetzung weiterer Fachkomponenten in Dienste ist zu erwarten, dass noch weitere IP-basierte Kommunikationsprotokolle zum Einsatz kommen (z. B. das RTP-Protokoll für das Streamen von Audio- und Videodaten). Darüber hinaus gehende Protokolle werden von der Version 1.0 von IBIS-IP nicht berücksichtigt.

3 Veröffentlichung und Kommunikation der Dienste

Applikationen und Dienste innerhalb eines IBIS-IP-Systems kennen

— den Dienstnamen und

— die IBIS-IP-Version

der Dienste, mit denen sie kommunizieren müssen.

Diese Eigenschaften sind Applikations- bzw. Dienst-inhärent, müssen also nicht konfiguriert werden.

Beispiel:
Die Applikation eines Fahrgast-Anzeigers benötigt den Dienst `CustomerInformationService` *(repräsentiert die Fahrgast-Informations-Ermittlung) in der Version X um Fahrgast-Informationsdaten anzeigen zu können. Mit Daten anderer Dienste oder Daten der Version X+1 kann der Fahrgast-Anzeiger nicht umgehen.*

Sie sind jedoch nicht ausreichend, um einen Dienst systemweit eindeutig zu adressieren (vgl. Kapitel 5.1.4.2.1) und damit nutzen zu können. Für eine systemweit eindeutige Findung des Dienstes ist es nötig zu wissen, auf welchem Gerät der erforderliche Dienst läuft. Wo und auf welchem Gerät ein welcher Dienst läuft, ist in der spezifischen Konfiguration eines Systems festgelegt, also nicht Applikations- oder Dienstinhärent.

3.1 Von Fachkomponenten zu Diensten

Ein IBIS-IP-System besteht aus mehreren Fachkomponenten, die miteinander Daten austauschen. Eine Fachkomponente kann

— eine abstrakte Schnittstelle zu einem anderen System oder Gerät,

— eine Applikation,

— ein Dienst oder

— ein Gerät

sein (vgl. VDV-Schrift 301-1).

Alle Applikationen und Dienste eines IBIS-IP-Systems laufen auf Geräten, die miteinander per IP kommunizieren. Dabei stellen IBIS-IP-Applikationen und IBIS-IP-Dienste jeweils softwaretechnische Umsetzungen von Fachkomponenten dar.

Applikationen und Dienste unterscheiden sich dadurch, dass Applikationen im Sinne der IBIS-IP-Systemarchitektur reine Informationsempfänger sind, also keine Daten über eine IBIS-IP-Schnittstelle anderen Diensten oder Applikationen zur Verfügung stellen. Dienste hingegen bieten Schnittstellen zum Datenaustausch für andere Dienste und Applikationen des IBIS-IP-Systems an. Dazu bieten die Dienste eine Reihe von Operationen an. Mit Hilfe dieser Operationen können Daten mit den Diensten ausgetauscht werden oder Aktionen bei Diensten ausgelöst werden.

Jedes Gerät, das an einem IBIS-IP-System teilnimmt und bzgl. seines Status und seiner Erreichbarkeit vom IBIS-IP-System überwacht werden soll, verfügt mindestens über einen Dienst `DeviceManagementService`, der die Fachkomponente Gerätemanagement repräsentiert. Das Gerätemanagement stellt insbesondere Operationen zur Verfügung, mit deren Hilfe weitere Dienste auf einem Gerät gestartet werden können.

Der Austausch der Daten mit den Diensten erfolgt in IBIS-IP in Form von XML-Daten. Je nach fachlichem Kontext erfolgt der Austausch der XML-Datenstrukturen über das HTTP- oder das UDP-Protokoll.

Beispiel:

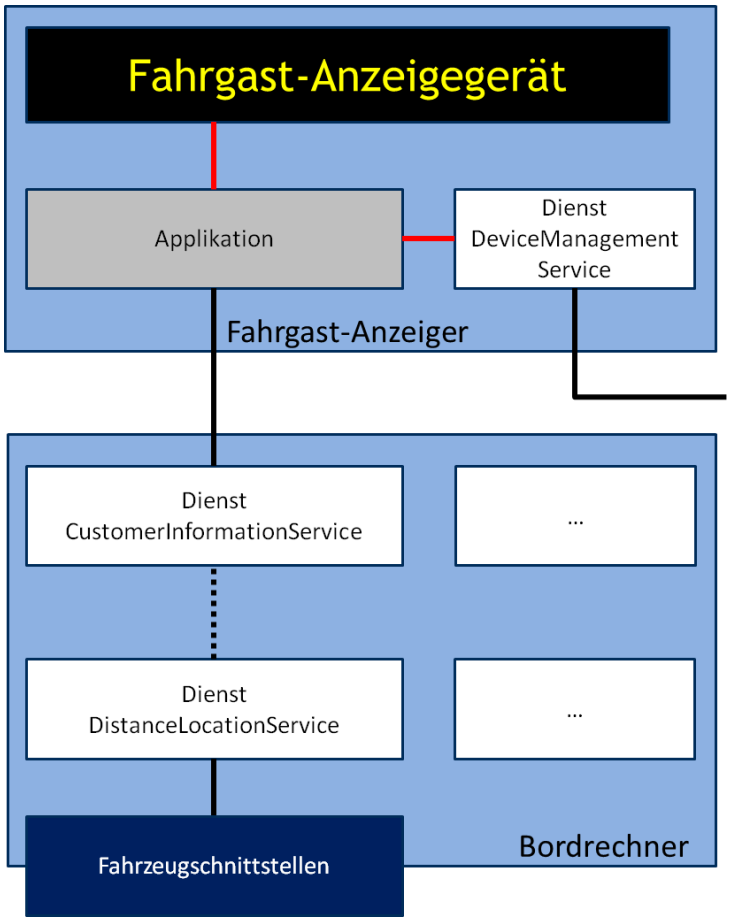

Abbildung 1 Beispiel zu den verschiedenen Repräsentationen von Fachkomponenten. Schwarze Linien: IBIS-IP, rote Linien: proprietär. Man beachte, dass die Fachkomponente Fahrgast-Anzeiger sowohl Geräte-, Applikations- wie auch Diensteigenschaften hat.

An dem System (vgl. Kapitel 1.4) sind zwei Geräte beteiligt

— ein Fahrgast-Anzeiger und

— ein Bordrechner

Auf dem Bordrechner laufen

— der Dienst `CustomerInformationService` (repräsentiert die Fahrgastinformationsermittlung) und

— der Dienst `DistanceLocationService` (repräsentiert einen Teil der physikalischen Ortung)

Der `DistanceLocationService` stellt als Dienst Daten zur Verfügung, die (über mehrere Zwischenschritte) dazu beitragen, die korrekten Informationen für die Fahrgast-Informations-Ermittlung zu ermitteln. Der `DistanceLocationService` bezieht dazu Daten (Odometer-Informationen, GNSS-Positionsdaten) von den Fahrzeugschnittstellen.

Der `CustomerInformationService` stellt als Dienst die jeweils aktuellen Fahrgastinformationen zur Verfügung.

Auf dem Fahrgast-Anzeiger läuft eine Applikation, die

— Daten vom Dienst `CustomerInformationService` abruft und

— über eine proprietäre Schnittstelle auf der Anzeige-Einheit des Fahrgast-Anzeigers entsprechend darstellt.

Die Applikation stellt anderen IBIS-IP-Fachkomponenten keine weiteren Informationen im Rahmen von IBIS-IP zur Verfügung, ist folglich kein Dienst.

Auf dem Fahrgast-Anzeiger läuft ein Dienst `DeviceManagementService`, über den eine Überwachung des Fahrgast-Anzeigers, im Sinne von Applikation und Gerät, durch das IBIS-IP-System erfolgen kann.

3.2 Spezifizierte Dienste

Die gemäss VDV301 spezifizierten Dienste werden fortlaufend angepasst und erweitert. Daher wird eine Übersicht in der VDV-Mittleung 3002 gepflegt.

3.3 Veröffentlichung via DNS-SD

Für die Verwendung und Verbreitung von Informationen zu Diensten wird bei diesem Verfahren auf bekannte Standardmechanismen der Netzwerktechnologie zurückgegriffen. Mit Hilfe von sogenannter „SRV-Records“ (vgl. Kapitel 3.3.1) und „TXT-Records“ (vgl. Kapitel 3.3.2) wird allgemein das Kommunikationsverfahren unter der Bezeichnung DNS-SD (Domain Name System – Service Discovery) zusammengefasst. Es gibt für verschiedene Betriebssysteme kostenfreie Bibliotheken [6], welche das Arbeiten mit dieser Technologie ermöglichen. Bei Verwendung dieser Bibliotheken ist es i. d. R. für den Nutzer transparent, welche Records für die Übertragung welcher Information verwendet werden.

3.3.1 Nutzung des SRV-Records

Aufbauend auf das standardisierte IP-Kommunikationsprotokoll UDP und den Mechanismen, mit denen in IP-basierten Netzen Informationen über Rechnernamen ausgetauscht werden (sogenannten DNS-Records), wurde im RFC 2782 von der Internet Engineering Task Force (IETF) eine Erweiterung dieses Namensaustauschverfahrens speziell für die Bekanntgabe von Diensten spezifiziert.

Mit dieser standardisierten Erweiterung der DNS-Records, den sogenannten SRV-Records, ist es möglich, dass innerhalb von IP-Netzen von den Geräten automatisch bekanntgegeben wird, welche Dienste auf diesen Geräten angeboten werden und wie diese Dienste angesprochen werden können.

Der Aufbau eines solchen SRV-Records folgt dem Muster

`<_Service._Proto.Name TTL Class SRV Priority Weight Port Target>`
und sieht exemplarisch wie folgt aus

```
<CustomerInformationService._ibisip_http.local.  3600  IN  SRV  10
0  389  OnboradUnit_1.local. >
```

Die Bedeutung der Felder ist nachfolgend beschrieben.

Datenfeld (nach RFC 2782)	**Beispielwert**	**Beschreibung**
`Service`	`CustomerInformationService.` (mit . am Ende)	Dienstname (mögliche Werte für Dienste in IBIS-IP finden sich in Kapitel Kapitel 3.2)
`Proto`	`_ibisip_http.` (mit . am Ende)	Kommunikationsprotokoll, in IBIS-IP gibt es nur zwei mögliche Werte: `_ibisip_udp` und `_ibisip_http` (vgl. auch 2.2, 3.4 und 3.5)
`Name`	`local.` (mit . am Ende)	die Domäne des Dienstes
`TTL (time to live)`	`3600`	Gültigkeitsdauer des Records (in Sekunden)
`Class`	`IN`	Klassenbegriff entsprechend RFC 1035, SRV-Records gehören in die `IN`-Klasse, `IN` steht hierbei für Internet
`SRV`	`SRV`	Art des Records, hier also ein `SRV`-Record
`Priority`	`10`	Priorität für den Inhalt (hier also der Dienst) gibt es zwei Dienste gleichen Namens, dann wird der Dienst mit geringerer Priorität bevorzugt
`Weight`	`0`	Gewicht für den Inhalt (hier also der Dienst), bei gleichem Gewicht wird der Dienst mit dem geringeren Gewicht bevorzugt
`Port`	`389`	gibt an, unter welchem Port der bekanntgegebene Dienst angesprochen werden kann
`Target`	`OnboardUnit_1.` (mit . am Ende)	DNS-Name oder IP-Adresse des Zielhostrechners, auf dem der Dienst läuft

Tabelle 2 Bedeutungen der SRV-Records in DNS-SD

3.3.2 Nutzung des TXT-Records

Da nicht alle für IBIS-IP relevanten Informationen für die Dienstidentifikation im Standard SRV-Record untergebracht werden können, werden weitere Informationen mit Hilfe von sogenannten TXT-Records bekannt gegeben (vgl. RFC 1464).

Der TXT-Record erlaubt die Übergabe beliebiger Attribut-Werte-Paare in der Form

```
<Attribut Name>=<Attribut Wert>.
```

In IBIS-IP werden folgende Attribute verwendet:

Attribut	**Optional/ Pflicht**	**Beispielwert**	**Beschreibung**
`ver`	Pflicht für alle IBIS-IP-Dienste	`1.0`	IBIS-IP-Version des Dienstes (vgl. Kapitel 1.6)
`path`	Optional	`testversion_1.1`	Pfadinformation, falls sie vorhanden ist, muss sie zusätzlich zu den Angaben Port und Zielhost aus dem SRV-Record bei der techn. Adressierung eines Dienstes berücksichtigt werden (vgl. Kapitel 5.1.4.2ff, vgl. auch Beispiel. in Kapitel 3.5)
`multicast`	Pflicht für UDP-Dienste	`239.0.0.1`	Angabe der Multicast-Adresse, mit der die Daten des UDP-Dienstes veröffentlicht werden.
`sntp-server`	Pflicht für den Dienst zur Zeitsynchronisation	`192.168.0.22`	Angabe der IP-Adresse unter welcher der SNTP-Server im IBIS-IP-Netzwerk zu finden ist.

Tabelle 3 Bedeutungen der TXT-Records in DNS-SD (inklusive Festlegungen für IBIS-IP)

3.4 Veröffentlichung von UDP-Diensten

UDP-Dienste geben bei Ihrer Veröffentlichung via DNS-SD mindestens folgende dienstspezifische Informationen an:

Im SRV-Record:

— Dienstname

— Protokoll = `_ibisip_udp`

Im TXT-Record:

— `ver`

— `multicast`

Um Informationen der UDP-Dienste zu bekommen, muss der Client der bekanntgegebenen Multicast-Gruppe beitreten. Dort werden von den Diensten die entsprechenden Nachrichten per UDP-Telegramm verbreitet. Mit dem Beitritt zur Multicast-Gruppe können die entsprechenden Telegramme empfangen werden.

Eine explizite Abfrage von Informationen oder auch andere Mechanismen, welche bei den HTTP-Diensten verfügbar sind (siehe Kapitel 2.4), sind in dieser Form nicht vorgesehen.

3.5 Veröffentlichung von HTTP-Diensten

HTTP-Dienste geben bei Ihrer Veröffentlichung via DNS-SD mindestens folgende dienstspezifische Informationen an:

im SRV-Record:

— Dienstname

— Protokoll = `_ibisip_http`

— Port

— Zielhost

Im TXT-Record:

— `ver` (Pflicht)

— `path` (optional)

Beispiel:
Um beispielsweise auf einem Gerät namens „tst01“ unter Port 1234 die Operation ***`GetDeviceConfiguration`*** *des* `DeviceManagementService` *auszuführen, ist die dafür maßgebliche Adresse für die HTTP-Kommunikation*

```
<tst01:1234/DeviceManagementService/GetDeviceConfiguration>
```

Ist zusätzlich im TXT-Record dieses Dienstes ein optionaler „`path=IBIS-IP`“ eingetragen, so lautet die anzusprechende Adresse

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

4 Konventionen für Dienste und Datenstrukturen

In dieser Schrift wird ausschließlich die Kommunikation mit den IBIS-IP-Diensten beschrieben, also insbesondere

— die Strukturierung der Daten, die in IBIS-IP ausgetauscht werden,

— welche Daten(Strukturen) von den Operationen als Ein- und Ausgabeparameter verwendet werden (Request-/Responsestrukturen) und

— welche Operationen durch welche Dienste bereitgestellt werden.

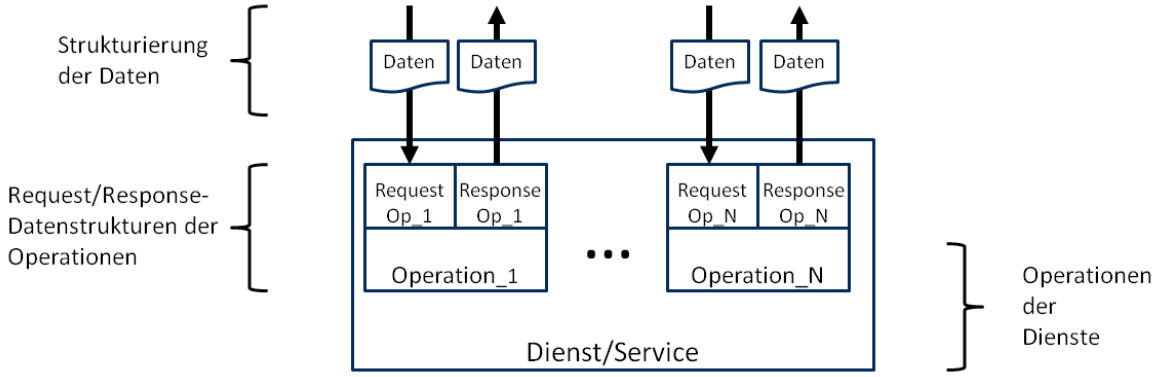

Abbildung 2 Darstellung der Strukturierung von Daten in Diensten

Dabei erfolgt sowohl eine Darstellung allgemeiner Prinzipien der Dienste-Kommunikation als auch die Beschreibung konkreter Umsetzungen von Fachkomponenten in Diensten.

Geräte werden dabei berücksichtigt,

— weil sie für den Kontext, in dem ein Dienst läuft, relevant sind und

— weil sie durch das IBIS-IP-System überwacht werden.

Diese Schrift behandelt **nicht**

— die Umsetzung von Fachkomponenten in Form einer Applikation und

— die abstrakten Schnittstellen zu andern Systemen und Geräten.

4.1 Konventionen für HTTP-Dienste

HTTP-Dienste in IBIS-IP stellen ihre Funktionalität über Operationen zur Verfügung. Die Operationen dienen dazu, Aktionen beim Dienst auszulösen, die bestimmte Objekte betreffen, die der Dienst verwaltet.

4.1.1 Verwendung von HTTP-POST und HTTP-GET

IBIS-IP-Dienste bieten entweder Operationen an, denen

— Daten übergeben werden. In diesem Fall erfolgt der Operationsaufruf immer via HTTP-POST

— keine Daten übergeben werden. In diesem Fall erfolgt der Operationsaufruf immer via HTTP-GET.

4.1.1.1 Operationsaufrufe mit HTTP-POST

Operationsaufrufe mit HTTP-POST werden immer dann verwendet, wenn dem Dienst Daten übergeben werden sollen.

Dabei ist durch den Dienst festgelegt, welche Datentypen bzw. Datenstrukturtypen vom Dienst akzeptiert und interpretiert werden können.

Beispiel:
Mit Hilfe der Operation ***`SetBlockNumber`*** *des Dienstes* `JourneyInformationService`*, kann ein neuer Umlauf beim* `JourneyInformationService` *eingestellt werden. Dazu wird dem* `JourneyInformationService` *eine entsprechende BlockNumber-Datenstruktur als Parameter übergeben.*

Der Aufruf erfolgt als HTTP-POST.

4.1.1.2 Operationsaufrufe mit HTTP-GET

Operationsaufrufe mit HTTP-GET werden immer dann verwendet, wenn keine Übergabe von Daten an den Dienst erfolgt, d. h. wenn es sich um eine reine Abfrage von Daten handelt.

Sämtliche HTTP-GET Operationen können insbesondere über einen passenden Adress-Eintrag mit einem Browser abgerufen werden.

Beispiel:
Mit Hilfe der Operation ***`GetDeviceStatus`*** *des Dienstes* `DeviceManagementService`*, wird der Status des Geräts, auf dem der* `DeviceManagementService` *läuft, ermittelt. Ein Parameter ist bei dieser Abfrage nicht erforderlich.*

Der Aufruf erfolgt als HTTP-GET.

Die Abfrage kann mit einem Browser durch Eingabe der Adresszeile
`[…] DeviceManagementService/GetDeviceStatus`
erfolgen.

Aufgrund der Namenskonvention aus kapitel 4.1.3 gilt insbesondere die Regel, dass für sämtliche **`Get`**-Operationen ein HTTP-GET verwendet wird.

4.1.2 Namenskonventionen für Operationen

Operationsnamen enthalten mindestens ein Verb, das die Aktion beschreibt und ein Objekt, auf das sich die Aktion bezieht.

Folgende Verben finden bislang in unterschiedlichen Bereichen Verwendung:

Bei rein datenorientierten Operationen, sind dies die Verben

— Set,

— Retrieve,

— Get,

— List,

— Subscribe und

— Unsubscribe.

Bei Steuerungsoperationen, insb. des `DeviceManagementService`, sind dies die Verben

— Start,

— Stop,

— Restart,

— Activate und

— Deactivate.

Für Validierungsoperationen ist es zusätzlich das Verb

— Validate.

Um zu verhindern, dass gleiche Verben für verschiedene Aktionen oder verschiedene Verben für gleiche Aktionen verwendet werden, wurden kurze Konventionen für den Einsatz der Verben erarbeitet.

Die Konstruktion der Dienstnamen und die Konventionen zur Verwendung der Verben werden nachfolgend erläutert.

4.1.2.1 Rein datenorientierte Operationen

Sofern es sich bei den Objekten um Daten bzw. Datenstrukturen des Dienstes handelt, gilt Operationsname = Verb + betreffende Datenstruktur

Beispiel*:*
Die Operation zur Abfrage der Gerätekonfiguration heißt
***`GetDeviceConfiguration`** = Get (Verb) + DeviceConfiguration (betreffende Datenstruktur).*

Die Daten können alle Arten von fachlichen Daten sein, z. B.

— Konfigurationsdaten,

— Ortungsdaten,

— Fahrgastinformationsdaten,

— Statusinformationen,

— etc.

Die datenorientierte Aktionen bezwecken in der Regel

— das Übergeben von Daten bzw. Datenstrukturen ODER

— das einmalige Abfragen von Daten bzw. Datenstrukturen ODER

— das Abonnieren auf Daten bzw. Datenstrukturen ODER

— das Beenden eines Abonnements auf Daten bzw. Datenstrukturen.

Die einzelnen Aktionen werden nachfolgend näher beschrieben.

4.1.2.1.1 Einstellen von Werten bei einem Dienst

Konvention:

Operationen, die dazu dienen, einen dienstinternen Wert festzulegen, beginnen mit dem Verb

Set (setzen, einstellen).

Der Name einer solchen Übergabe-Operation setzt sich außerdem aus einer Beschreibung des Daten- bzw. Datenstrukturobjekts zusammen, das von der Übergabe betroffen ist.

Beispiel:
Die Operation zur Einstellung der Gerätekonfiguration heißt
`SetDeviceConfiguration`

4.1.2.1.2 Einmalige Abfrage von aktuell gültigen Daten bzw. Datenstrukturen

Konvention:
Operationen, die der Abfrage von aktuell gültigen, dienstinternen Daten bzw. Datenstrukturen dienen, beginnen mit dem Verb

Get (erhalten).

Der Name einer solchen Abfrage-Operation setzt sich außerdem aus einer Beschreibung des Daten- bzw. Datenstrukturobjekts zusammen, das abgefragt wird.

Beispiel:
Die Operation zur Abfrage des Gerätestatus heißt
`GetDeviceStatus`

4.1.2.1.3 Abonnieren auf Daten bzw. Datenstrukturen

Damit ein Dienstnutzer von Änderungen einer Datenstruktur eines Dienstes erfährt, kann der Dienstnutzer den Dienst zyklisch mit Hilfe der Get-Abfragen abfragen. Solange sich der Wert der Datenstruktur nicht ändert, werden dabei unnötig Daten ausgetauscht. Daher ist in IBIS-IP durch die Verwendung von Abonnements ein solches Verfahren zu vermeiden.

Durch ein Abonnement auf eine Datenstruktur eines Dienstes wird der Abonnent automatisch vom Dienst über eine Änderung der Datenstruktur informiert.

Konvention:

Operationen, die der Einrichtung eines Abonnements dienen, beginnen mit dem Verb

Subscribe (abonnieren, anmelden).

Beispiel:
Die Operation zur Einrichtung eines Abonnements auf den Status eines Gerätes heißt
`SubscribeDeviceStatus`

4.1.2.1.4 Beenden eines Abonnements auf Daten bzw. Datenstrukturen

Um ein eingerichtetes Abonnement zu beenden, bedarf es einer weiteren Operation.

Konvention:

Operationen, die der Beendigung eines Abonnements dienen, beginnen mit dem Verb

Unsubscribe (abbestellen, abmelden).

Beispiel:
Die Operation zur Beendigung eines Abonnements auf den Status eines Gerätes heißt
`UnsubscribeDeviceStatus`

4.1.2.1.5 Operationen zur Abfrage von spezifischen Informationen

Dienste können nicht nur Operationen bereitstellen, mit denen man auf die Werte aktuell gültiger Daten bzw. Datenstrukturen des Dienstes zugreifen kann, sondern auch Operationen, mit denen weitere spezifische Daten ermittelt werden, die nicht Teil des aktuell gültigen Datenstandes sind. Ein Abonnement auf derartige Daten ist nicht sinnvoll und ist deshalb nicht vorgesehen.

Konvention:

Operationen, die dazu dienen, spezifische Informationen zu ermitteln, die von einem übergebenen Parameter abhängen, beginnen mit dem Verb

Retrieve (abfragen, abrufen)

Wird kein Parameter übergeben, so wird eine Liste aller passenden Informationen in der Antwortstruktur zurückgegeben.

Beispiele:
Die Operation zur Ermittlung aller Informationen zu einem bestimmten Umlauf heißt
`RetrieveSpecificBlockInformation.`

Die Operation zur Ermittlung aller (IBIS-IP verlangt die letzten 10) Logdaten-Einträge heißt
`RetrieveLogMessages`

4.1.2.1.6 Operationen zur Abfrage von Datenlisten aus der Grunddatenversorgung

Konvention:

Operationen, die dazu dienen, alle im System vorhandenen Elemente einer bestimmten Information zu ermitteln, beginnen mit dem Verb

List (auflisten)

Der Name einer solchen Abfrage-Operation setzt sich außerdem aus einer Beschreibung des Daten- bzw. Datenstrukturobjekts zusammen, das abgefragt wird.

Beispiel:
Die Operation, um alle dem System bekannten Routen zu erhalten heißt
`ListAllRoutes`

Da bei diesen Operationen mit der Anfrage keine Parameter übergeben werden können, gelten die in Kapitel 4.1.1.2 dargestellten Möglichkeiten.

4.1.2.2 Steuerungsoperationen

Neben den in Kapitel 4.1.1 beschriebenen datenorientierten Operationen gibt es Operationen, bei denen das Objekt der Operation nicht Daten oder Datenstrukturen des Dienstes sind, sondern mit denen das Verhalten eines Dienstes oder Geräts gesteuert wird. Dies ist insbesondere beim Gerätemanagement der Fall.

Beim Gerätemanagement können über Operationen Dienste auf dem Gerät oder gar das gesamte Geräte gestoppt bzw. neu gestartet werden. Das Objekt ist in diesem Fall das Gerät (Device) oder der Dienst (Service).

Die Namensgebung der Dienste folgt dabei dem oben beschriebenen Muster aus Verb und Objekt.

4.1.2.2.1 Starten

Operationen, die dem Starten eines Prozesses dienen, beginnen mit dem Verb

Start

Beispiel:
Die Operation zum Start eines Dienstes auf einem Gerät heißt
`StartService`.

4.1.2.2.2 Neustart auslösen

Operationen, die dem Neustart eines Prozesses dienen, beginnen mit dem Verb

Restart

Beispiel:
Die Operation, mit dem der Neustart eines Gerätes ausgelöst wird, heißt
`RestartDevice`.

4.1.2.2.3 Stoppen

Operationen, die dem Stoppen eines Prozesses dienen, beginnen mit dem Verb

Stop

Beispiel:
Die Operation zum Stoppen eines Dienstes auf einem Gerät heißt
`StopService`.

4.1.2.2.4 Deaktivieren eines Gerätes

Operationen, die dem Deaktivieren eines Geräts dienen, beginnen mit dem Verb

Deactivate

Beispiel:
Die Operation zum Deaktivieren eines Geräts heißt
`DeactivateDevice`

Ein über „Deactivate“ deaktiviertes Gerät kann nur durch Aufruf der Activate-Operation wieder in seinen vorherigen Betriebszustand zurückkehren.

4.1.2.2.5 Aktivieren eines Gerätes

Operationen, die dem Aktivieren eines Geräts dienen, beginnen mit dem Verb

Activate

Beispiel:
Die Operation zum Aktivieren eines Geräts heißt
`ActivateDevice`

4.1.2.3 Operationen zur Gültigkeitsprüfung

Operationen, die dazu dienen, einen übergebenen Datensatz auf Gültigkeit zu prüfen, beginnen mit dem Verb

Validate

Beispiel:
Die Operation zur Prüfung der Gültigkeit eines Ticketdatensatzes heißt

`ValidateTicket`

4.1.3 Konvention zu Get/Subscribe/Unsubscribe

Zu allen Daten bzw. Datenstrukturen, die man mit Hilfe einer **`Get`**-Operation abrufen kann, existiert auch eine Subscribe- bzw. Unsubscribe-Operation.

Zu allen Daten bzw. Datenstrukturen, auf die man mit Hilfe einer **`Subscribe`**-Operation ein Abonnement einrichten kann, existiert auch eine **`Unsubscribe`**-Operation zur Beendigung des Abonnements und eine **`Get`**-Operation zur einmaligen Abfrage des Werts der Datenstruktur.

Zu jeder **`Unsubscribe`**-Operation auf eine Datenstruktur gehört auch eine **`Subscribe`**-Operation sowie eine **`Get`**-Operation auf dieselbe Datenstruktur.

4.1.4 Unterscheidung zwischen Get und Retrieve

Mit **`Get`**-Operationen greift man immer auf aktuell gültige Daten zu. Der Zugriff erfolgt ohne die Übergabe von Daten oder Datenstrukturen, weil durch den Objektnamen der Operation festgelegt ist, mit welcher Datenstruktur der Dienst antwortet. Deshalb können hierfür auch Abonnements eingerichtet werden.

`Retrieve`-Operationen beziehen sich nicht auf aktuell gültige Daten, sondern übergeben immer einen Datensatz, von dessen Inhalt die Antwort des Dienstes abhängt.

4.1.5 Konvention zu Deactivate/Activate

Ein durch eine **`Deactivate`**-Operation deaktiviertes Gerät zeichnet sich dadurch aus, dass es sich im Rahmen der IBIS-IP-Kommunikation, wie ein abgeschaltetes Gerät verhält. D. h. insbesondere, dass keiner der Dienste auf dem Gerät mehr via dem IBIS-IP-Protokoll kommuniziert.

Durch die **`Activate`**-Operation wird ein zuvor deaktiviertes Gerät wieder in Zustand versetzt, den es nach dem Starten des Gerätes hat. Insbesondere sind keine Dienste außer dem *`DeviceManagementService`* nach Ausführung des Activate-Befehls aktiv.

4.2 Konventionen für UDP-Dienste

UDP-Dienste werden wie alle fachlichen Dienste, durch das Systemmanagement gestartet. Dabei veröffentlichen UDP-Dienste ihre Daten an eine Multicast-IP-Adresse. Andere Dienste oder Applikationen, die diese Daten abrufen wollen, melden sich an der Multicast-IP-Adresse an und empfangen somit zyklisch neue Daten. Eine Adressierung des Dienstes oder einzelner Operationen dieses Dienstes gibt es nicht.

4.3 Konventionen für besonderes Dienstverhalten

4.3.1 Konventionen für Zustände

Dienste in IBIS-IP können in einem der folgenden Zustände sein:

- **notrunning**
  In diesem Zustand befinden sich alle Dienste bei Systemstart. Sie bleiben insbesondere in diesem Zustand, sofern sie nicht durch den *`SystemManagementService`* gestartet werden.
- **running**
  In diesen Zustand gelangt ein Dienst erst dann, wenn

- er selbst sämtliche Daten von anderen Diensten beziehen kann, von denen er abhängt und
- wenn er daraus gültige Daten ermitteln und bereitstellen kann und
- wenn er als Dienst per DNS-SD veröffentlicht wurde.

— **defective**
In diesem Zustand gelangt ein Dienst, wenn er sich
- auch nach Ablauf interner Timeouts nicht mit allen notwendigen Diensten verbinden kann oder
- wenn keine gültigen Daten mehr bereitgestellt werden können (z. B. durch einen Verbindungsabbruch zu erforderlichen Diensten oder weil die gelieferten Daten nicht verwertbar sind)

— **starting**
In diesem Zustand befindet sich ein Dienst, der gestartet wurde, aber noch nicht den Zustand „running“ erreicht hat. Sollte nach Ablauf eines Timeouts der Zustand „running“ nicht erreicht werden, so wird der Zustand „defective“ gemeldet.

— standby
In diesem Zustand gelangt ein Dienst, wenn
- er selbst sämtliche Daten von anderen Diensten beziehen kann, von denen er abhängt und
- wenn er daraus gültige Daten ermitteln und bereitstellen kann und
- wenn er NICHT als Dienst per DNS-SD veröffentlicht wurde.

Der Status der Dienste kann individuell beim jeweiligen `DeviceManagementService` des Gerätes abgefragt werden, auf dem der Dienst läuft. Er wird außerdem zyklisch durch den SystemManagementService abgefragt.

4.3.2 Konventionen für Fehlermeldungen

Innerhalb von IBIS-IP gibt es verschiedene Ebenen, auf denen in der Kommunikation Fehler auftreten können und die je nach Ebene unterschiedlich gehandhabt werden.

Kommt es auf der Ebene der Netzwerkkommunikation zu Fehlern, so kommen die Mechanismen der Fehlerbehandlung von TCP bzw. HTTP und UDP zum Einsatz. Wird also beispielsweise eine Anfrage an eine falsche Adresse geschickt, dann wird dies mit einem normalen HTTP-Fehler 404 „File not found“ beantwortet.

Kommt die Netzwerkkommunikation erfolgreich zustande, wird aber für eine Anfrage ein fehlerhaftes XML geschickt, dann wird dies von dem angefragten Dienst mit einer XML-Nachricht beantwortet, in der sich weitergehende Informationen zu der fehlerhaften Anfrage finden. Diese Informationen werden in Form eines frei wählbaren Textes übertragen.

Hat ein Dienst nicht alle Grunddaten, die er zu einer vollständigen Antwort benötigt oder fehlen ihm Dienste, auf die seine eigenen Informationen aufbauen, so ist es in IBIS-IP vorgesehen, dass er trotzdem eine syntaktisch korrekte Antwort gibt, auf die fehlerhalten oder fehlenden Daten jedoch mit einem weiteren Attribut hinweist. Dafür wurden in IBIS-IP die XML-Standarddatentypen durch eigene IBIS-IP-Typen ersetzt, die neben dem eigentlichen Wert auch noch einen Fehlercode enthalten können. Kommt es beispielsweise in einem IBIS-IP-System zu einem Ausfall der GPS-Ortung, weil das Fahrzeug sich in einem Tunnel befindet, so kann der Dienst `NetworkLocationService` trotzdem weiterhin Daten senden, allerdings werden die übermittelten Werte im XML in diesem Fall mit einem ErrorCode „DataEstimated“ versehen, so

dass der datenkonsumierende Dienst in diesem Fall die Möglichkeit hat, die gelieferten Informationen zu verwerfen oder sie mit einem entsprechenden Hinweis versehen, weiterzugeben.

4.4 Konventionen für die Datenstrukturierung

Um die Daten innerhalb von IBIS-IP über XML-Schema-Dateien in einer einheitlichen Art und Weise abzubilden, gelten die folgenden Konventionen für die Beschreibung der XML-Strukturen für Operationen:

— Besitzt eine Operation spezifische Frage- und Antwort-Strukturen, so sind diese nach folgendem Schema zu benennen

  — für Anfragen: Dienstname + „.“ + Operationsname + „RequestStructure“

  — für Antworten: Dienstname + „.“ + Operationsname + „ResponseStructure“

Beispiel:
<JourneyInformationService.RetrieveAllRoutesPerLineRequestStructure>

<JourneyInformationService.RetrieveAllRoutesPerLineResponseStructure>

— Die Antwort-Strukturen bestehen immer aus einem „choice“ (Auswahlmöglichkeit im XML) zwischen dem eigentlichen Dateninhalt und einer freien Fehlermeldung. Hierbei ist der eigentliche Dateninhalt entsprechend zu benennen und eine Struktur dafür anzulegen, die nach dem Schema

  — Dienstname + „.“ + Dateninhalt + „Data“

  zu benennen ist.

Beispiel:
<JourneyInformationService.AllRoutesData>

5 Prinzipien in IBIS-IP

Der Systemstart, das anschließende Starten aller Dienste entsprechend der Konfiguration (vgl. Kapitel 1.2) sowie das Finden der Dienste untereinander erfolgt in IBIS-IP weitgehend automatisch.

Die Findung des Gesamtsystems beruht auf verschiedenen grundlegenden Prinzipien, die in IBIS-IP angewandt werden. Diese Prinzipien werden nachfolgend kurz beschrieben.

5.1 Grundlagen

5.1.1 Systemkonfiguration muss vorliegen

In der Systemkonfiguration wird auf rein fachlicher Ebene festgelegt

- welche Geräte überhaupt am IBIS-IP-System beteiligt sind. Durch diese Information kann eine Fehlermeldung generiert werden, falls eines der Geräte nicht erreichbar ist.
- welcher Dienst auf welchem Gerät in welcher IBIS-IP-Version laufen soll. Mit dieser Information kann der SystemManagementService dafür sorgen, dass auf den Geräten die korrekten Dienste gestartet werden.

Dabei ist zu beachten, dass es nicht erforderlich ist, explizit in der Systemkonfiguration anzugeben, dass ein `DeviceManagementService` (vgl. VDV 301-2-2 auf einem Gerät läuft. Die Angabe eines Gerätes impliziert das Vorhandensein eines `DeviceManagementService` (vgl. Kapitel 1.1)

Beispiel:
In der Systemkonfiguration wird festgelegt, dass der `CustomerInformationService` *(vgl.* VDV 301-2-3*) in der IBIS-IP-Version 1.0 auf dem Gerät OnBoardUnit 1 (vgl.* Kapitel *1.6) laufen soll.*

Für die Konsistenz der Konfiguration ist der Ersteller der Konfiguration verantwortlich. Fehlen Dienste oder Geräte in der Konfiguration, oder werden IBIS-IP-Versionen für Dienste angegeben, mit denen andere Dienstnutzer nicht umgehen können, so wird das System nicht wie geplant funktionieren. Werden entgegen der IBIS-IP Spezifikation Dienste mit gleicher IBIS-IP-Version doppelt (z. B. auf verschiedenen Geräten) gestartet, so kann dies zu Inkonsistenzen führen (vgl. auch Kapitel 5.1.2). Ein solcher Zustand ist daher strikt zu vermeiden.

Die Systemkonfiguration wird vom `SystemDocumentationService` (vgl. VDV 301-2-4) über die Operation ***`GetSystemConfiguration`*** bereitgestellt. Damit können sowohl das Ablageformat als auch der Ablageort der Systemkonfigurationsdaten herstellerspezifisch implementiert werden, die Inhalte sind jedoch über eine Schnittstelle des `SystemDocumentationService` ansprech- und änderbar.

5.1.2 Nur ein Dienst pro Fachlichkeit

In IBIS-IP darf nur genau ein Dienst für eine fachliche Aufgabe im System gestartet sein. Damit werden Zustände vermieden, in denen es durch mehrfach vorhandene Dienste zu Mehrdeutigkeiten kommt.

Beispiel:
Würden in einem System zwei `CustomerInformationServices` *laufen, so wäre es (aufgrund von* Kapitel *5.1.3) nicht klar, welcher Fahrgast-Anzeiger seine Informationen von welchem der beiden* `CustomerInformationServices` *holt. Insbesondere kann es zu Zuständen kommen, in denen zwei Anzeiger zumindest zwischenzeitlich verschiedene Informationen anzeigen.*

Die bedeutet nicht, dass es nicht zwei Dienste gleichen Typs in einem IBIS-IP-System geben darf. Sie müssen aber sowohl fachlich verschiedene Aufgaben im IBIS-IP-System übernehmen als auch auf verschiedenen technischen Wegen adressierbar sein.

Beispiel 1:
In einem IBIS-IP-System mit 5 Geräten, gibt es auch fünf `DeviceManagementServices` *Diese erfüllen jedoch fachlich verschiedene Aufgaben (sie managen verschiedene Geräte) und sie sind technisch auf verschiedenen Wegen adressierbar.*

Beispiel 2:
Wenn in einem Fahrzeug mit drei Türen drei Fahrgastzählgeräte angebracht sind, so kann jedes dieser Geräte einen Dienst zur Abholung der Fahrgastzahlen bereitstellen. Die Geräte erfüllen fachlich verschiedene Aufgaben (sie zählen Ein-/Aussteiger an verschiedenen Türen) und sind technisch auf verschiedenen Wegen adressierbar.

Beispiel 3:
Wenn in einem IBIS-IP-System ein älterer Fahrgastanzeiger ausschließlich mit Daten des `CustomerInformationService` *der IBIS-IP-Version 1.0 umgehen kann, man aber gleichzeitig auf einem moderneren Fahrgastanzeiger Features des* `CustomerInformationService` *der IBIS-IP-Version 2.0 nutzen will, so ist es möglich, zwei* `CustomerInformationServices` *im IBIS-IP-System zu betreiben. Die Dienste erfüllen fachlich verschiedene Aufgaben und sind technisch auf verschiedenen Wegen adressierbar. Die Adressierung verschiedener Versionen eines Dienstes auf einem Gerät erfolgt durch Verwendung verschiedener Ports oder verschiedener Pfade für die verschiedenen Versionen des Dienstes.*

Die Informationen über die in einem IBIS-IP vorhandenen Geräte bzw. alle benötigten Dienste in den benötigten IBIS-IP-Versionen werden vom `SystemDocumentationService` bereitgestellt.

Eine Sonderrolle spielt der `SystemManagementServivce`, der tatsächlich nur genau ein einziges Mal im gesamten IBIS-IP-System vorkommen darf (also auch in nur genau einer IBIS-IP-Version). Auf diese Weise wird sichergestellt, dass nicht durch den Parallelbetrieb zweier `SystemManagementServices` zwei parallele IBIS-IP-Systeme verwaltet werden.

5.1.3 Kenntnis der erforderlichen Dienste

In der Systemkonfiguration wird NICHT festgelegt, welche Applikation bzw. Dienst sich mit welchem Dienst verbinden soll. Die Information, von welchen anderen Diensten mit welcher IBIS-IP-Version eine Applikation oder ein Dienst abhängig ist, ist immer Teil der Implementierung der Applikation bzw. des Dienstes selbst.

Der Dienst bzw. die Applikation verbindet sich immer mit dem Dienst, der als erstes mit passendem Dienstnamen und passender IBIS-IP-Version via DNS-SD gefunden wird (vgl. Kapitel 3.1und 5.1.4.2). Deshalb ist es wichtig, durch die Systemkonfiguration sicherzustellen, dass es nur genau einen Dienst pro Fachlichkeit gibt.

Wenn ein Dienst bzw. eine Applikation Daten verschiedener IBIS-IP-Versionen eines anbietenden Dienstes handhaben kann, so ist auf Seiten des nutzenden Dienstes bzw. der nutzenden Applikation festzulegen, welche Version zu verwenden bzw. zu bevorzugen ist.

Beispiel:
Die Applikation auf einem Fahrgast-Anzeiger ist so implementiert, dass sie immer nach einem Dienst `CustomerInformationService` *in der IBIS-Version 2.0 sucht und von diesem Dienst Daten abruft und anzeigt, sobald der Dienst für die Applikation auf dem Fahrgast-Anzeiger verfügbar ist.*

5.1.4 Identifikation von Diensten und Geräten in IBIS-IP

Nachfolgend wird beschrieben, wie Geräte und Dienste in IBIS-IP identifiziert werden.

Dabei wird unterschieden zwischen einer fachlichen Identifikation und einer technischen Identifikation.

Fachliche Identifikation

Die fachliche Identifikation dient der Beschreibung und Konfiguration des Systems. Hierbei liegt der Fokus auf einer verständlichen Darstellung des Systems. Geräte und Dienste können mit Hilfe der fachlichen Identifikation konfiguriert werden, ohne dass dabei die spezifische technische Umsetzung, also IP-Adressen, Ports oder Pfade der Dienste, bekannt sein müssen.

Fachliche Identifikationsdaten werden insbesondere bei Systemstart zwischen dem `SystemManagementService` und dem `SystemDocumentationService` ausgetauscht, um die System-Konfiguration des IBIS-IP-Systems zu ermitteln und die korrekten Dienste zu starten. Fachliche Identifikationsdaten sind also für den System- bzw. Dienststart relevant.

Technische Identifikation

Die technische Identifikation hingegen dient der IP-basierten Kommunikation mit den Diensten in IBIS-IP. Technische Identifikationsdaten lassen sich per DNS-SD ermitteln und sind für die Dienstnutzung relevant.

5.1.4.1 Geräte

Die Zuordnung von fachlicher zu technischer Identifikation eines Gerätes erfolgt durch den `SystemManagementService`. Der `SystemManagementService` ist der einzige Dienst, der überhaupt Geräte überwacht und managt (vgl. Kapitel 7.3) bzw. eine Zuordnung zwischen fachlicher und technischer Identifikation vornimmt.

Alle anderen Dienste und Applikationen kommunizieren nur mit Diensten, die sich per DNS-SD anhand ihrer technischen Identifikation adressieren lassen, d. h., insbesondere ohne Berücksichtigung der Geräte, auf denen die Dienste laufen oder sonstiger fachlicher Identifikationsdaten.

5.1.4.1.1 Fachliche Identifikation

Ein Gerät wird in IBIS-IP systemweit eindeutig durch

— einen Gerätetyp und

— eine Geräte-ID

identifiziert.

Mögliche Gerätetypen finden sich in Abschnitt 1.4 aufgelistet.

Eine Geräte-ID ist eine eindeutige Identifikationsnummer, die einen Rückschluss auf den Einbauort des Gerätes erlaubt. Die Kombination aus Gerätetyp und Geräte-ID erlaubt eine systemweit eindeutige fachliche Identifikation.

Beispiel:
Ein Entwerter-Gerät (Validator), der sich an der Einbauposition 3 befindet, kann fachlich als Entwerter 3 (Validator 3)bezeichnet werden.

5.1.4.1.2 Technische Identifikation

Technisch kann ein Gerät innerhalb des Netzwerkes über

— seine systemweit eindeutige IP-Adresse oder

— seinen systemweit eindeutigen DNS-Namen

identifiziert werden.

5.1.4.2 Dienste

5.1.4.2.1 Fachliche Identifikation

Ein Dienst wird systemweit fachlich eindeutig identifiziert durch

— Dienstname

— IBIS-IP-Version

— Gerätetyp und

— Geräte-ID.

Das Konzept erlaubt es insbesondere, dass mehrere Versionen eines IBIS-IP-Dienstes auf einem Gerät vorhanden sind und parallel in Betrieb sind. (Ausnahme SystemManagementService, vgl. Kapitel 7.3)

Beispiel:
Der `CustomerInformationService` *mit IBIS-IP-Version 1.0 auf der Geräteklasse OnBoardUnit mit Geräte ID 1 ist systemweit eindeutig identifiziert.*

5.1.4.2.2 Technische Identifikation

Ein Dienst wird systemweit technisch eindeutig identifiziert durch

— die systemweit eindeutige IP-Adresse oder den systemweit eindeutigen DNS-Namen des Gerätes auf dem der Dienst läuft

— den Dienstnamen,

— den Ports und

— (*optional*) des Pfads, unter dem der Dienst ansprechbar ist.

Verschiedene Versionen eines Dienstes müssen von einem Gerät dabei unter verschiedenen Ports oder verschiedenen Pfaden veröffentlicht werden.

Der Protokolltyp (_ibisip_udp bzw. _ibisip_http, vgl. Kapitel 3.4 und 3.5) wird nicht zur Identifikation herangezogen, da der Protokolltyp sich direkt aus dem Dienst ergibt und ein Dienst nur von einem der beiden Typen sein kann.

Beispiel 1:
Der Dienst `CustomerInformationService` *aus VDV 301-2-3 bietet seine Operationen der IBIS-IP-Version 1.0 unter dem Port 34567 und dem Pfad „/1.0/“ an. Operationen der IBIS-IP-Version 2.0 werden dagegen unter dem Port 34567 und dem Pfad „/2.0/“ angeboten.*

Dabei ist der `CustomerInformationService` *immer ein _ibisip_http-Dienst.*

Beispiel 2:
Der `CustomerInformationService` *aus VDV 301-2-3 wird technisch systemweit eindeutig identifiziert durch*

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

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

wobei `192.168.47.11` *die IP-Adresse*, `vdv-test-dev` *der zugehörige DNS-Name,* `34567` *der Port*, `1.0` *der Pfad und* `CustomerInformationService` *der Dienstname ist.*

5.2 Das Konzept des Systemstarts

Der Start eines IBIS-IP-Systems ist ein zweistufiger Prozess. Bei beiden Stufen werden im Wesentlichen folgende Teilschritte durchlaufen:

— Start der Dienste

— Veröffentlichung der Dienste

— Verbindungsaufbau mit den veröffentlichten Diensten und Ausführung fachlicher Aufgaben

In der ersten Stufe erfolgt der Teilschritt „Start der Dienste“ automatisch beim Start der Geräte. Es werden dabei nur die systemrelevanten Dienste der Fahrzeugbetriebsfunktionen“ (`DeviceManagementService`, `SystemManagementService` und `SystemDocumentationService`) gestartet.

In der zweiten Stufe wird der Teilschritt „Start der Dienste“ durch den dann laufenden `SystemManagementService` gesteuert.

5.2.1 Stufe 1: Start systemrelevanter Fahrzeugbetriebsfunktionen

5.2.1.1 Start der Dienste der Fahrzeugbetriebsfunktionen

Beim Start der Geräte eines IBIS-IP-Systems startet auf jedem der Geräte zuerst der Dienst `DeviceManagementService` (vgl. 301-2-2).

Für jedes dieser Geräte ist festgelegt, ob weitere Dienste der Fahrzeugbetriebsfunktionen (d. h. `SystemDocumentationService` und/oder `SystemManagementService`) auf dem Gerät automatisch gestartet werden sollen. Die Festlegung, dass und ggf. welche Dienste automatisch gestartet werden, erfolgt durch eine geräte- bzw. herstellerspezifische Konfiguration, die nicht im Rahmen von IBIS-IP standardisiert, jedoch über eine Schnittstelle des `DeviceManagementServices` abrufbar und änderbar ist.

Die Festlegung muss so erfolgen, dass nach Start aller Geräte und Durchführung des Starts aller automatischen zu startenden Dienste der Fahrzeugbetriebsfunktionen folgende Situation eintritt:

— Auf jedem Gerät des IBIS-IP-Systems läuft genau ein `DeviceManagementService`

— Auf genau einem der Geräte des IBIS-IP-Systems läuft genau ein `SystemDocumentationService`. Es gibt keinen weiteren `SystemDocumentationService` im IBIS-IP-System (vgl. auch VDV 301-2-4).

— Auf genau einem der Geräte des IBIS-IP-Systems läuft genau ein `SystemManagementService`. Es gibt keinen weiteren `SystemManagementService` im IBIS-IP-System. (vgl. auch Kapitel 7.3).

Beispiel:
In einem System aus einem Bordrechner und mehreren Anzeigern starten durch den Start der Geräte zunächst alle `DeviceManagemetnServices`*. Auf dem Bordrechner werden durch den* `DeviceManagementService` *zusätzlich die Dienste* `SystemDocumentationServcie` *und* `SystemManagementService` *gestartet. Auf allen Anzeigern bleiben dagegen die* `DeviceManagementServices` *die einzigen vorerst laufenden Dienste.*

(Anmerkung: In Abbildung 3, Kapitel 5.3 wird der etwas komplexere Fall beschrieben, bei dem der `SystemDocumentationService` und der `SystemManagementService` auf verschiedenen Geräten betrieben werden.)

5.2.1.2 Veröffentlichung der Dienste der Fahrzeugbetriebsfunktionen

Alle Dienste in IBIS-IP nutzen den in Kapitel 3 beschriebenen Mechanismus der Veröffentlichung per DNS-SD.

Durch die Veröffentlichung werden folgende Informationen zu diesem Dienst systemweit bekannt gemacht:

— Dienstname

— IBIS-IP-Version des Dienstes

— Protokoll, über das der Dienst angesprochen werden kann (bei Diensten der Fahrzeugbetriebsfunktionen immer `_ibisip_http`

— IP-Adresse bzw. DNS-Name des Gerätes, auf dem der Dienst läuft

— Port, unter dem der Dienst zu erreichen ist

— Ggf. Pfad, unter dem der Dienst zu erreichen ist

5.2.1.3 Verbindungsaufbau mit den veröffentlichten Diensten der Fahrzeugbetriebsfunktionen und Ausführung fachlicher Aufgaben

Nach der Veröffentlichung der Dienste können die Dienste insbesondere des *SystemManagementService* angesprochen werden. Dabei werden die nachfolgend beschriebenen drei Teilschritte durchlaufen:

5.2.1.3.1 Abfrage der Systemkonfiguration

Der *SystemManagementService* erfragt zunächst beim *SystemDocumentationService* die Konfiguration des IBIS-IP-Systems und erhält eine fachliche Beschreibung des Systems, in der festgelegt ist, auf welchem Gerät (identifiziert durch Gerätetyp und Geräte-ID) welcher Dienst (identifiziert durch Dienstname und IBIS-IP-Version) laufen soll (vgl. auch Kapitel 5.1.1).

5.2.1.3.2 Abfrage der DeviceManagementServices

Der SystemManagementService verbindet sich anschließend mit allen verfügbaren *DeviceManagementServices*. Zu diesem Zeitpunkt ist dem *SystemManagementService* noch nicht bekannt, mit welchem *DeviceManagementService* auf welchem Gerät es kommuniziert. Deren technische Verbindungsdaten sind ihm aber durch die Veröffentlichung via DNS-SD bekannt.

Der *SystemManagementService* erfragt bei den *DeviceManagementServices* der Geräte insbesondere

— von welchem Gerätetyp das zugehörige Gerät ist und

— welche Geräte-ID es hat und

— welche Dienste auf dem Gerät verfügbar sind.

5.2.1.3.3 Zuordnung technischer und fachlicher Verbindungsdaten

Durch Kombination der technischen Verbindungsdaten aus der Veröffentlichung mit den fachlichen Informationen über Gerätetyp und Geräte-ID ist es dem *SystemManagementService* möglich, beide Daten gegenseitig korrekt zuzuordnen.

5.2.1.3.4 Startbefehle an die DeviceManagementServices

Mit dieser Zuordnung kann der *SystemManagementService* entsprechend den fachlichen Konfigurationsdaten vom *SystemDocumentationService* sich auch mit den *DeviceManagementServices* der einzelnen Geräte technisch korrekt verbinden, um auf dem betreffenden Gerät weitere Dienste zu starten.

Sollte der *SystemManagementService* feststellen, dass Dienste auf Geräten nicht verfügbar sind, auf denen sie aber laut Konfiguration laufen sollen, so erkennt dies der *SystemManagementService* und speichert den Fehlerzustand.

Sollte der *SystemManagementService* feststellen, dass Geräte nicht verfügbar sind, die laut Konfiguration vorhanden sein sollen, so erkennt dies der *SystemManagementService* und speichert den Fehlerzustand.

5.2.2 Stufe 2: Start der fachlichen Dienste

5.2.2.1 Start weiterer Dienste auf den Geräten

Nachdem der *SystemManagementService* sowohl die Systemkonfiguration kennt als auch alle *DeviceManagementServices* der beteiligten Geräte zuordnen und ansprechen kann, ist der *SystemManagementService* in der Lage, weitere Dienste des IBIS-IP-Systems auf den Geräten zu starten.

Dazu ruft der *SystemManagementService* Operation **StartService** bei den *DeviceManagementServices* der beteiligten Geräte auf. Auf diese Weise werden nacheinander alle weiteren Dienste gestartet, die in der Konfiguration angegeben sind.

5.2.2.2 Veröffentlichung weiterer Dienste

Alle Dienste in IBIS-IP nutzen den in Kapitel 3 beschriebenen Mechanismus der Veröffentlichung per DNS-SD.

Durch die Veröffentlichung werden folgende Informationen zu diesen Diensten systemweit bekannt gemacht:

Bei HTTP-Diensten

— Dienstname,

— IBIS-IP-Version des Dienstes,

— Protokoll, über das der Dienst angesprochen werden kann,

— IP-Adresse bzw. DNS-Name des Gerätes, auf dem der Dienst läuft,

— Port, unter dem der Dienst zu erreichen ist,

— Ggf. Pfad, unter dem der Dienst zu erreichen ist.

Bei UDP-Diensten

— Dienstname,

— IBIS-IP-Version des Dienstes,

— Protokoll, über das der Dienst angesprochen werden kann,

— IP-Adresse der Multicast-Gruppe, über die die Informationen des Dienstes verbreitet werden.

5.2.2.3 Verbindungsaufbau mit den veröffentlichten Diensten und Ausführung fachlicher Aufgaben

Nach der Veröffentlichung aller benötigten bzw. konfigurierten Dienste werden die informationskonsumierenden Dienste und Applikationen entweder die entsprechenden Abfragen oder die gewünschten Daten-Abonnements einrichten.

Zur Sicherstellung der Funktionsfähigkeit von Datenabonnements (da es Datensendungen nur bei Änderung der Informationen gibt, wissen die Datenkonsumenten in aller Regel nicht, ob ein Dienst noch tätig ist) wird empfohlen, dass die Dienste und Applikationen, die ein Datenabonnement bei einem Dienst einrichten, sich zusätzlich über den aktuellen Systemzustand informieren lassen.

Beispiel:
Die Applikation auf einem Fahrgast-Anzeiger ist so implementiert, dass sie immer nach einem Dienst `CustomerInformationService` *in der IBIS-Version 2.0 sucht und von diesem Dienst Daten abruft und anzeigt, sobald der Dienst für die Applikation auf dem Fahrgast-Anzeiger verfügbar ist. Es ist insbesondere nicht erforderlich, durch eine manuelle Konfiguration diese Festlegung zu treffen*

5.3 Beispiel eines Systemstarts in IBIS-IP

Nachfolgend ist der Systemstart eines IBIS-IP-Systems für eine Beispielkonfiguration beschrieben, an der zwei Geräte, Gerät A und Gerät B, beteiligt sind.

Auf Gerät A laufen folgende Dienste

— `DeviceManagementService`

— `SystemManagementService`

— `Dienst X`

Auf Gerät B laufen folgende Dienste:

— `DeviceManagementService`

— `SystemDocumentationService`

— *`Dienst Y`*, bei dem man ein Abo einrichten kann.

Der *`SystemManagementService`* fragt außerdem regelmäßig per Statusabfrage bei beiden *`DeviceManagementServices`* den Status der Dienste auf den Geräten sowie den Status des Gerätes ab (vgl. auch 7.3 und VDV 301-2-2).

Auch systemrelevante Dienste der Fahrzeugbetriebsfunktionen werden im Beispiel vom *`DeviceManagementService`* des Gerätes gestartet. Dies kann aber auch auf anderem Weg (z. B. durch ein Skript, eine Applikation oder eine sonstige Autostart-Funktion) geschehen.

5.3.1 Stufe 1: Start systemrelevanter Fahrzeugbetriebsfunktionen

5.3.1.1 Start der Dienste der Fahrzeugbetriebsfunktionen

Die *`DeviceManagementServices`* beider Geräte starten automatisch durch den Gerätestart.

Auf Gerät A sorgt dann die gerätespezifische Konfiguration dafür, dass durch den *`DeviceManagementService`* des Gerätes A auch der Dienst *`SystemManagementService`* gestartet wird.

Auf Gerät B sorgt dann die gerätespezifische Konfiguration dafür, dass durch den *`DeviceManagementService`* des Gerätes B auch der Dienst *`SystemDocumentationService`* gestartet wird.

5.3.1.2 Veröffentlichung der Dienste der Fahrzeugbetriebsfunktionen

Diese Stufe wird zeitlich versetzt durchlaufen. Jeder Dienst veröffentlicht sich, sobald er betriebsbereit ist. Die *DeviceManagementServices* veröffentlichen sich zeitlich also vor den anderen Diensten der Fahrzeugbetriebsfunktionen.

5.3.1.3 Verbindungsaufbau mit veröffentlichten Diensten der Fahrzeugbetriebsfunktionen und Ausführung fachlicher Aufgaben

Der *SystemManagementService* wird per DNS-SD informiert, wenn ein neuer Dienst zur Dienstliste von DNS-SD hinzukommt. Sobald ein *SystemDocumentationService* in dieser Liste auftaucht, verbindet sich der *SystemManagementService* mit diesem Dienst und fragt dort die System-Konfiguration (in der Abbildung 3, SDS/GetSystemConfiguration) ab.

Sobald ein *DeviceManagementService* in der Dienstliste des DNS-SD auftaucht, verbindet sich der *SystemManagementService* mit diesem Dienst und ruft dort die Device-Information (für die Geräteklasse), Device-Configuration (für die Geräte-ID) und Service-Informationen (für die Liste der verfügbaren Dienste) ab. Auf diese Weise kann der SystemManagementService

— eine Zuordnung von fachlicher Identifikation (Geräteklasse und Geräte-ID) zu technischer Identifikation (Dienstadressierung nach DNS-SD) machen und

— prüfen, ob alle erforderlichen Dienste auf den in der System-Konfiguration vorgesehenen Geräten vorhanden sind.

5.3.2 Stufe 2: Start der fachlichen Dienste

5.3.2.1 Start weiterer Dienste

Der *Dienst X* auf Gerät A wird durch den *SystemManagementService* gestartet, indem die Dienst-Start-Operation des *DeviceManagementService* des Geräts A aufgerufen und der Name des Dienstes übergeben wird (vgl. Kapitel 7.1.23). Analog erfolgt der Start des *Dienstes Y* auf Gerät B über den *DeviceManagementService* von Gerät B.

5.3.2.2 Veröffentlichung weiterer Dienste

Diese Stufe wird zeitlich versetzt durchlaufen. Jeder Dienst veröffentlicht sich, sobald er betriebsbereit ist. Die Dienste *Dienst X* und *Dienst Y* veröffentlichen sich jeweils, sobald sie betriebsbereit sind.

5.3.2.3 Verbindungsaufbau mit den veröffentlichten Diensten und Ausführung fachlicher Aufgaben

Der *SystemManagementService* fragt zyklisch den Status aller Dienste und Geräte bei den *DeviceManagementServices* der Geräte ab.

Im Beispiel richtet außerdem der *Dienst Y* beim *SystemManagementService* ein Abonnement ein, um kontinuierlich über Änderungen des Systemzustands informiert zu sein.

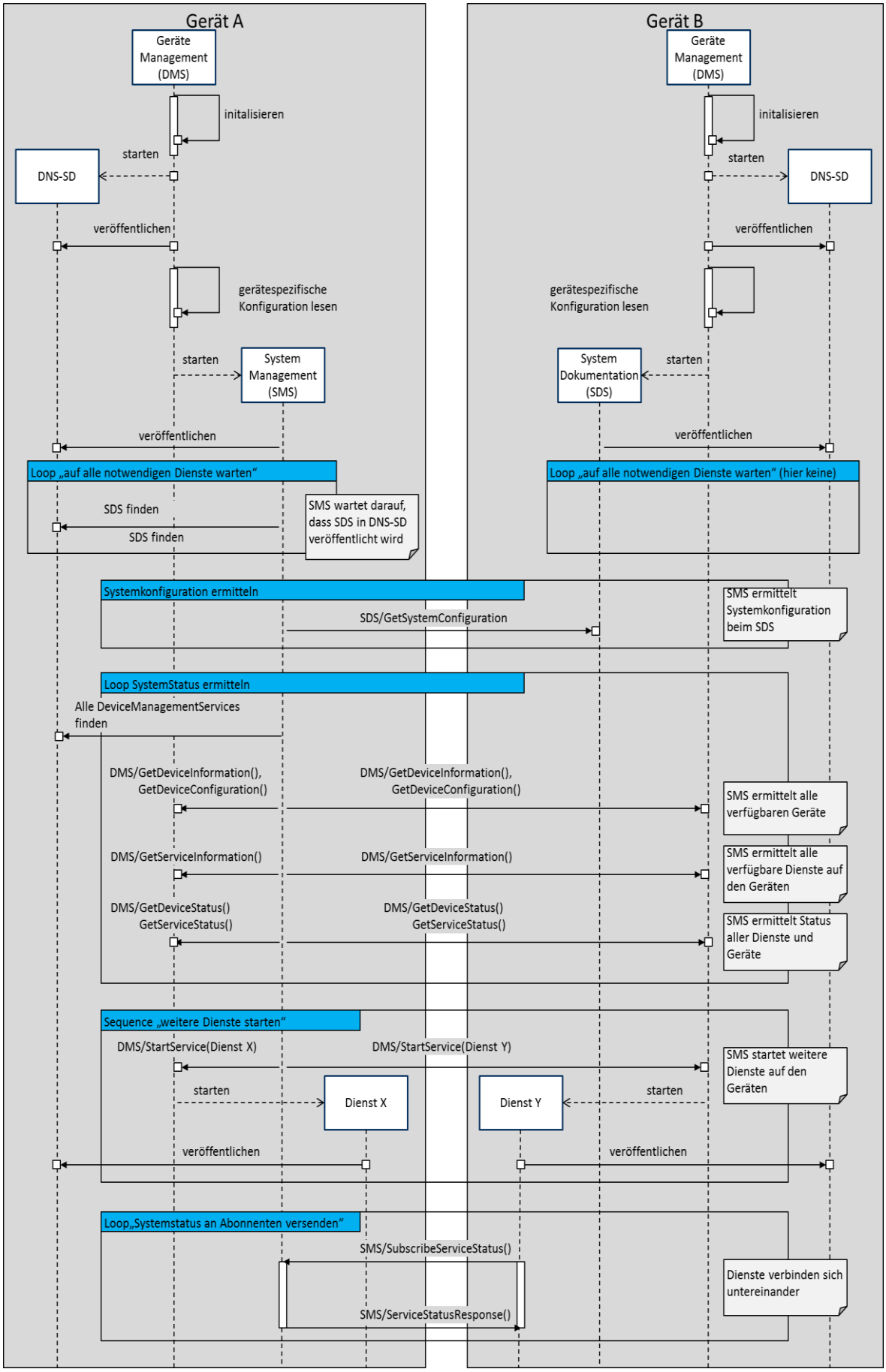

Abbildung 3 Beispielhafter Startup eines IBIS-IP-Systems

6 Strukturierung der Informationsinhalte

In IBIS-IP werden Dateninhalte in Klartext übertragen. Das bedeutet insbesondere, dass mehrere Informationen nicht in einem Datenfeld kodiert übertragen werden dürfen, sondern entsprechend in mehrere Datenfelder einer Datenstruktur aufzubrechen sind.

Es wird weiterhin Wert auf die Verwendung etablierter und weit verbreiteter Verfahren zur Übertragung strukturierter Daten gelegt.

Informationsinhalte werden deshalb in IBIS-IP mit Hilfe von XML-Datenstrukturen übertragen und können mit Hilfe eines XML-Schemas (XSD = XML Schema Definition) entsprechend validiert werden. Neben der hier dargestellten Form stellt der VDV über seine Webseite die XSD-Dateien der spezifizierten Dienste kostenlos zur Verfügung.

6.1 Notation der XML-Elemente und -Strukturen

Die in diesem Dokument vorgestellten IBIS-IP-Schnittstellen werden mit Hilfe von XML-Schema definiert. Die Objekte, die über die Schnittstelle ausgetauscht werden, liegen folglich als XML-Elemente vor. Die Beschreibung der XML-Elemente wird in diesem Dokument in einer Tabellenform vorgenommen, die aus SIRI (CEN, TS 15531 Part 1) stammt. Sie ist sehr kompakt, übersichtlich und bietet eine Vielzahl an strukturellen Informationen, die ansonsten nur in der XML-Schema-Definition sichtbar wird. Dieses Kapitel erläutert die Notation der Tabellenform, die ab Kapitel 7 genutzt wird.

Alle Namen von Elementen, Datentypen und Attributen sind in Englisch gehalten, um eine etwaige Normierung auf europäischer Ebene vorzubereiten und den Austausch mit europäischen Partnern zu erleichtern.

6.1.1 Darstellung von XML-Elementen im Text

Eine konsistente Notation der XML-Elemente soll helfen, technisch wichtige Information beim Lesen bereit zu stellen.

— XML-Elemente werden in Groß-Klein-Schreibweise (Upper Camel Case) fett und kursiv geschrieben, z. B.: ***VehicleJourneyRef***. Die Elementnamen sind – wo immer möglich und sinnvoll – an Begriffe aus TransModel angelehnt. Fehlt in TransModel ein geeigneter Begriff für ein Konzept oder Objekt, so wurde versucht, den entsprechenden Begriff aus JourneyWeb oder das passende Konzept aus DELFI zu übernehmen.

— Datentypen werden kursiv dargestellt, z. B: *xsd:boolean.*

— Code-Beispiele werden in kleinerer Schrift wiedergegeben.

6.1.2 Tabellennotation für Operationen

Darüber hinaus werden Operationen eines Dienstes in einer Tabelle der folgenden Form beschrieben.

Operation	**Request/ Response**	**Verwendeter Datentyp, Datenstruktur**
GetData	Req.	-
	Resp.	BeaconLocationService. GetDataResponseStructure
SubscribeData	Req.	SubscribeRequestStructure
	Resp.	SubscribeResponseStructure
UnsubscribeData	Req.	TerminateSubscribeRequestStructure
	Resp.	TerminateSubscribeResponseStructure

Tabelle 4 Beispiel für Operationsnotation in IBIS-IP

6.1.3 Tabellennotation von XML-Strukturen

In dieser Schrift werden XML-Strukturen in einer Tabellennotation dargestellt (vgl. Tabelle 5). Für jedes wichtige Dienst-Anfrage/Antwort-Element findet sich eine eigene Tabelle. Weitere Tabellen werden für alle wesentlichen Kind-Elemente, aus denen die komplexen Strukturen aufgebaut sind, angegeben. Um Platz zu sparen, werden die Spaltenüberschriften nur im Beispiel in Tabelle 5 angezeigt und bei allen folgenden Tabellen nicht wiederholt. In den Tabellen wird ein konsistenter Satz an Regeln zur Beschreibung der XML-Elemente und der daran geknüpften Bedingungen verwendet.

Gruppierung	**Elementname**	**Min : Max**	**Datentyp**	**Erläuterung**
ContinuousServiceStructure			*+Structure*	Eine Fahrgastbewegung mit Hilfe eines kontinuierlichen, nicht fahrplangebundenen Verkehrsmittels.
	a ***ContinuousMode***	**-1:1**	*walk | parkAndRide | demandResponsive*	Modalität für kontinuierliche Verkehre
	b ***IndividualMode***		*walk | cycle | taxi | self-drive-car | others-drive-car | motorcycle | truck*	Verkehrsmittelmodalität für Individualverkehr
DatedService	***OperatingDay***	**1:1**	*→OperatingDay*	Betriebstag der Fahrt.
	VehicleRef	0:1	*→Vehicle*	Fahrzeug-ID.
ServiceJourney	***JourneyRef***	**1:1**	*→Journey*	Fahrt-ID. .
LineIdentity	***LineRef***	**1:1**	*→Line*	Linien-ID. .
	DirectionRef	**1:1**	*→Direction*	Richtungs-ID..
Service	***Mode***	**1:1**	*+Mode*	Verkehrsmitteltyp. .
	PublishedLineName	**1:1**	*InternationalText*	Liniennummer oder -name, wie in der Öffentlichkeit bekannt.
	OperatorRef	0:1	*→Operator*	Operator-ID. .

	RouteDescription	0:1	*InternationalText*	Beschreibung des Fahrwegs.
	Via	0:*	*+ServiceViaPoint*	Wichtige Halte auf dem Fahrweg.
	Attribute	0:*	*+GeneralAttribute*	Hinweise und Attribute (mit Klassifikationen) zur Fahrt.
ServiceOrigin	*OriginStopPointRef*	0:1	*→StopPoint*	ID des ersten Haltepunkts der Fahrt; Starthaltestelle.
	OriginText	0:1	*InternationalText*	Name des ersten Haltepunkts der Fahrt, der Starthaltestelle.

Tabelle 5 Beispiel (für Kapitel 7 für die tabellarische Notation einer XML-Struktur)

6.1.3.1 Gruppierung

In der ersten Spalte befindet sich gelegentlich ein Bezeichner, der die Elemente in sinnvolle Gruppierungen einteilt, z. B. *Service* oder *ServiceOrigin*. Dies dient rein zu Dokumentationszwecken und entspricht in den meisten Fällen den Namen einer XML-Gruppe, die im XML-Schema verwendet wurde. Die Verwendung von Gruppierungen hat nur den Zweck, die Elemente zu organisieren und damit für mehr Klarheit und bessere Wiederverwendbarkeit zu sorgen.

6.1.3.2 Elementname

Elementnamen werden kursiv in der zweiten Spalte wiedergegeben, z. B. *OperatingDay*. Handelt es sich um ein verpflichtendes Element, so wird es **fett** gedruckt. Optionale Elemente werden nicht fett gedruckt. Der Name der Struktur selbst ist links oben in der Tabelle angegeben.

Elemente, die geerbt (XML: “derived by extension”) oder anonym verwendet werden, tragen im Namensfeld drei Doppelpunkte “:::” zur Kennzeichnung.

6.1.3.3 Multiplizität & Choice (Min:Max)

Die Bedingungen, ob ein Element verpflichtend oder optional ist oder ob es einfach oder mehrfach innerhalb des übergeordneten Elements auftreten kann, werden in der dritten Spalte Min:Max angegeben. Dabei werden die üblichen UML-Konventionen „min:max“ angewendet, so steht z. B. „0:1“ für ein optionales, einfaches Element, „**1:1**“ zeigt ein verpflichtendes, einfaches Element an, „0:*“ steht für ein optionales, mehrfaches Element usw. Verpflichtende Elemente werden **fett** gedruckt.

In manchen Fällen muss ein Element aus seiner Menge ausgewählt werden (XML-Choice). Dies wird durch ein vorangestelltes Minuszeichen symbolisiert, z. B. „**-1:1**“. In diesem Fall steht vor dem Elementnamen noch ein Kleinbuchstabe, der die Auflistung der Wahlmöglichkeiten anzeigt. Bei optionalen Auswahlmöglichkeiten (Choices) steht im Min-Wert eine Null: „-0:1“.

6.1.3.4 Datentyp

Die Datentypen werden in der vierten Spalte kursiv angegeben, z. B. *InternationalText*. Falls der Namensraum (namespace) vom Standard XML Namensraum abweicht, wird er mit angegeben, z. B. „*xs:dateTime*“ oder „*siri:PtSituationElement*“.

— Ein komplexer Datentyp, der selbst Strukturen als Kind-Elemente enthält, wird in der Spalte Datentyp mit „*+Structure*“ gekennzeichnet.

— Wo Elemente als Referenzen (Fremdschlüssel) auf andere Objekte verwendet werden, wird als Datentyp der Typ des referenzierten Objekts mit vorangestelltem Pfeil verwendet. Zum Beispiel „→*StopPoint*“ als Typ einer Referenz (*StopPointRefStructure*) auf ein Objekt vom Typ „StopPointType“.

— Aufzählungstypen (Enumerated types) werden an den meisten Stellen unmittelbar mit den verwendbaren Werten dargestellt, z. B. „*walk | cycle*“. Nur in einigen Fällen mit sehr umfangreichen Aufzählungen, die an mehreren Stellen wiederverwendet werden, wird ein Typ deklariert und referenziert.

— Um Platz zu sparen, werden bei der Angabe der Datentypen Abkürzungen verwendet. Z. B. wird auf die Endungen “Structure” und “Type” durchgehend verzichtet. Statt beispielsweise „*InternationalTextStructure*“ wird also immer „*InternationalText*“ als Datentyp angegeben.

6.1.3.5 Erläuterung

Alle Elemente erhalten in der letzten Spalte eine Erläuterung ihres Verwendungszwecks. An vielen Stellen wird auf weitere Stellen im Text hingewiesen, so z. B. bei komplexen Kind-Elementen an die Stelle, wo ihre Tabellenbeschreibung zu finden ist. An einigen Stellen ist die Erläuterung zu umfangreich und würde die Tabellenform sprengen. Dann finden sich diese Anmerkungen im Text unterhalb der Tabelle.

Ergänzend zu diesem Dokument existiert eine XSD-Datei mit den Inhalten in digitaler Form. Diese steht, falls nicht mit diesem Dokument mitgeliefert, zum Download unter www.vdv.de/ip-kom-oev.aspx zur Verfügung.

7 Beschreibung einzelner Fachdienste

7.1 Dienst DeviceManagementService

Der Dienst *DeviceManagementService* ist der Dienst, welcher die Aufgaben der Fachkomponente Geräte-Management innerhalb von IBIS-IP umsetzt.

Der Dienst ist für alle Daten als http-Dienst spezifiziert.

Dieser Dienst ist auf jedem an IBIS-IP teilnehmenden Gerät einmal vorhanden und wird mit dem Ausführen der Applikation auf dem Gerät automatisch gestartet.

7.1.1 Operationen des DeviceManagementService

Die vom *DeviceManagementService* angebotenen Operationen lassen sich unterteilen in Operationen die das Gerät betreffen

- Statische Informationen über das Gerät abfragen
- Konfiguration des Geräts ändern
- Konfiguration des Geräts abfragen
- Gerät restarten, deaktivieren, aktivieren
- Status des Geräts abfragen
- Fehlermeldungen abfragen

sowie Operationen, die die Dienste auf dem Gerät betreffen

- Statische Informationen über die Dienste
- Dienste starten/stoppen/restarten
- Zustand der Dienste abfragen

Operation	Request/ Response	Verwendeter Datentyp, Datenstruktur
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
UnsubscribeDeviceErrorMessages	Req.	UnsubscribeRequestStructure
	Resp.	UnsubscribeResponseStructure

Operation	**Request/ Response**	**Verwendeter Datentyp, Datenstruktur**
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

Tabelle 6 Beschreibung von Operationen des DeviceManagementService

7.1.2 Datenstrukturen der Operation GetDeviceInformation

7.1.2.1 Request

Beim Request der Operation ***GetDeviceInformation*** werden keine Daten übergeben.

7.1.2.2 Response

DeviceManagementService.GetDeviceInformationResponse				*+Structure*	**Struktur für die Beantwortung einer Anfrage nach nicht konfigurierbaren Informationen über das Gerät**
				choice	eine der beiden folgenden Möglichkeiten
	a	***DeviceManagementService.GetDeviceInformationResponseData***	**–1:1**	*+DeviceManagementService.GetDeviceInformationResponseData*	Ausführliche Antwortstruktur für Angaben zu nicht konfigurierbaren Geräteeigenschaften (siehe unten)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Rückmeldung von Fehlern

Tabelle 7 Beschreibung DeviceManagementService.GetDeviceInformationResponse

DeviceManagementService.GetDeviceInformationResponseData			*+Structure*	**Ausführliche Antwortstruktur für Angaben zu nicht konfigurierbaren Geräteeigenschaften**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Angabe des Zeitpunkts der Antwort
	DeviceInformation	**1:1**	*+DeviceInformationStructure*	Ausführliche Antwortstruktur (vgl. 301-2-1)

Tabelle 8 Beschreibung DeviceManagementService.GetDeviceInformationResponseData

Für die Einrichtung von Abonnements werden die der VDV 301-2-1 beschriebenen Datenstrukturen verwendet.

7.1.3 Datenstrukturen der OperationUnsubscribeDeviceInformation

Für das Beenden von Abonnements werden die VDV 301-2-1 beschriebenen Datenstrukturen verwendet.

7.1.4 Datenstrukturen der Operation GetDeviceConfiguration

Die Operation ***GetDeviceConfiguration*** dient dazu, den einzigen eventuell variablen Parameter eines Gerätes, nämlich die Geräte-ID, abzufragen.

7.1.4.1 Request

Beim Request der Operation ***GetDeviceConfiguration*** werden keine Daten übergeben.

7.1.4.2 Response

DeviceManagementService.GetDeviceConfigurationResponse				*+Structure*	**Struktur für die Beantwortung einer Anfrage nach Konfiguration eines Geräts**
				choice	eine der beiden folgenden Möglichkeiten
	a	***DeviceManagementService.GetDeviceConfigurationResponseData***	**–1:1**	*+DeviceManagementService.GetDeviceConfigurationResponseData*	Ausführliche Antwortstruktur zur Konfiguration eines Gerätes (siehe unten)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Rückmeldung von Fehlern

Tabelle 9 Beschreibung DeviceManagementService.GetDeviceConfigurationResponse

DeviceManagementService.GetDeviceConfigurationResponseData			*+Structure*	**Ausführliche Antwortstruktur zur Konfiguration eines Gerätes**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Angabe des Zeitpunkts der Antwort
	DeviceID	**1:1**	*IBIS-IP.int*	Geräte-Identifaktionsnummer bzw. Geräte-Einbauort

Tabelle 10 Beschreibung DeviceManagementService.GetDeviceConfigurationResponseData

7.1.5 Datenstrukturen der Operation SetDeviceConfiguration

Über diese Operation kann die Geräte-ID (Device-ID) eines Gerätes eingestellt werden. Die Einstellung erfolgt einmalig nach Geräte-Einbau und wird dann in einer gerätespezifischen Konfigurationsdatei abgelegt. Die Operation dient somit als Rückfallebene zur Einstellung der Geräte-ID, falls die Erkennung der Geräte-ID anhand der Einbauposition nicht automatisch erfolgt.

7.1.5.1 Request

DeviceManagementService.SetDeviceConfigurationRequest			*+Structure*	**Struktur für Einstellung der Geräte-ID in der gerätespezifischen Konfiguration**
	DeviceID	**1:1**	*IBIS-IP.int*	Angabe des einzustellenden Geräte-ID

Tabelle 11 Beschreibung DeviceManagementService.SetDeviceConfigurationRequest

7.1.5.2 Response

Der Empfang des Requests wird mit Hilfe der DataAcceptedResponseStructure (vgl. VDV 301-2-1) bestätigt.

7.1.6 Datenstrukturen der Operation SubscribeDeviceConfiguration

Für die Einrichtung von Abonnements werden die in der VDV 301-2-1 beschriebenen Datenstrukturen verwendet.

7.1.7 Datenstrukturen der Operation UnsubscribeDeviceConfiguration

Für das Beenden von Abonnements werden die in der VDV 301-2-1 beschriebenen Datenstrukturen verwendet.

7.1.8 Datenstrukturen der Operation GetDeviceStatus

7.1.8.1 Request

Beim Request der Operation ***GetDeviceStatus*** werden keine Daten übergeben.

7.1.8.2 Response

DeviceManagementService.GetDeviceStatusResponse				*+Structure*	**Struktur für die Beantwortung einer Anfrage nach dem Status eines Geräts**
				choice	eine der beiden folgenden Möglichkeiten
	a	***DeviceManagementService.GetDeviceStatusResponseData***	**–1:1**	*+DeviceManagementService.GetDeviceStatusResponseData*	Ausführliche Antwortstruktur für Angaben zum Status eines Geräts (siehe unten)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Rückmeldung von Fehlern

Tabelle 12 Beschreibung DeviceManagementService.GetDeviceStatusResponse

DeviceManagementService.GetDeviceStatusResponseData			*+Structure*	**Ausführliche Antwortstruktur für Angaben zum Status eines Geräts**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Angabe des Zeitpunkts der Antwort
	DeviceState	**1:1**	*+DeviceStateEnumeration*	Angabe zum Gerätestatus (vgl. 9.3)

Tabelle 13 Beschreibung DeviceManagementService.GetDeviceStatusResponseData

7.1.9 Datenstrukturen der Operation SubscribeDeviceStatus

Für die Einrichtung von Abonnements werden die in der VDV 301-2-1 beschriebenen Datenstrukturen verwendet.

7.1.10 Datenstrukturen der Operation UnsubscribeDeviceStatus

Für das Beenden von Abonnements werden die in der VDV 301-2-1 beschriebenen Datenstrukturen verwendet.

7.1.11 Datenstrukturen der Operation GetDeviceErrorMessages

7.1.11.1 Request

Beim Request der Operation ***GetDeviceErrorMessage*** werden keine Daten übergeben.

7.1.11.2 Response

DeviceManagementService.GetDeviceErrorMessagesResponse				*+Structure*	**Struktur für die Beantwortung einer Anfrage nach Fehlermeldungen eines Geräts**
				choice	eine der beiden folgenden Möglichkeiten
	a	***DeviceManagementService.GetDeviceErrorMessagesResponseData***	**–1:1**	*+DeviceManagementService.GetDeviceErrorMessagesResponseData*	Ausführliche Antwortstruktur für Fehlermeldungen eines Geräts (siehe unten)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Rückmeldung von Fehlern

Tabelle 14 Beschreibung DeviceManagementService.GetDeviceErrorMessagesResponse

DeviceManagementService.GetDeviceErrorMessagesResponseData			*+Structure*	**Ausführliche Antwortstruktur für Fehlermeldungen eines Geräts**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Angabe des Zeitpunkts der Antwort
	ErrorMessage	**10:***	*+Message*	Nur Fehlermeldungen und Warnungen (keine Statusmeldungen) (vgl. 9.5)

Tabelle 15 Beschreibung DeviceManagementService.GetDeviceErrorMessagesResponseData

7.1.12 Datenstrukturen der Operation SubscribeDeviceErrorMessages

Für die Einrichtung von Abonnements werden die in der VDV 301-2-1 beschriebenen Datenstrukturen verwendet.

7.1.13 Datenstrukturen der Operation UnsubscribeDeviceErrorMessages

Für das Beenden von Abonnements werden in der VDV 301-2-1 beschriebenen Datenstrukturen verwendet.

7.1.14 Datenstrukturen der Operation RestartDevice

7.1.14.1 Request

Beim Request der Operation **`RestartDevice`** werden keine Daten übergeben.

7.1.14.2 Response

Der Empfang des Requests wird mit Hilfe der DataAcceptedResponseStructure (vgl. VDV 301-2-1) bestätigt.

7.1.15 Datenstrukturen der Operation DeactivateDevice

7.1.15.1 Request

Beim Request der Operation ***`DeactivateDevice`*** werden keine Daten übergeben.

7.1.15.2 Response

Der Empfang des Requests wird mit Hilfe der DataAcceptedResponseStructure (vgl. VDV 301-2-1) bestätigt.

7.1.16 Datenstrukturen der Operation ActivateDevice

7.1.16.1 Request

Beim Request der Operation ***ActivateDevice*** werden keine Daten übergeben.

7.1.16.2 Response

Der Empfang des Requests wird mit Hilfe der DataAcceptedResponseStructure (vgl. VDV 301-2-1) bestätigt.

7.1.17 Datenstrukturen der Operation GetServiceInformation

7.1.17.1 Request

Beim Request der Operation ***GetServiceInformation*** werden keine Daten übergeben.

7.1.17.2 Response

DeviceManagementService.GetServiceInformationResponse				*+Structure*	**Struktur für die Beantwortung einer Anfrage nach den auf einem Gerät verfügbaren Diensten**
				choice	eine der beiden folgenden Möglichkeiten
	a	***DeviceManagementService.GetServiceInformationResponseData***	**–1:1**	*+DeviceManagementService.GetServiceInformationResponseData*	Ausführliche Antwortstruktur für Angaben zu den Diensten, die auf einem Gerät verfügbar sind (siehe unten)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Rückmeldung von Fehlern

Tabelle 16 Beschreibung DeviceManagementService.GetServiceInformationResponse

DeviceManagementService.GetServiceInformationResponseData			*+Structure*	**Ausführliche Antwortstruktur für Angaben zu den Diensten, die auf einem Gerät verfügbar sind**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Angabe des Zeitpunkts der Antwort
	ServiceInformationList	**1:1**	*+ServiceInformationList*	Liste der auf dem Gerät verfügbaren Dienste (vgl. VDV 301-2-1)

Tabelle 17 Beschreibung DeviceManagementService.GetServiceInformationResponseData

7.1.18 Datenstrukturen der Operation SubscribeServiceInformation

Für die Einrichtung von Abonnements werden die in der VDV 301-2-1 beschriebenen Datenstrukturen verwendet.

7.1.19 Datenstrukturen der Operation UnsubscribeServiceInformation

Für das Beenden von Abonnements werden die in der VDV 301-2-1 beschriebenen Datenstrukturen verwendet.

7.1.20 Datenstrukturen der Operation GetServiceStatus

7.1.20.1 Request

Beim Request der Operation ***GetServiceStatus*** werden keine Daten übergeben.

7.1.20.2 Response

DeviceManagementService.GetServiceStatusResponse				*+Structure*	**Struktur für die Beantwortung einer Anfrage nach dem Status der Dienste auf einem Gerät**
				choice	eine der beiden folgenden Möglichkeiten
	a	***DeviceManagementService.GetServiceStatusResponseData***	**–1:1**	*+DeviceManagementService.GetServiceStatusResponseData*	Ausführliche Antwortstruktur für Statusinformationen zu Diensten auf einem Gerät (siehe unten)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Rückmeldung von Fehlern

Tabelle 18 Beschreibung DeviceManagementService.GetServiceStatusResponse

DeviceManagementService.GetServiceStatusResponseData			*+Structure*	**Ausführliche Antwortstruktur für Statusinformationen zu Diensten auf einem Gerät**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Angabe des Zeitpunkts der Antwort
	ServiceSpecificationWithStateList	**1:1**	*+ServiceSpecificationWithStateList*	Liste der Dienste auf dem Gerät mit ihrem jeweiligem Status (vgl. VDV 301-2-1)

Tabelle 19 Beschreibung DeviceManagementService.GetServiceStatusResponseData

7.1.21 Datenstrukturen der Operation SubscribeServiceStatus

Für die Einrichtung von Abonnements werden die in der VDV 301-2-1 beschriebenen Datenstrukturen verwendet.

7.1.22 Datenstrukturen der Operation UnsubscribeServiceStatus

Für das Beenden von Abonnements werden die in der VDV 301-2-1 beschriebenen Datenstrukturen verwendet.

7.1.23 Datenstrukturen der Operation StartService

7.1.23.1 Request

DeviceManagementService.StartServiceRequestStructure			*+Structure*	**Struktur, mit deren Hilfe ein bestimmter Dienst (identifiziert über die ServiceSpecification) auf einem Gerät gestartet wird.**
	ServiceSpecification	**1:1**	*+ServiceSpecification*	Verweis auf den zu startenden Dienst (vgl. VDV 301-2-1)

Tabelle 20 Beschreibung DeviceManagementService.StartServiceRequestStructure

7.1.23.2 Response

Der Empfang des Requests wird mit Hilfe der DataAcceptedResponseStructure (vgl. VDV 301-2-1) bestätigt.

7.1.24 Datenstrukturen der Operation StopService

7.1.24.1 Request

DeviceManagementService.StopServiceRequestStructure			*+Structure*	**Struktur, mit deren Hilfe ein bestimmter Dienst (identifiziert über die ServiceSpecification) auf einem Gerät gestoppt wird.**
	ServiceSpecification	**1:1**	*+ServiceSpecification*	Verweis auf den zu startenden Dienst (vgl. VDV 301-2-1)

Tabelle 21 Beschreibung DeviceManagementService.StopServiceRequestStructure

7.1.24.2 Response

Der Empfang des Requests wird mit Hilfe der DataAcceptedResponseStructure (vgl. VDV 301-2-1) bestätigt.

7.1.25 Datenstrukturen der Operation RestartService

7.1.25.1 Request

DeviceManagementService.RestartServiceRequestStructure			*+Structure*	**Struktur, mit deren Hilfe ein bestimmter laufender Dienst (identifiziert über die ServiceSpecification) auf einem Gerät neu gestartet wird.**
	ServiceSpecification	**1:1**	*+ServiceSpecification*	Verweis auf den zu startenden Dienst (vgl. VDV 301-2-1)

Tabelle 22 Beschreibung DeviceManagementService.RestartServiceRequestStructure

7.1.25.2 Response

Der Empfang des Requests wird mit Hilfe der DataAcceptedResponseStructure (vgl. VDV 301-2-1) bestätigt.

7.2 Dienst SystemDocumentationService

Der Dienst `SystemDocumenationService` ist in IBIS-IP die Umsetzung der Aufgaben der Fachkomponente System-Dokumentation. Seine Hauptaufgaben bestehen darin, Systemmeldungen zu dokumentieren und die Systemkonfiguration zur Verfügung zu stellen. Da das Hauptaugenmerk auf einer Sicherstellung der Informationsübertragung liegt, ist dieser Dienst in IBIS-IP als HTTP-Dienst umgesetzt worden.

Dieser Dienst wird auf dem Gerät, auf dem per Gerätekonfiguration festgelegt ist, dass er gestartet werden soll, mit dem Ausführen der Applikation auf dem Gerät automatisch gestartet.

7.2.1 Operationen des SystemDocumentationService

Operation	Request/ Response	Verwendeter Datentyp, Datenstruktur
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

Tabelle 23 Beschreibung von Operationen des SystemDocumentationService

7.2.2 Datenstruktur der Operation GetSystemConfiguration

7.2.2.1 Request

Da es sich um eine ***Get***-Operation handelt, gibt es bei dieser Operation keine Request-Struktur.

7.2.2.2 Response

SystemDocumentationService.GetSystemConfigurationResponse				*+Structure*	**Struktur für die Beantwortung eine Anfrage nach Informationen über die Systemkonfiguration**
				Choice	eine der beiden folgenden Strukturen
	a	***SystemConfiguration***	**–1:1**	*+SystemDocumentationService.SystemConfigurationData*	ausführliche Antwortstruktur (siehe unten)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Rückmeldung von Fehlern

Tabelle 24 Beschreibung von SystemDocumentationService.GetSystemConfigurationResponse

SystemDocumentationService.SystemConfigurationData			*+Structure*	**Struktur zur Beschreibung der Systemkonfiguration**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Angabe des Zeitpunkts der Antwort
	ServiceStartList	**1:1**	*+ServiceStartList*	Liste mit Informationen der zu startenden Dienste (siehe auch VDV 301-2-1)
	DeviceSpecificationList	**1:1**	*+DeviceSpecificationList*	Liste mit Informationen der vorhandenen Geräte (siehe auch VDV 301-2-1)
	HertbeatIntervall	0:1	*IBIS-IP.duration*	Wert zur Konfiguration des Intervalls für die Heartbeat-Abfrage des SystemManagementService

Tabelle 25 Beschreibung von SystemDocumentationService.SystemConfigurationData

7.2.3 Datenstruktur der Operation SubscribeSystemConfiguration

Für die Einrichtung von Abonnements werden die in der VDV 301-2-1 beschriebenen Datenstrukturen verwendet.

7.2.4 Datenstruktur der Operation UnsubscribeSystemConfiguration

Für das Beenden von Abonnements werden die in der VDV 301-2-1 beschriebenen Datenstrukturen verwendet.

7.2.5 Datenstruktur der Operation StoreSystemConfiguration

7.2.5.1 Request

SystemDocumentationService.StoreSystemConfigurationRequest			*+Structure*	**Struktur für die Übertragung einer zu speichernden Systemkonfiguration**
	ServiceStartList	**1:1**	*+ServiceStartList*	Liste mit Informationen der zu startenden Dienste (siehe auch VDV 301-2-1)
	DeviceSpecificationList	**1:1**	*+DeviceSpecificationList*	Liste mit Informationen der vorhandenen Geräte (siehe auch VDV 301-2-1)
	HertbeatIntervall	0:1	*IBIS-IP.duration*	Wert zur Konfiguration des Intervalls für die Heartbeat-Abfrage des SystemManagementService

Tabelle 26 Beschreibung von SystemDocumentationService.StoreSystemConfigurationRequest

7.2.5.2 Response

Da es sich um eine Übermittlung von Informationen an einen Dienst handelt, wird die Antwort mit der dafür vorgesehenen allgemeinen Struktur gegeben (siehe VDV 301-2-1).

7.2.6 Datenstruktur der Operation StoreLogMessages

7.2.6.1 Request

SystemDocumentationService.StoreLogMessagesRequest			*+Structure*	**Struktur für die Übertragung einer zu speichernden Log-Meldung**
	LogMessage	0:*	*+LogMessage*	Information über die zu speichernde Nachricht (siehe auch VDV 301-2-1).
	DataVersion	0:*	*+DataVersion*	Information über die zu speichernde Datenversion(siehe auch VDV 301-2-1)

Tabelle 27 Beschreibung von SystemDocumentationService.StoreLogMessagesRequest

7.2.6.2 Response

Da es sich um eine Übermittlung von Informationen an einen Dienst handelt, wird die Antwort mit der dafür vorgesehenen allgemeinen Struktur gegeben (siehe VDV 301-2-1).

7.2.7 Datenstruktur der Operation RetrieveLogMessages

7.2.7.1 Request

SystemDocumentationService.RetrieveLogMessagesRequest			*+Structure*	**Struktur für die Abfrage von Log-Meldungen**
	InformationType	**1:1**	*SystemDocumentationInformationEnumeration*	Information über die Art der zu liefernden Log-Meldungen (siehe auch Kapitel 9.10)
	NumberOfEntries	**1:1**	*IBIS-IP.int*	Anzahl an zu liefernden Meldungen

Tabelle 28 Beschreibung von SystemDocumentationService.RetrieveLogMessagesRequest

7.2.7.2 Response

SystemDocumentationService.RetrieveLogMessagesResponse				*+Structure*	**Struktur für die Beantwortung eine Anfrage nach Log-Meldungen**
				choice	eine der beiden folgenden Strukturen
	a	***LogMessageData***	**–1:1**	*+SystemDocumentationService.LogMessageData*	ausführliche Antwortstruktur (siehe VDV 301-2-1)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Rückmeldung von Fehlern

Tabelle 29 Beschreibung von SystemDocumentationService.RetrieveLogMessagesResponse

SystemDocumentationService.SystemConfigurationData			*+Structure*	**Struktur zur Beschreibung der Systemkonfiguration**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Angabe des Zeitpunkts der Antwort
	Message	**1:***	*+Message*	Nachrichteninhalt (siehe auch VDV 301-2-1)

Tabelle 30 Beschreibung von SystemDocumentationService.SystemConfigurationData

7.3 Dienst SystemManagementService

Der Dienst `SystemManagementService` ist in IBIS-IP die Umsetzung der Aufgaben der Fachkomponente System-Management. Seine Hauptaufgaben bestehen darin,

— alle Dienste, die laut Systemkonfiguration zum IBIS-IP-System gehören sollen, zu starten (vgl. Kapitel 5.2 bis 5.3)

— den Status aller Geräte und Dienste des IBIS-IP-Systems zu überwachen

— den Status aller Geräte und Dienste des IBIS-IP-Systems zu veröffentlichen

Dieser Dienst wird auf dem Gerät, auf dem per Gerätekonfiguration festgelegt ist, dass er gestartet werden soll, mit dem Ausführen der Applikation auf dem Gerät automatisch gestartet.

In einem IBIS-IP-System darf es grundsätzlich nur einen *SystemManagementService* geben. Dieser *SystemManagementService* darf sich mit nur einer IBIS-IP-Version eines *SystemDocumentationService* verbinden.

Der Parallelbetrieb zweier *SystemManagementServices* (z. B. mit zwei verschiedenen IBIS-IP-Versionen) entspräche dem Parallelbetrieb zweier IBIS-IP-Systeme. Ein solcher Zustand ist nicht zulässig.

Da das Hauptaugenmerk auf einer Sicherstellung der Informationsübertragung liegt, ist dieser Dienst in IBIS-IP als HTTP-Dienst umgesetzt worden.

7.3.1 Operationen des SystemManagementService

Operation	**Request/ Response**	**Verwendeter Datentyp, Datenstruktur**
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

Tabelle 31 Beschreibung von Operationen des SystemManagementService

7.3.2 Datenstrukturen der Operation GetDeviceStatus

7.3.2.1 Request

Beim Request der Operation ***GetDeviceStatus*** werden keine Daten übergeben.

7.3.2.2 Response

SystemManagementService.GetDeviceStatusResponse				*+Structure*	**Struktur für die Beantwortung einer Anfrage nach dem Status aller Geräte eines IBIS-IP-Systems**
				choice	eine der beiden folgenden Möglichkeiten
	a	***SystemManagementService.GetDeviceStatusResponseData***	**–1:1**	*+SystemManagementService.GetDeviceStatusResponseData*	Ausführliche Antwortstruktur für Angaben zum Status aller Geräte eines IBIS-IP-Systems (siehe unten)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Rückmeldung von Fehlern

Tabelle 32 Beschreibung von SystemManagementService.GetDeviceStatusResponse

SystemManagementService.GetDeviceStatusResponseData			*+Structure*	**Ausführliche Antwortstruktur für Angaben zum Status aller Geräte eines IBIS-IP-Systems**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Angabe des Zeitpunkts der Antwort
	DeviceSpecificationWithStateList	**1:1**	*+DeviceSpecificationWithStateList*	Verweis auf eine Liste mit Geräten und deren zugehörigen Zuständen (vgl. VDV 301-2-1)

Tabelle 33 Beschreibung von SystemManagementService.GetDeviceStatusResponseData

7.3.3 Datenstrukturen der Operation SubscribeDeviceStatus

Für die Einrichtung von Abonnements werden die in der VDV 301-2-1 beschriebenen Datenstrukturen verwendet.

7.3.4 Datenstrukturen der Operation UnsubscribeDeviceStatus

Für das Beenden von Abonnements werden die in der VDV 301-2-1 beschriebenen Datenstrukturen verwendet.

7.3.5 Datenstrukturen der Operation GetServiceStatus

7.3.5.1 Request

Beim Request der Operation ***`GetServiceStatus`*** werden keine Daten übergeben.

7.3.5.2 Response

SystemManagementService.GetServiceStatusResponse				*+Structure*	**Struktur für die Beantwortung einer Anfrage nach dem Status aller Dienste eines IBIS-IP-Systems**
				choice	eine der beiden folgenden Möglichkeiten
	a	***SystemManagementService.GetServiceStatusResponseData***	**–1:1**	*+SystemManagementService.GetServiceStatusResponseData*	Ausführliche Antwortstruktur für Angaben zum Status aller Dienste eines IBIS-IP-Systems (siehe unten)
	b	***OperationErrorMessage***		*IBIS-IP.string*	Rückmeldung von Fehlern

Tabelle 34 Beschreibung von SystemManagementService.GetServiceStatusResponse

SystemManagementService.GetServiceStatusResponseData			*+Structure*	**Ausführliche Antwortstruktur für Angaben zum Status aller Dienste eines IBIS-IP-Systems**
	TimeStamp	**1:1**	*IBIS-IP.dateTime*	Angabe des Zeitpunkts der Antwort
	ServiceIdentificationWithStateList	**1:1**	*+ServiceIdentificationWithStateList*	Verweis auf eine Liste mit Diensten und deren zugehörigen Zuständen (vgl. VDV 301-2-1)

Tabelle 35 Beschreibung von SystemManagementService.GetServiceStatusResponseData

7.3.6 Datenstrukturen der Operation SubscribeSystemStatus

Für die Einrichtung von Abonnements werden die in der VDV 301-2-1 beschriebenen Datenstrukturen verwendet.

Datenstrukturen der Operation UnsubscribeSystemStatus

Für das Beenden von Abonnements werden die in der VDV 301-2-1 beschriebenen Datenstrukturen verwendet.

8 Besondere IBIS-IP-Datentypen

8.1 IBIS-IP.anyURI

IBIS-IP.anyURI			*+Structure*	**IBIS-IP-Struktur zur Beschreibung eines Adresswertes**
	Value	**1:1**	*xs:anyURI*	Angabe des eigentlichen Wertes
	ErrorCode	0:1	*ErrorCodeEnumeration*	Wert für einen aufgetretenen Fehler

Tabelle 36 Beschreibung von IBIS-IP.anyURI

8.2 IBIS-IP.boolean

IBIS-IP.boolean			*+Structure*	**IBIS-IP-Struktur zur Beschreibung eines boolschen Wertes**
	Value	**1:1**	*xs:boolean*	Angabe des eigentlichen Wertes
	ErrorCode	0:1	*ErrorCodeEnumeration*	Wert für einen aufgetretenen Fehler

Tabelle 37 Beschreibung von IBIS-IP.boolean

8.3 IBIS-IP.byte

IBIS-IP.byte			*+Structure*	**IBIS-IP-Struktur zur Beschreibung eines Byte-Wertes**
	Value	**1:1**	*xs:byte*	Angabe des eigentlichen Wertes
	ErrorCode	0:1	*ErrorCodeEnumeration*	Wert für einen aufgetretenen Fehler

Tabelle 38 Beschreibung von IBIS-IP.byte

8.4 IBIS-IP.date

IBIS-IP.date			*+Structure*	**IBIS-IP-Struktur zur Beschreibung eines Datumswertes**
	Value	**1:1**	*xs:date*	Angabe des eigentlichen Wertes
	ErrorCode	0:1	*ErrorCodeEnumeration*	Wert für einen aufgetretenen Fehler

Tabelle 39 Beschreibung von IBIS-IP.date

8.5 IBIS-IP.dateTime

IBIS-IP.dateTime			*+Structure*	**IBIS-IP-Struktur zur Beschreibung eines Datumswertes mit Zeit**
	Value	**1:1**	*xs:dateTime*	Angabe des eigentlichen Wertes
	ErrorCode	0:1	*ErrorCodeEnumeration*	Wert für einen aufgetretenen Fehler

Tabelle 40 Beschreibung von IBIS-IP.dateTime

8.6 IBIS-IP.double

IBIS-IP.double			*+Structure*	**IBIS-IP-Struktur zur Beschreibung eines Gleitkommawertes**
	Value	**1:1**	*xs:double*	Angabe des eigentlichen Wertes
	ErrorCode	0:1	*ErrorCodeEnumeration*	Wert für einen aufgetretenen Fehler

Tabelle 41 Beschreibung von IBIS-IP.double

8.7 IBIS-IP.duration

IBIS-IP.duration			*+Structure*	**IBIS-IP-Struktur zur Beschreibung einer Zeitdauer**
	Value	**1:1**	*xs:duration*	Angabe des eigentlichen Wertes
	ErrorCode	0:1	*ErrorCodeEnumeration*	Wert für einen aufgetretenen Fehler

Tabelle 42 Beschreibung von IBIS-IP.duration

8.8 IBIS-IP.int

IBIS-IP.int			*+Structure*	**IBIS-IP-Struktur zur Beschreibung eines Ganzzahlwertes**
	Value	**1:1**	*xs:int*	Angabe des eigentlichen Wertes
	ErrorCode	0:1	*ErrorCodeEnumeration*	Wert für einen aufgetretenen Fehler

Tabelle 43 Beschreibung von IBIS-IP.int

8.9 IBIS-IP.language

IBIS-IP.language			*+Structure*	**IBIS-IP-Struktur zur Beschreibung einer Sprachangabe**
	Value	**1:1**	*xs:language*	Angabe des eigentlichen Wertes
	ErrorCode	0:1	*ErrorCodeEnumeration*	Wert für einen aufgetretenen Fehler

Tabelle 44 Beschreibung von IBIS-IP.language

8.10 IBIS-IP.NMTOKEN

IBIS-IP.NMTOKEN			*+Structure*	**IBIS-IP-Struktur zur Beschreibung eines Index-Wertes**
	Value	**1:1**	*xs:NMTOKEN*	Angabe des eigentlichen Wertes
	ErrorCode	0:1	*ErrorCodeEnumeration*	Wert für einen aufgetretenen Fehler

Tabelle 45 Beschreibung von IBIS-IP.NMTOKEN

8.11 IBIS-IP.nonNegativeInteger

IBIS-IP.nonNegativeInteger			*+Structure*	**IBIS-IP-Struktur zur Beschreibung einer nicht negativen Ganzzahl**
	Value	**1:1**	*xs:nonNegativeInteger*	Angabe des eigentlichen Wertes
	ErrorCode	0:1	*ErrorCodeEnumeration*	Wert für einen aufgetretenen Fehler

Tabelle 46 Beschreibung von IBIS-IP.nonNegativeInteger

8.12 IBIS-IP.normalizedString

IBIS-IP.normalizedString			*+Structure*	**IBIS-IP-Struktur zur Beschreibung eines normalisierten Strings**
	Value	**1:1**	*xs:normalizedString*	Angabe des eigentlichen Wertes
	ErrorCode	0:1	*ErrorCodeEnumeration*	Wert für einen aufgetretenen Fehler

Tabelle 47 Beschreibung von IBIS-IP.normalizedString

8.13 IBIS-IP.string

IBIS-IP.string			*+Structure*	**IBIS-IP-Struktur zur Beschreibung eines Stringwertes**
	Value	**1:1**	*xs:string*	Angabe des eigentlichen Wertes
	ErrorCode	0:1	*ErrorCodeEnumeration*	Wert für einen aufgetretenen Fehler

Tabelle 48 Beschreibung von IBIS-IP.string

8.14 IBIS-IP.time

IBIS-IP.time			*+Structure*	**IBIS-IP-Struktur zur Beschreibung eines Zeitwertes**
	Value	**1:1**	*xs:time*	Angabe des eigentlichen Wertes
	ErrorCode	0:1	*ErrorCodeEnumeration*	Wert für einen aufgetretenen Fehler

Tabelle 49 Beschreibung von IBIS-IP.time

8.15 IBIS-IP.unsignedInt

IBIS-IP.unsignedInt			*+Structure*	**IBIS-IP-Struktur zur Beschreibung einer vorzeichenlosen Ganzzahl**
	Value	**1:1**	*xs:unsignedInt*	Angabe des eigentlichen Wertes
	ErrorCode	0:1	*ErrorCodeEnumeration*	Wert für einen aufgetretenen Fehler

Tabelle 50 Beschreibung von IBIS-IP.unsignedInt

8.16 IBIS-IP.unsignedLong

IBIS-IP.unsignedLong			*+Structure*	**IBIS-IP-Struktur zur Beschreibung einer vorzeichenlosen Ganzzahl**
	Value	**1:1**	*xs:unsignedLong*	Angabe des eigentlichen Wertes
	ErrorCode	0:1	*ErrorCodeEnumeration*	Wert für einen aufgetretenen Fehler

Tabelle 51 Beschreibung von IBIS-IP.unsignedLong

8.17 InternationalTextType

InternationalTextType			*+Structure*	**IBIS-IP-Struktur zur Beschreibung eines Fremdsprachigen Textes**
	Value	**1:1**	*IBIS-IP.string*	Angabe des eigentlichen Wertes
	Language	**1:1**	*IBIS-IP.language*	Angabe der Sprache
	ErrorCode	0:1	*ErrorCodeEnumeration*	Wert für einen aufgetretenen Fehler

Tabelle 52 Beschreibung von InternationalTextType

9 Gemeinsam genutzte Aufzählungstypen

Das nachfolgende Kapitel beschreibt die in IBIS-IP verwendeten Aufzählungstypen, die abhängig vom Kontext im Datenaustausch gesetzt werden müssen.

9.1 ConnectionStateEnumeration

Bezeichnung des Aufzählungstypen	Mögliche Werte	Beschreibung
ConnectionStateEnumeration	ConnectionBroken ConnectionOK NoInformationAvailable	Angabe über den Sicherungszustand der Umsteigebeziehung

Tabelle 53 Beschreibung von ConnectionStateEnumeration

9.2 ConnectionTypeEnumeration

Bezeichnung des Aufzählungstypen	Mögliche Werte	Beschreibung
ConnectionTypeEnumeration	Interchange ProtectedConnection	Angabe über die Art der Umsteigebeziehung

Tabelle 54 Beschreibung von ConnectionTypeEnumeration

9.3 DeviceClassEnumeration

Bezeichnung des Aufzählungstypen	Mögliche Werte	Beschreibung
DeviceClassEnumeration	OnBoardUnit SideDisplay FrontDisplay InteriorDisplay Validator TicketVendingMachine AnnouncementSystem MMI VideoSystem APC MobileInterface Other TestDevice	Angabe über die Geräteklasse (vgl. Kapitel 1.4)

Tabelle 55 Beschreibung von DeviceClassEnumeration

9.4 DeviceStateEnumeration

Bezeichnung des Aufzählungstypen	Mögliche Werte	Beschreibung
DeviceStateEnumeration	defective notavailable running	Angabe über den Status eines Geräts (vgl. Kapitel 4.3.1)

Tabelle 56 Beschreibung von DeviceStateEnumeration

9.5 ErrorCodeEnumeration

Bezeichnung des Aufzählungstypen	Mögliche Werte	Beschreibung
ErrorCodeEnumeration	DataEstimated FaultData NoScheduleDataAvailable DeviceMissing NoServiceResponse ImportantDataNotAvailable DataNotValid	Angabe über einen Fehlerzustand (vgl. Kapitel 4.3.2)

Tabelle 57 Beschreibung von ErrorCodeEnumeration

9.6 IBIS-IP-VersionEnumeration

Bezeichnung des Aufzählungstypen	Mögliche Werte	Beschreibung
IBIS-IP-VersionEnumeration	1.0	Angabe der IBIS-IP-Version

Tabelle 58 Beschreibung von IBIS-IP-VersionEnumeration

9.7 MessageTypeEnumeration

Bezeichnung des Aufzählungstypen	Mögliche Werte	Beschreibung
MessageTypeEnumeration	Status Warning Error	Angabe über die Art einer Nachricht

Tabelle 59 Beschreibung von MessageTypeEnumeration

9.8 ServiceNameEnumeration

Bezeichnung des Aufzählungstypen	Mögliche Werte	Beschreibung
ServiceNameEnumeration	BeaconLocationService CustomerInformationService DeviceManagementService	Name des Dienstes (vgl. Kapitel 3.2)

Bezeichnung des Aufzählungstypen	Mögliche Werte	Beschreibung
	DistanceLocationService GNSSLocationService JourneyInformationService NetworkLocationService PassengerCountingService SystemDocumentationService SystemManagementService TicketingService TimeService TestService	

Tabelle 60 Beschreibung von ServiceNameEnumeration

9.9 ServiceStateEnumeration

Bezeichnung des Aufzählungstypen	Mögliche Werte	Beschreibung
ServiceStateEnumeration	defective notrunning running starting standby	Angabe zum Status eines Diensts

Tabelle 61 Beschreibung von ServiceStateEnumeration

9.10 SystemDocumentationInformationEnumeration

Bezeichnung des Aufzählungstypen	Mögliche Werte	Beschreibung
SystemDocumentationInformationEnumeration	ErrorMessage StatusMessage WarningMessage All	Art der Information

Tabelle 62 Beschreibung von SystemDocumentationInformationEnumeration

10 Begriffe

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

Abkürzung	**Beschreibung**
Abonnement	Mit einem solchen Verfahren wird der anfragende Teilnehmer automatisch mit aktuellen Informationen versorgt.
AnnouncementSystem	Beschreibt die Geräteklasse Elektroakustische-Anlage inkl. Fahrgastsprechstellen.
Applikation	Als Applikation wird die auf den Geräten laufende Software bezeichnet. Diese herstellerspezifische Software bedient allerdings spezifizierte Schnittstellen.
BeaconLocationService	Ein IBIS-IP Dienst der die Informationen der Ortsbaken übermittelt.
CustomerInformationService	Als zentrale Informationsstelle für alle Belange der Fahrgastinformation sorgt in IBIS-IP dieser Dienst für eine konsistente Bereitstellung aller Daten.
DeviceManagementService	Der auf jedem Gerät in IBIS-IP vorherrschender Dienst stellt Informationen über das Gerät und die laufenden Dienste bereit.
DistanceLocationService	Die Auswertung der Odometerimpulse aus der Wegstrecke übernimmt in IBIS-IP dieser Dienst.
FrontDisplay	Entspricht der Geräteklasse Fahrzeugzielanzeige.
Geräteklassen	Zur Transparenz der angeschlossenen Geräte im Netzwerk wurden Geräteklassen spezifiziert, die alle derzeit denkbaren Geräte abdecken.
GNSS-Koordinaten	Auf Grund einer satellitenbasierte Ortung (z. B. GPS, Galileo) gewonnene Koordinaten eines Punktes.
GNSSLocationService	Für die Verteilung von in Koordinaten beschreibbaren aktuellen Standort des Fahrzeugs auf Basis des NMEA Telegramms.
HTTP-GET	Es handelt sich um eine Anfragefunktion des HTTP. I. d. R. werden dieser Frage keine Daten mit gesendet.
HTTP-POST	Es handelt sich um eine Anfragefunktion des HTTP. I. d. R. werden dieser Frage Daten mit gesendet.
HTTP-Protokoll	Mit dieser Bezeichnung wird innerhalb von IBIS-IP eine Übertragungsart für sich eher selten ändernde Informationen verwendet.
HTTP-Protokoll-Stack	Bezeichnet die komplette Protokolleinheit (von TCP bis HTTP) die für eine sichere Übertragung notwendig ist.

Abkürzung	Beschreibung
InteriorDisplay	Beschreibt die Geräteklasse der Anzeiger im Innenraum des Fahrzeuges, des können Fahrgastinformationsanzeiger aber auch digitale Werbeanzeiger sein.
JourneyInformationService	Dieser Dienst bietet Fahrplandaten den abfragenden Diensten zur Verfügung.
JourneyWeb	Ist ein XML-Protokoll für Fahrgastinformation.
MobileInterface	Beschreibt die Geräteklasse der Schnittstelle an der die Fahrgäste mit ihrem mobilen Endgerät Informationen vom Fahrzeug abholen können.
Multicast	Kommunikationsverfahren zum Ansprechen vieler dedizierter Empfänger.
Multicast-Gruppe	Beschreibt die dedizierten Empfänger einer Multicast-Nachricht.
NetworkLocationService	Der Dienst stellt Informationen über den aktuellen Standort auf einem geplanten Linienfahrweg in einem Netz des ÖV bereit.
Odometer	Auf raddrehzahl basierter Entfernungsmesser, der eine bestimmte Anzahl Impuls pro gefahrenem Meter produziert.
OnBordUnit	Entspricht der Geräteklasse des Zentralrechners aus IBIS Zeiten.
Other (Geräteklasse)	Hierunter lassen sich Geräte anbinden die noch nicht in diesem Standarad Klassifiziert wurden.
PassengerCountigService	Dieser Dienst stellt die Fahrgastzähldaten einer jeder Tür dem IBIS-IP-System zur Verfügung.
Port	Ist eine „Zugangstür“ des Protokolls, das der Applikation eine eindeutige Quellenzuordnung ermöglicht
Request/Response	Beschreibt den Kommunikationsablauf indem einer Frage von Teilnehmer A an Teilnehmer B eine Antwort von Teilnehmer B folgt.
SideDisplay	Beschreibt die Geräteklasse der seitliche Aussenanzeige an einem Fahrzeug.
SRV-Records	Gemäß RFC 2782 lassen sich mit diesem Textfeld zusätzliche Informationen (z. B. Rechnername) an den DNS-Dienst anhängen.
SystemDocumentationService	Der Dienst ist in IBIS-IP dafür verantwortlich Systemmeldungen zu dokumentieren und die Systemkonfiguration zur Verfügung zu stellen.

Abkürzung	Beschreibung
SystemManagementService	Die Hauptaufgaben bestehen darin, über den aktuellen Status der Geräte und der Dienste zu informieren, sowie das System zu koordinieren.
TestDevice	Platzhalter für die Geräteklasse der Testgeräte, die im Falle einer Verifikation des Systems angeschlossen werden.
TicketingService	Der Dienst stellt Ticketing Funktionen im IBIS-IP-System zur Verfügung.
TicketVendingMachine	Beschreibt die Geräteklasse der Fahrscheinverkaufsgeräte innerhalb des Fahrzeuges. Dies können fahrerbediente aber auch autarke Automaten sein.
TimeService	Der Zeitdienst stellt via SNTP dem System die aktuelle Uhrzeit, das Datum und die Zeitzone zur Verfügung.
TransModel	Ist ein Referenz Daten Model nach EN12896.
TXT-Records	Gemäß RFC 1464 lassen sich mit diesem Textfeld zusätzliche Informationen (z. B. Rechnername) an den DNS-Dienst anhängen.
UDP-Protokoll	Mit dieser Bezeichnung wird innerhalb von IBIS-IP eine Übertragungsart für sich zyklisch ändernde Informationen verwendet.
UML-Konvention	Formatvorgaben aus der UML-Programmiersprache
USB-Stick	Mobiler Datenspeicher
Validator	Geräteklasse für Stempelautomat oder elektronischer Entwerter für E-Tickets.
VideoSystem	Beschreibt die Geräteklasse des Videoüberwachungssystems, das über IBIS-IP gesteuert wird bzw. Videodaten überträgt
Zeitserver	Ist eine zentrale Komponente innerhalb von IBIS-IP, die die Systemzeit zur Verfügung stellt.
Zero Conf	Ist eine Methode und eine Zusammenfassung spezieller Technologien zur automatischen Koppelung von Geräten in einem IP Netzwerk

Regelwerke – Normen und Empfehlungen

(1)	**CEN/TS 13149-7**	**Öffentlicher Verkehr - Planungs- und Steuerungssysteme für Straßenfahrzeuge - Teil 7: IP-basierende Vernetzung in einem Fahrzeug, Netzwerk- und Systemarchitektur (FprCEN/TS 13149-7:2015)**
(2)	CEN/TS 13149-8	Öffentlicher Verkehr - Planungs- und Steuerungssysteme für Straßenfahrzeuge - Teil 8: Physikalische Schicht für IP-Kommunikation; Englische Fassung CEN/TS 13149-8:2013
(3)	EN 15531	Service Interface for Real Time Information
(4)	EN 1545	Identification card systems.
(5)	VDV 301-1	Internetprotokoll basiertes integriertes Bordinformationssystem IBIS-IP - Teil 1: Systemarchitektur
(6)	VDV 301-2	Internetprotokoll basiertes integriertes Bordinformationssystem IBIS-IP - Teil 2: Schnitstellenspezifikation
(7)	VDV 301-2-1	IBIS-IP Beschreibung der Dienste, Gemeinsame Datenstrukturen und Aufzählungstypen
(8)	VDV 300	Integriertes Bordinformationssystem IBIS
(9)	VDV 3001	Kommunikation im ÖV (IP-KOM-ÖV) - Technische Anforderungen für Anwendungen im Integrierten Bordinformationssystem (IBIS)
(10)	VDV 453	VDV-Ist-Datenschnittstellen
(11)	VDV 454	VDV-Ist-Datenschnittstellen
(12)	Bibliotheken „Bonjour“, „Avahi“	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.
(13)	RFC 2927	MIME Directory Profile for LDAP Schema
(14)	RFC 6365	Terminology Used in Internationalization in the IETF
(15)	RFC 2782	A DNS RR for specifying the location of services (DNS SRV)
(16)	RFC 1035	Domain Names – Iimplementation an Specification
(17)	RFC 1464	Using the Domain Name System to Store Arbitrary String Attributes
(18)	RFC 4330	Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI

Die IBIS-IP XSD-Dateien stehen unter www.vdv.de/ip-kom-oev.aspx zum Download bereit.

Bildverzeichnis

Abbildung 1	Beispiel zu den verschiedenen Repräsentationen von Fachkomponenten. Schwarze Linien: IBIS-IP, rote Linien: proprietär. Man beachte, dass die Fachkomponente Fahrgast-Anzeiger sowohl Geräte-, Applikations- wie auch Diensteigenschaften hat.	24
Abbildung 2	Darstellung der Strukturierung von Daten in Diensten	29
Abbildung 3	Beispielhafter Startup eines IBIS-IP-Systems	49

Tabellenverzeichnis

Tabelle 1	IBIS-IP Geräteklassen Klassifikation (Sortierung beliebig)	18
Tabelle 2	Bedeutungen der SRV-Records in DNS-SD	26
Tabelle 3	Bedeutungen der TXT-Records in DNS-SD (inklusive Festlegungen für IBIS-IP)	27
Tabelle 4	Beispiel für Operationsnotation in IBIS-IP	51
Tabelle 5	Beispiel (für Kapitel 7 für die tabellarische Notation einer XML-Struktur)	52
Tabelle 6	Beschreibung von Operationen des DeviceManagementService	56
Tabelle 7	Beschreibung DeviceManagementService.GetDeviceInformationResponse	57
Tabelle 8	Beschreibung DeviceManagementService.GetDeviceInformationResponseData	57
Tabelle 9	Beschreibung DeviceManagementService.GetDeviceConfigurationResponse	58
Tabelle 10	Beschreibung DeviceManagementService.GetDeviceConfigurationResponseData	58
Tabelle 11	Beschreibung DeviceManagementService.SetDeviceConfigurationRequest	58
Tabelle 12	Beschreibung DeviceManagementService.GetDeviceStatusResponse	59
Tabelle 13	Beschreibung DeviceManagementService.GetDeviceStatusResponseData	59
Tabelle 14	Beschreibung DeviceManagementService.GetDeviceErrorMessagesResponse	60
Tabelle 15	Beschreibung DeviceManagementService.GetDeviceErrorMessagesResponseData	60
Tabelle 16	Beschreibung DeviceManagementService.GetServiceInformationResponse	61
Tabelle 17	Beschreibung DeviceManagementService.GetServiceInformationResponseData	61
Tabelle 18	Beschreibung DeviceManagementService.GetServiceStatusResponse	62
Tabelle 19	Beschreibung DeviceManagementService.GetServiceStatusResponseData	62
Tabelle 20	Beschreibung DeviceManagementService.StartServiceRequestStructure	62
Tabelle 21	Beschreibung DeviceManagementService.StopServiceRequestStructure	63
Tabelle 22	Beschreibung DeviceManagementService.RestartServiceRequestStructure	63

Tabelle 23	Beschreibung von Operationen des SystemDocumentationService	64
Tabelle 24	Beschreibung von SystemDocumentationService.GetSystemConfigurationResponse	64
Tabelle 25	Beschreibung von SystemDocumentationService.SystemConfigurationData	65
Tabelle 26	Beschreibung von SystemDocumentationService.StoreSystemConfigurationRequest	65
Tabelle 27	Beschreibung von SystemDocumentationService.StoreLogMessagesRequest	65
Tabelle 28	Beschreibung von SystemDocumentationService.RetrieveLogMessagesRequest	66
Tabelle 29	Beschreibung von SystemDocumentationService.RetrieveLogMessagesResponse	66
Tabelle 30	Beschreibung von SystemDocumentationService.SystemConfigurationData	66
Tabelle 31	Beschreibung von Operationen des SystemManagementService	67
Tabelle 32	Beschreibung von SystemManagementService.GetDeviceStatusResponse	68
Tabelle 33	Beschreibung von SystemManagementService.GetDeviceStatusResponseData	68
Tabelle 34	Beschreibung von SystemManagementService.GetServiceStatusResponse	68
Tabelle 35	Beschreibung von SystemManagementService.GetServiceStatusResponseData	69
Tabelle 36	Beschreibung von IBIS-IP.anyURI	70
Tabelle 37	Beschreibung von IBIS-IP.boolean	70
Tabelle 38	Beschreibung von IBIS-IP.byte	70
Tabelle 39	Beschreibung von IBIS-IP.date	70
Tabelle 40	Beschreibung von IBIS-IP.dateTime	71
Tabelle 41	Beschreibung von IBIS-IP.double	71
Tabelle 42	Beschreibung von IBIS-IP.duration	71
Tabelle 43	Beschreibung von IBIS-IP.int	71
Tabelle 44	Beschreibung von IBIS-IP.language	71
Tabelle 45	Beschreibung von IBIS-IP.NMTOKEN	72
Tabelle 46	Beschreibung von IBIS-IP.nonNegativeInteger	72
Tabelle 47	Beschreibung von IBIS-IP.normalizedString	72
Tabelle 48	Beschreibung von IBIS-IP.string	72
Tabelle 49	Beschreibung von IBIS-IP.time	73
Tabelle 50	Beschreibung von IBIS-IP.unsignedInt	73

Tabelle 51	Beschreibung von IBIS-IP.unsignedLong	73
Tabelle 52	Beschreibung von InternationalTextType	73
Tabelle 53	Beschreibung von ConnectionStateEnumeration	74
Tabelle 54	Beschreibung von ConnectionTypeEnumeration	74
Tabelle 55	Beschreibung von DeviceClassEnumeration	74
Tabelle 56	Beschreibung von DeviceStateEnumeration	75
Tabelle 57	Beschreibung von ErrorCodeEnumeration	75
Tabelle 58	Beschreibung von IBIS-IP-VersionEnumeration	75
Tabelle 59	Beschreibung von MessageTypeEnumeration	75
Tabelle 60	Beschreibung von ServiceNameEnumeration	76
Tabelle 61	Beschreibung von ServiceStateEnumeration	76
Tabelle 62	Beschreibung von SystemDocumentationInformationEnumeration	76

Impressum

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

Ansprechpartner

Dipl.-Ing. Berthold Radermacher
T 0221 57979-141
F 0221 57979-8141
radermacher@vdv.de

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