

Betrifft: **Universelles Kommunikationssystem für Luftgüte-Messnetzkomponenten wie Zentralen, Stationsrechner, Messgeräte u.a.m. auf Basis Netzwerk-orientierter Datenaustauschmethoden und –verfahren**

## Änderungsliste:

---

Vers: 1.0	A. Jakob (AJ), E. Kitzmüller (EK), H. Ehm (HE) / HORIBA (Tulln)	
	Erstentwurf mit Grundlagendefinitionen und grober Umfang-/ Aufwandsbestimmung	28.05.2015
Vers: 1.1.1	EK	Protokolldetails zu 4.4 DATENTRANSFER-ÜBERTRAGUNGSSPEZIFIKA
		16.06.2015
Vers: 1.1.2	AJ	Überarbeitung 2.4/2.5/2.6 .... Vorbedingungen
		17.06.2015
Vers: 1.1.3	AJ, HE	Überarbeitung 1.3 ... Umsetzungsplanung
		22.09.2015
Vers: 1.2	EK, AJ	Neu – 5. Geräteerkennungsprotokoll
		02.11.2015
Vers: 1.3	EK, AJ	Überarbeitung nach Feedback von M.Junkers/Leineweber
		17.11.2015
Vers: 1.4	EK	JSON-Strukturen und Descriptoren geändert
		10.03.2016
Vers: 1.5	EK	Neu – Liste der möglichen Stati und Modi bei Selbstbeschreibung(*)
		14.03.2017

---

## Preamble

Die steigenden Erfordernisse der Datenübertragung, sowie die gestiegenen Anforderungen betreffend Qualitätssicherung können mit den bestehenden Übertragungstechniken (z.B. Bayern-Hessen-Protokoll, GESYTEC-Protokoll u.ä.), auch in Ihrer LAN-eingebetteten Ausformung, immer weniger ausreichend messnetzübergreifend abgedeckt werden.

Durch die neuen Staubdirektive, die EU-Richtlinie zur Qualitätssicherung mit ihren entsprechenden EN-Normen und damit auch in nationales Recht umgesetzten Ausformungen, trifft dies insbesondere auf die, damit ausgelösten gerätetechnischen Diagnoseerweiterungen zu, welche dadurch größtenteils im Moment nicht ausreichend oder nur herstellereigen übertragen werden können.

Weiters stoßen die damit verbundenen und benötigten Verarbeitungsgeschwindigkeiten mit den ihnen unterlegten zumeist seriellen Abläufen (Protokollen) ebenso an ihre Grenzen, abgesehen von den damit steigenden Übertragungskosten.

Diese Erkenntnis mündete schon in der UBIS-Usertagung aus 2009 in den Beschluss, eine neue Technik einzuführen, wobei auf der Tagung in 2010 in Richtung Zentrale – Stationsrechner bereits eine Teillösung in Echtzeit vorgeführt werden konnte. Seitdem sind alle Messnetze mit UBIS-Servern mit dieser ersten Teillösung in Betrieb genommen worden.

### (\*) zur Änderung:

Im Pkt. 4.4.8 wurden neue Descriptoren aufgenommen, die eine Liste der möglichen Betriebsmodi („OperatingModes“), eine Liste der möglichen Betriebsstati („OperatingStatuses“) und eine Liste der möglichen Fehlerstati („ErrorStatuses“) liefern.

## 1. Aufgabenstellung

## 1.1 Zielsetzung

Die Schaffung von –

- einem auf TCP/IP aufsetzenden, entwickelten universell einsetzbaren Datenaustauschprotokolls für Immission, das mit Programmkomponenten, die der „GNU Lesser General Public License“ (LGPL) unterliegen, implementiert werden kann<sup>1</sup>, als Ablöse der weiter oben erwähnten existierenden (BHP, GESYTEC u.a.m.)

sowie

- einem darauf basierenden MG-Interface, welches als direkte und ein BHP-IF ersetzende Schnittstelle für jegliche Art von Messdatenerfassung, also auch zur Nachrüstung von z.B. Messgeräte (MG) der Serie HORIBA 360/370 dienen kann.

Damit soll die Weitergabe sowohl echtzeitnaher und damit laufend zu übertragender Messdaten, als auch abgesetzter bzw. archivierter Messdatenreihen, optional samt jeweils ihren Attributen, ermöglicht werden.

## 1.2 Umsetzung in ein Produkt

### Detailprojekte:

Es sind zwei Entwicklungsstufen vorgesehen und folgende Ausformung dazu geplant –

- 1. Stufe: Basiskonzept und Festlegung des Funktionsumfangs [ **Endredaktion Dez'15** ] initial mit Fa. Horiba kooperativ; Ergänzungsbeteiligungen (\* - angefragt): EAS-Envimet (\*), Grimm, Leineweber, PALAS
- 2a. Stufe: Umsetzung je eines Interface(**NetworkProtocol**)Converter(UDAP-IFC ) basierend auf Raspberry-Pi 2 Modell B, durch AIP >
  - > als B-H-P <> UIDAP Interface für SHARP 5030, [ **Feldtest: seit Sep'15** ]
  - > als IF-Stecker kompatibles LAN-Interface für HORIBA Serie 370 Messgeräte
- 2b. Stufe: Softwareentwicklung (SWE) zu Datenaustausch-Protokoll und MG-Treiberprototyp zu „UIDAP“ durch AIP
- 3. Stufe: Softwareentwicklung (SWE) mit MN weiter Implementation durch AIP

## 1.3 Mittel und Wege

### Einsetzbare Ressourcen personell:

SWE – AIP:

Es werden generell, also für Konzeption, Entwicklung, Tests und Dokumentation 2 SW-Ingenieure für die Konzeption, Programmierung, Doku sowie ein Testingenieur sowohl für Stufen 1b., als auch 2a. und 2b. zur Verfügung gestellt.

Vorgesehene Zeitaufwendungen:

Es wird AIP-seitens die Kapazität für die Stufen 2a. von ca. 1x1+1x2, für 2b. von ca 1x4 und für 3. Von ca. 2x3 SWE Mannwochen und insgesamt 1x3 Test-Mannwochen angesetzt.

### Konzeptmitarbeit:

- HORIBA: Basiskonzept und Zielsetzung
- Leineweber: Anregungen und konstruktive Kritik
- andere: ???

---

<sup>1</sup>das Protokoll darf nicht von proprietären Komponenten abhängig sein, wie das z.B. bei OPC der Fall wäre

**IF Adaptionen:**

Grundsätzlich ist vorgesehen sämtliche Komponenten mittels der für den Rasperry-Pi erhältlichen Add-Ons abzudecken; dies wird angenommener Weise jedoch nicht für Stecker und MG-interne Verbindungen gelten (?)

Beteiligung -

- Lds. Amt Thüringen: Katalog MG-Funktionen, Bereitstellung MG-Doku und Test-Umgebung
- HORIBA: Bereitstellung MG-Doku, begleitende Tests, Abstimmung der LAN-Kopplung und HW-seitige Adaptionen (?)

**SWE-Geräte und Infrastruktur:**

Wesentlich wird diese auf LINUX-basierten Servern von AIP betrieben und getestet; die Implementierung erfolgt in der Programmiersprache Java.

Für die Tests zu UDAP-IFC stellt –

- Lds. Amt Thüringen ein (1) SHARP 5030 über die erforderliche Dauer (ca 2-3 Wochen) zur Verfügung
- Fa. HORIBA ein (1) MG Serie 370 über die erforderliche Dauer (ca 4-6 Wochen) zur Verfügung.

## 2. Grundsatzfragen

### 2.1 Ausgangslage

Zum derzeitigen Zeitpunkt ist in existierenden Messnetzen von 3 unterschiedlichen Kommunikationsebenen auszugehen –

- |                               |   |
|-------------------------------|---|
| (1) MessNetzZentrale (MNZ)    | – MessStationsRechner (MSR)                         |
| (2) MessStationsRechner (MSR) | – MessGerät (MG)                                    |
| (3) ArbeitsPlatzRechner (APR) | – Sondergerät (z.B. Kalibrator) oder MessGerät (MG) |

Alle hier vorgenannten Geräte und Systeme sowie meteorologische und andere Sensoren, Staubsammler, Multikomponentenanalysatoren (wie PGC, MS,...) werden in diesem Dokument als MessNetzElemente (kurz **MNE**) bezeichnet.

Damit Hand in Hand gehen die unterschiedlichsten Grafischen User Interfaces (GUI) oder auch Anwendungsschirme, im besten Falle werden 2 von 3 mit einem einheitlichen GUI abgedeckt.

### 2.2 Abgrenzung

Mit der angestrebten Projektlösung ist keine Beschreibung oder Darstellung einzelner Messnetze (MN) mit ihren MNZ, MSR u.a.m. vorgesehen, sondern soll ausschließlich die Zugänglichkeit und damit Übertragung der erfassten und u.U. aufbereiteten Daten über die Luft- und Wassergüte allen, innerhalb und ausserhalb eines MN, agierenden MN-Elementen ermöglichen.

Mit der angestrebten Projektlösung sind damit weiters keine Spezifikation oder Beschreibungen verbunden, welche die Art und Weise der Anbindung einzelner Messnetzelemente wie z.B. MNZ, MSR, Modems u.a. vorsehen oder welche und wie fachlichen Zusatzaggregate wie Probenahmesysteme, Kalibratoren und deren Kalibrierarten u.ä. eingesetzt werden.

In den weiteren Kapiteln finden sich Kästen mit der Beschriftung „Abgrenzung“, die zusätzliche Informationen zu Themen enthalten, die nicht Teil der UIDEP-Spezifikation sind, sondern auf einer anderen Ebene behandelt werden müssen.

### 2.3 Wirtschaftliche Vorbedingungen

Im Wissen um die, defacto immer historisch gewachsene, Infrastruktur eines Luftgütemessnetzes hat eine Erneuerung auf eine optimale Umstellung unter verlängerter Nutzung des Altbestandes unbedingt Rücksicht zu nehmen.

Mit wesentlicher Unterstützung der die Investitionskosten stark beeinflussenden Messgerätehersteller ist nicht zu rechnen, da mit großen Absatzmengen wie in der Aufbauphase in Zukunft nicht mehr gerechnet werden kann. Wesentliche Belastungen seitens der Messnetzbetreiber werden auch nur bei wesentlichen Verbesserungen für diese, möglich bzw. erwartbar sein.

Daher sind bestehende Messnetzkomponenten mit ihrem, wenn auch dateninhaltsmäßig eingeschränkten, Datenlieferungsumfang, mittels Interfaceeinheiten so einbindbar zu machen, dass die Kosten in der gleiche Größenordnung wie derzeitig übliche Protokollinterfaces zu liegen kommen. Dies ist für Messgeräte und Kalibratoren genauso anzusetzen, wie für Zentralserver, Anzeigeeinheiten, Messstationsrechner und dergl. mehr.

Mithilfe einer Pilotinstallation, bei welchem Messnetzbetreiber auch immer, wird zu zeigen sein, dass sowohl ein stufen- / stationsweises Umstellen, als auch ein Parallelbetrieb möglich sein wird.

Damit sollten die notwendigen Investitionen problemlos auf mehrere Budgetjahre aufgeteilt werden können, ohne Einbußen im täglichen Betrieb befürchten zu müssen.

## 2.4 Gerätetechnische Vorbedingungen

Als Träger eines universellen Kommunikationssystems sind grundsätzlich zwei alternative Ausformungen angedacht –

- die dem beteiligten Gerät eingebaute Steuer- und Kontrolleinheit (CPU / Micro-Controller o.ä.)
- ein externer Micro-Controller als Protokollquelle, -senke oder -wandler.

Im vorliegenden Falle beschäftigen wir uns ausschließlich mit zweiterer Variante – dem externen Micro-Controller, hier als Plattform Raspberry-Pi 2 Modell B mit folgender Basispezifikation:

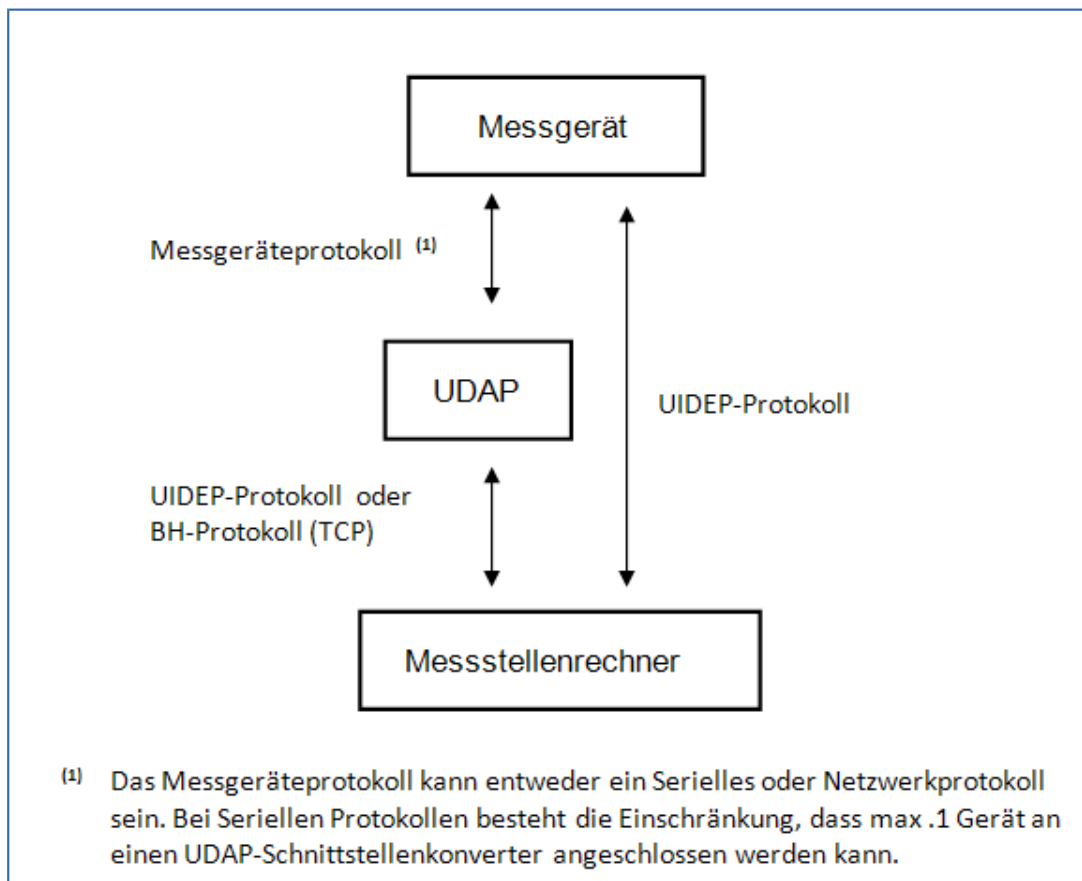
ARM Cortex-A7 Quad-Core-Prozessor mit VideoCore IV Dual-Core-GPU  
 1 GB SDRAM, HD 1080p-Videoausgang, Composite-Video-Ausgang (PAL/NTSC)  
 Ethernet-Anschluss 10/100 BaseT RJ45 + Video/Audiobuchse HDMI 1.3 und 1.4  
 4-polige Audio/Composite-Video-OUT-Buchse, 3,5 mm  
 4 USB 2.0-Buchsen  
 15-poliger MPI CSI-2-Steckverbinder für die Raspberry Pi HD-Videokamera  
 15-poliger serieller Display-Schnittstellensteckverbinder  
 + MicroSD-Kartensteckplatz  
 Stromversorgung extern: +5 V @ 2 A über micro-USB-Buchse  
 Abmessungen: 86 x 56 x 20 mm

Aufgrund des geringen Gewichts, seiner Abmessungen ist der direkte Einbau in Geräte angedacht, wobei die externe Stromversorgung u.U. auch aus dem Hostgerät genommen werden kann.

Da der rein batteriegestützte Betrieb als Sonderfall gelten kann, wurde der Fokus bei der Plattformauswahl auf Systemparameter wie Performance, Skalierbarkeit, standardisierte HW-Zusätze ( Kamera, HDMI, Kbd/CRT, ser.IF+clock ) und Auf-/ Abwärtskompatibilität zu unserer bestimmten Serverfamilien und Entwicklungssystemen gelegt.

## 2.5 Funktionstechnische Vorbedingungen

Sämtliche Funktionen des B-H-P finden sich in diesem Lösungsansatz wieder und sind als Alternative zum UIDEP-Protokoll definiert, damit man den UDAP-Protokollkonverter bereits an bestehende Systeme wie auch den IOX anbinden kann.



Der Datenaustausch zwischen zwei oder mehr Messnetzzentralen ist derzeit kein geplanter Anwendungsfall dieses Protokolls (ausgenommen es handelt sich um die Übertragung lediglich von Messwerten mit Attributen).

Bei der Integration dieses neuen Datenaustauschprotokolls wird zu berücksichtigen sein, dass in lokalen Netzen die Berücksichtigung von Sicherheits- (IT-Security-) Maßnahmen der Betreiber, wie die Unterstützung von Firewall-Rechnern zur Trennung eines Firmen/Behörden-Netzes vom öffentlichen Netz gewahrt bleibt. Daher sind diesbezügliche Methoden und Verfahren kein Bestandteil der vorliegenden Protokolldefinitionen.

Ebenso dient das Protokoll nicht dazu, messortspezifische Daten wie z.B. Stationsdaten zu Geographie und Administration, EU-Bezug u.dergl. austauschen zu können.

## **2.6 Informationsverteilung**

Der bidirektionale, rein textuell ausgerichtete Datenfluß mit einem strengen „request-answer“ Reglement des Bayern-Hessen-Protokolls (B-H-P) wird stattdessen zu einem multi-direktionalen, Text-/Bildobjekt orientierten Informationsaustausch mit aufgelöstem „request-answer“ Reglement und ungebundener Inhaltsübertragungsabfolge, erweitert. Als Text-Bildobjekte sind PDF-Dateien ebenso wie Grafikdateien oder simple-text Dateien vorgesehen.

### 3. Protokolltechnische Vorgaben

Mit diesem Protokoll werden sowohl Daten als auch Metadaten zwischen den einzelnen MN-Elementen ausgetauscht.

Um eine gute Lesbarkeit insbesondere zu Diagnosezwecken zu erreichen, werden Deskriptoren zu den Metadaten wie z.B. „Einheit“, „Stationsname“, u.ä. nur in Klartext angegeben.

Um eine hohe Flexibilität im Umgang mit oft unbekanntem Informationsgehalt zu erreichen, werden keine Vorgaben für die Reihenfolge und das Vorhandensein der den jeweiligen Sender und Empfänger beschreibenden Metadaten gemacht. Ein Nichtvorhandensein bedeutet im übertragenen Sinne „wir sind diesbezüglich in der gleichen Messwerterfassungsumgebung oder einander bekannt“, also der Empfänger erwartet keine anderen Daten als die gerade zu empfangenden.

Für den Datentransfer wird das Protokoll http (optional auch https) verwendet und nach den Prinzipien von „REST“ ( Representational State Transfer ) gestaltet. Dies hat den Vorteil, dass auf eine Vielzahl bereits existierender Hard- und Softwarekomponenten zurückgegriffen werden kann, die dieses Protokoll implementieren (ein Test ist z.B. mit einem WEB-Browser möglich).

Inhaltlich wird zwischen Messwerten = Messdaten und Metadaten unterschieden, welche zwei grundlegende Funktionalitäten bereitzustellen haben:

- Funktionen zur **Abfrage von Messdaten** mit / ohne Vorgabe von Attributen, **Zustandsdaten** oder **Metadaten** bei Bedarf
- Funktionen zur **eigenständigen Lieferung von Daten**, aufgrund des Eintreffens bestimmter Ereignisse (z.B. bei Einlangen neuer Messwerte, Änderung von Metadaten)
- Funktionen zur **Steuerung des Setzens von gerätespezifischen Funktionen** des angesprochenen MN-Elementes

#### **Abfrage von Mess- Zustands- oder Metadaten**

Es wird unterschieden werden zwischen –

- Identität und Status des Gegenüber-MN Elements
- laufender Daten
- zeitbereichsbasierter (archivierter) Daten

#### **Abgrenzung: Eigenständige Lieferung von Daten**

Es wäre denkbar, dass ein Empfänger von Daten beim Lieferanten der Daten die Lieferung abonniert; der Lieferant würde dann selbständig immer wieder die abonnierten Daten zum Empfänger schicken, sobald diese verfügbar sind. Aktuell ist ein solcher Mechanismus aber nicht im UIDEP-Protokoll vorgesehen.

#### **Steuerung zur Datenabfrage**

Es wird unterschieden werden zwischen –

„simple output“ = lediglich die primären Messdaten

„complex output“ = alle primären Messdaten plus alle dazu verfügbaren Attribute und Zustandsinformationen

#### **Festlegung von einheitlichen Funktionen**

Zunächst sollen vom Messgerät bestimmte Abfragen unterstützt werden, unabhängig vom Messgerätetyp. Durch eine eigene Abfragefunktion sollen Einstellungen und verfügbare bzw unterstützte Antwortfunktionen des Messgeräts ausgelesen werden können.

Die Übertragung wird mittels Diensten realisiert siehe Pkt 4.4

**Steuerung des Setzens von Zuständen**

Mittels Dateiformaten in JSON (JavaScript Object Notation), welche einer programmatischen Strukturprüfung zur Laufzeit, bei gleichzeitig geringerem Datenvolumen, unterliegen.

Ergänzend ist die Möglichkeit des Parsens für die schnelle Diagnose vorgesehen.



## 4. PROTOKOLLSTRUKTUR

Die praktische Situation und damit sowohl Topologie, als auch Workflow eines Messnetzbetriebes stellen die Grundlage der Überlegungen zur optimalen Abbildung jeglichen Informationsaustausches dar. Es wird daher lediglich zwischen einer Sender- und Empfänger-Ebene unterschieden, Bereiche wie Validierungen, Zwischenspeicherungen u.ä. werden nicht berücksichtigt.

>>> Der Zustand der Messdaten (Messwerte) und ihrer zugehörigen Metadaten ist mangels international gegebener Validierungsnormen ausschließlich in der Verantwortung des Senders.

### 4.1 PROTOKOLLAUFBAU

Die Struktur der übertragenen Daten ist in mehreren Hierarchieebenen aufgebaut; abhängig von Datenquelle und -Ziel werden die nicht benötigten Hierarchieebenen weggelassen.

Die Hierarchieebenen laufen:

- Organisation (Messnetzbetreiber)
- Station
- Messgerät
- Sensor/Komponente/Kanal
- Wert

Beispiel für JSON-Daten<sup>2</sup>, die ein MSR von einem Messgerät erhält:

```
{
  "Device": "Horiba APNA/370",
  "SN": "12345678",
  "Components": [
    {
      "ID": "178",
      "Value": 6.543
    },
    {
      "ID": "179",
      "Value": 2.345,
      "ErrSts": ["F"]
    }
  ]
}
```

---

<sup>2</sup>Leerzeichen und Zeilenumbrüche sind im JSON-Format optional, hier zwecks Lesbarkeit eingefügt

Wenn der MSR später die Daten an die Zentrale schickt, kommen einerseits weitere Hierarchieebenen hinzu, andererseits werden die bekannten Ebenen mit zusätzlichen Feldern ergänzt:

```
{
  "Station": "AIP-Teststation",
  "Time": "2015-05-19T11:30:00+01:00",
  "AvgTime": 1800,
  "Devices": [
    {
      "Device": "Horiba APNA/370",
      "SN": "12345678",
      "Components": [
        {
          "Component": "NO2",
          "ID": "178",
          "Unit": "ppb",
          "Value": 6.543,
          "MinValue": 0.112,
          "MaxValue": 11.778,
          "StdDev": 1.44,
          "Valid": true
        },
        {
          "Component": "NO",
          "ID": "179",
          "Unit": "ppb",
          "Value": 2.345,
          "MinValue": 0.012,
          "MaxValue": 4.778,
          "StdDev": 0.74,
          "Valid": true
        }
      ]
    }
  ],
  {
    "Device": "Horiba APOA/370",
    "SN": "12345679",
    "Components": [
      {
        "Component": "O3",
        "ID": "211",
        "Unit": "ppb",
        "Value": 16.543,
        "MinValue": 3.112,
        "MaxValue": 31.778,
        "StdDev": 2.44,
        "OpSts": ["N"],
        "Valid": true
      }
    ]
  }
]
```

Folgende Regeln gelten:

- Jede Quelle liefert nur die Felder, deren korrekten Inhalt sie kennt; alle anderen Felder werden ganz weggelassen. z.B. liefert ein Messgerät ohne eingebaute Uhr keine Information über den Messzeitpunkt.
- Jedes Feld ist logisch genau einer Hierarchie-Ebene zugeordnet
- Felder von Objekten unterer Hierarchie-Ebenen können in der Darstellung in höhere Hierarchie-Ebenen hinaufgezogen werden, sie gelten dann für die zugehörigen Objekte der untergeordneten Hierarchie-Ebenen. Beispielsweise ist der Messzeitpunkt auf der Ebene „Wert“ angesiedelt, kann aber auch in einer höheren Ebene, z.B. „Station“ angegeben werden; das hat zur Folge, dass für alle Werte unterhalb der Station dieser Messzeitpunkt gilt
- Wenn es in einer Hierarchie-Ebene nur genau ein Objekt gibt (z.B. nur eine Komponente in einem Gerät), so kann dieses in der JSON-Darstellung mit dem darüberliegenden Objekt verschmelzen (Verflachung)
- Die Reihenfolge der Felder innerhalb eines Objekts spielt keine Rolle, es ist keine bestimmte Reihenfolge vorgeschrieben und jede mögliche Reihenfolge führt zum gleichen Resultat

## 4.2 MESSNETZ-MESSDATEN

### 4.2.1 Momentanwerte\*

### 4.2.2 Kurzzeitmittelwerte\*

### 4.2.3 Halbstundenmittelwerte\*

### 4.2.4 Sondermittelwerte

#### 4.2.4.1 Windrichtung sektoriell (Häufigkeitsauszählung)

#### 4.2.4.2 Windrichtung vektoriell gemittelt

#### 4.2.4.3 Windspitze

### 4.2.5 Zusatzdaten zu Mittelwerten QS

#### 4.2.5.1 Minimum der Einzelwerte, die in den Mittelwert eingegangen sind

#### 4.2.5.2 Maximum der Einzelwerte, die in den Mittelwert eingegangen sind

#### 4.2.5.3 Standardabweichung der Einzelwerte, die in den Mittelwert eingegangen sind

#### 4.2.5.4 Perzentile / Andere ???

### 4.2.6 Betriebs- und Fehlerstati

### 4.2.7 Kalibrierwerte

\* ... Unterscheidung kann wegfallen, wenn Mittelungsintervall Bestandteil des Konfigurationsblockes ist.

## 4.3 MESSNETZ-METADATEN

Darunter verstehen sich alle Informationen, die die Struktur und Inhalte eines Meßnetzes beschreiben wie z.B. Stations-, Messgeräte- und Komponentenbeschreibungen

### 4.3.1 Zentrale

Name, IP-Adresse(n), DB-Spezifika (?)

### 4.3.2 Messstation

Name, Koordinaten (mit Systemkennung – Gauß-Krüger(GK) / geo.Lg.+geo.Br...), MSR-Typ, Seriennummer, MSR-SW-Version, MSR-IP-Adresse(n), Bestückung an Messgeräten

### 4.3.3 Messgeräte / Kalibratoren

ModellNr., Ser.Nr., SW-VersionsNr., IP-Adresse (ggf. des Protokollkonverters), Erfassungstyp (?)

### 4.3.4 Sensor / Komponente / Kanal

Kennung, Messbereich(e)

### 4.3.5 Übertragungswerte (auch Kalibrierwerte)

Abfrage-/Lieferzeitraum, Intervall, Einheit, Validierungs-/Kontrollstufe bzw. gültigkeit (?-eigene Konvention)

#### 4.4 DATENTRANSFER-ÜBERTRAGUNGSSPEZIFIKA

Die von den Geräten angebotenen Webdienste, mit deren Hilfe Daten nach UIDEP ausgetauscht werden können, sind REST-ful (REST = Representational State Transfer) konzipiert.

Das bedeutet:

- Die Dienste sind zustandslos, das Service muss sich keine Kommunikationsdaten über einzelne Anfragen hinweg merken
- Die angebotenen Dienste stehen in einer verzeichnisartigen Struktur zur Verfügung
- http-Requests GET, POST, PUT, DELETE werden verwendet, um Daten abzufragen, anzulegen, abzuändern bzw. zu löschen
- Vermittler irgendwo im Netzwerk (firewall, gateways, u.ä.) haben keinen die Schnittstelle verändernden Einfluß

##### **Abgrenzung: Authentifizierung und Rechtevergabe**

Diese sind nicht Bestandteil der Protokollspezifikation; wenn nötig, kann dies über HTTP-BASIC-Authentifizierung abgewickelt werden (dadurch werden Benutzername und Passwort übergeben). Für eine wirksame Verschlüsselung der Kommunikation kann UIDEP auch über HTTPS übertragen werden; Client- und Server-Zertifikate können dabei verwendet werden, um die Kommunikationspartner sicher zu authentifizieren. Weiters können Firewalls eingerichtet und so konfiguriert werden, dass die UIDEP-fähigen Geräten nur von befugten Gegenstellen erreichbar sind; das betrifft auch z.B. den Fall eines externen Technikers, der von einem abgeschotteten Gast-Netzwerk aus über die Firewall so freigeschaltet wird, dass er genau die zu wartenden Geräte erreicht und sonst niemanden. Die UIDEP-Spezifikation trifft keine Aussage darüber, welche dieser Massnahmen eingerichtet wird. Ebensovienig soll hier festgelegt werden, welche Rollen es gibt, wie die Benutzer zu diesen Rollen zugeordnet werden und wie die Rechte pro Rolle bzw. pro Einzelbenutzer verwaltet werden.

##### 4.4.1 Von einem Messgerät bereitgestellte Webdienste (URL) für die Abfrage von Messwerten

Erlaubte http-Methode: GET

Primäre Messdaten: /values/simple

Messdaten mit allen Zusatzinfos: /values/complex

Ohne weitere Parameter werden die Daten aller Komponenten des Messgeräts für die zuletzt durchgeführte Messung geliefert.

Optionale Parameter bei der Abfrage:

?component= <i>id</i>	Einschränkung auf Messdaten einer Komponente
?start= <i>JJJJ-MM-TT-hh-mm-ss</i>	historische Daten ab dem Zeitpunkt liefern (inkl.)
?end= <i>JJJJ-MM-TT-hh-mm-ss</i>	historische Daten bis zum Zeitpunkt liefern (inkl.)
?avgtime= <i>n</i>	Messdaten mit angegebenem Mittelungsintervall liefern
?valid=true	Nur gültige Messdaten anfragen

Beispiel:

```
{
  "Device": "Horiba APNA/370",
  "SN": "12345678",
  "Components": [
    {
      "ID": "178",
      "Value": 6.543
    }
  ]
}
```

#### 4.4.2 Von einem Messgerät bereitgestellte Webdienste (URL) für die Steuerung (z.B. Einleiten einer Funktionskontrolle)

Erlaubte http-Methoden: GET (Betriebsmodus abfragen), PUT (Betriebsmodus ändern)  
Betriebsmodus lesen/schreiben: /mode

Beispiel:

```
{  
  "OperatingMode": "M"  
}
```

#### 4.4.3 Von einem Messgerät bereitgestellte Webdienste (URL) für die Konfigurierung

Erlaubte http-Methoden: GET (Konfiguration abfragen), PUT (Konfiguration ändern)  
Konfiguration lesen/schreiben: /configuration

Beispiel:

```
{  
  "DevConfiguration": [  
    {  
      "SettingName": "foo",  
      "DataType": "string",  
      "MaxLength": 10,  
      "CurrentSetting": "bar"  
    },  
    {  
      "SettingName": "xyz",  
      "DataType": "integer",  
      "MinValue": 0,  
      "MaxValue": 100,  
      "CurrentSetting": 50  
    }  
  ]  
}
```

#### 4.4.4 Von einem MSR bereitgestellte Webdienste (URL ) für die Abfrage von Messwerten

Erlaubte http-Methoden: GET

Primäre Messdaten von allen Geräten: /values/simple

Messdaten mit allen Zusatzinfos: /values/complex

Kalibrierwerte: /functioncheck

Ohne weitere Parameter werden die Daten aller angeschlossener Messgeräte für die zuletzt durchgeführte Messung mit dem primären Mittelungsintervall (i.d.R. HMW) geliefert.

Optionale Parameter bei der Abfrage:

?device=seriennummer	Einschränkung auf Messdaten eines Geräts
?start=JJJJ-MM-TT-hh-mm-ss	Messdaten ab dem angegebenen Zeitpunkt liefern (inkl.)
?end=JJJJ-MM-TT-hh-mm-ss	Messdaten bis zum angegebenen Zeitpunkt liefern (inkl.)
?avgtime=n	Messdaten mit angegebenem Mittelungsintervall liefern

Beispiel:

```
{
  "Station": "AIP-Teststation",
  "Time": "2015-05-19T11:30:00.000+01:00",
  "AvgTime": 1800,
  "AmsType": "GEMI IOX",
  "Version": "3.02",
  "Devices": [
    {
      "Device": "Horiba APNA/370",
      "SN": "12345678",
      "Components": [
        {
          "Component": "NO2",
          "ID": "178",
          "Unit": "ppb",
          "Value": 6.543,
          "MinValue": 0.112,
          "MaxValue": 11.778,
          "StdDev": 1.44,
          "Valid": true,
          "ChkTime": "2015-05-19T11:30:00.000+01:00",
          "Refs": [
            {
              "RefNo": 0,
              "NominalValue": 0.0,
              "CheckValue": 0.2
            },
            {
              "RefNo": 1,
              "NominalValue": 120.0,
              "CheckValue": 118.85
            },
            {
              "RefNo": 2,
              "NominalValue": 200.0,
              "CheckValue": 201.7
            }
          ]
        }
      ]
    }
  ]
}
```

#### 4.4.5 Von einem Messstellenrechner bereitgestellte Webdienste (URL) für die Konfigurierung

Erlaubte http-Methoden: GET (Konfiguration abfragen), PUT (Konfiguration ändern)  
Konfiguration lesen/schreiben: /configuration

##### Beispiel:

```
{
  "Station": "AIP-Teststation",
  "StationConfiguration": [
    {
      "SettingName": "foo",
      "DataType": "string",
      "MaxLength": 10,
      "CurrentSetting": "bar"
    },
    {
      "SettingName": "xyz",
      "DataType": "integer",
      "MinValue": 0,
      "MaxValue": 100,
      "CurrentSetting": 50
    }
  ],
  "Devices": [
    {
      "Device": "Horiba APNA/370",
      "SN": "12345678",
      "DevConfiguration": [
        {
          "SettingName": "bar",
          "DataType": "string",
          "MaxLength": 10,
          "CurrentSetting": "foo"
        },
        {
          "SettingName": "abc",
          "DataType": "integer",
          "MinValue": 0,
          "MaxValue": 60,
          "CurrentSetting": 20
        }
      ]
    }
  ]
}
```

#### 4.4.6 Von einem Messstellenrechner bereitgestellte Webdienste (URL) für die Übertragung von Dokumenten

Erlaubte http-Methoden: GET (Liste abfragen, Dokument herunterladen), POST (Dokument hochladen)

Liste der Dokumente: /documents

Dokument herunterladen/hochladen: /documents/*dateiname*

Beispiel:

```
{
  "Documents": [
    "somedocument.pdf",
    "anotherdocument.zip"
  ]
}
```

#### 4.4.7 Von der Messnetzzentrale bereitgestellte Webdienste (URL) für Eigenmeldungen

Erlaubte http-Methode: POST

Eigenmeldung schicken: /eventnotification

```
{
  "Station": "AIP-Teststation",
  "Time": "2015-05-19T11:37:15.000+01:00",
  "EventType": "Invalid function check",
  "EventText": "Span gas bottle empty",
  "Device": "Horiba APNA/370",
  "SN": "12345678",
  "Components": [
    {
      "ID": "178"
    }
  ]
}
```

Folgende Ereignistypen ("EventType") sind definiert:

- "Operating Status" → Übermittlung von Betriebs- und/oder Fehlerstati
- "Limit exceedance" → Meldung aufgrund einer Grenzwertüberschreitung
- "Device change" → Meldung aufgrund eines Messgerätetausches
- "Invalid function check" → Meldung wegen einer fehlgeschlagenen Funktionskontrolle
- "Restart" → Meldung wegen eines Neustarts des Geräts (MG oder MSR)
- "System" → sonstige vom System ausgelöste Meldungen
- "Manual" → Vom Benutzer vor Ort ausgelöste Meldung
- "Document transfer" → Es sind Dokumente von der Station in die Zentrale zu übertragen



#### 4.4.8 Abfrage der Fähigkeiten eines Geräts (*Neu in 1.5: OperatingModes, OperatingStatuses, ErrorStatuses*)

Erlaubte http-Methoden: GET

Liste der Fähigkeiten: /capabilities

Beispiel:

```
{
  "DevCapabilities": {
    "Descriptors": [
      "Active", "AvgTime", "DevURL", "DevCapabilities",
      "CheckValue", "Component", "Components", "CurrentSetting",
      "DataType", "DevConfiguration", "Device", "ErrSts",
      "EventText", "EventType", "FunctionChecks", "HistoricalData",
      "ID", "IntSts", "LastCalibration", "MaxSetting", "MeasuredValues",
      "MinSetting", "NominalValue", "NotificationURL",
      "OperatingMode", "OpSts", "RefNo", "Refs", "Services",
      "SN", "StoragePeriod", "Time", "Unit", "Valid", "Value",
      "Version"
    ],
    "Services": [
      "values/simple", "values/complex", "configuration",
      "capabilities"
    ]
  },
  "DevConfiguration": [
    {
      "SettingName": "SomeSetting",
      "DataType": "integer",
      "MinSetting": 0,
      "MaxSetting": 100,
      "CurrentSetting": 42
    },
    {
      "SettingName": "AnotherSetting",
      "DataType": "string",
      "MaxLength": 50,
      "CurrentSetting": "FooBar"
    }
  ],
  "LastCalibration": "2015-05-19T11:37:15.000+01:00",
  "Features": {
    "TimeSynchronized": true,
    "HistoricalData": true,
    "Aggregations": [
      {
        "AvgTime": 5,
        "StoragePeriod": 86400
      },
      {
        "AvgTime": 300,
        "StoragePeriod": 864000
      },
      {
        "AvgTime": 1800,
        "StoragePeriod": 86400000
      }
    ]
  },
  "OperatingModes": [
    {
      "OperatingMode": "M",
      "Description": "Measuring"
    }
  ]
}
```

```
    },
    {
      "OperatingMode": "Z",
      "Description": "Zero Function Check"
    },
    {
      "OperatingMode": "S",
      "Description": "Span Function Check"
    },
    {
      "OperatingMode": "P",
      "Description": "Power saving mode (Standby)"
    }
  ],
  "OperatingStatuses": [
    {
      "Status": "Z",
      "Description": "Zero function check active"
    },
    {
      "Status": "S",
      "Description": "Span function check active"
    },
    {
      "Status": "U",
      "Description": "Device in startup procedure"
    },
    {
      "Status": "D",
      "Description": "Device in shutdown procedure"
    }
  ]
  "ErrorStatuses": [
    {
      "Status": "T",
      "Description": "Operating temperature not in acceptable range"
    },
    {
      "Status": "R",
      "Description": "No response from sensor unit"
    }
  ]
}
```

#### 4.4.9 Abfrage der Zeit eines Geräts

Erlaubte http-Methoden: GET

Liste der Fähigkeiten: /currenttime

Beispiel:

```
{
  "CurrentTime": "2015-11-18T10:22:05.000+01:00",
  "TimeSynchronized": true
}
```

Mit dieser Methode kann man überprüfen, ob die eigene Uhr mit der des Kommunikationspartners ausreichend übereinstimmt.

Als Reaktion auf ermittelte Differenzen kann man die von diesem Gerät gelieferten Messwerte

- so übernehmen, wie sie kommen (allfällige Zeitdifferenz wird ignoriert)
- zeitversetzt übernehmen (die Zeitdifferenz wird kompensiert)
- verwerfen (Messwerte werden wegen der Zeitdifferenz als unbrauchbar angesehen)

Dies betrifft nur den Fall, dass man sich historische Daten liefern lässt. Bei der Abfrage der aktuellen Messwerte kann man die Zeitangabe des Kommunikationspartners komplett ignorieren und die eigene Uhr als Referenzzeitmesser verwenden.

#### 4.4.10 Liste der Deskriptoren (ausgenommen Konfigurations-Deskriptoren)

Deskriptor	Typ	Beschreibung	Ebene
Active	Boolean	Aktiv-Kennzeichen	Gerät
AmsType	String	MSR-Typ	Station
AvgTime	Integer	Mittelungsintervall in Sekunden	Wert
ChkTime	String (ISO-Zeit)	Zeitpunkt einer Funktionskontrolle	FK
CheckValue	Gleitkomma	Istwert der Funktionskontrolle**	FK-Ref
Component	String	Komponente (Parameter) - Bezeichnung	Komp.
Components	Array	Liste der Komponenten	Gerät
Configuration	Object	Konfigurationsdaten (gerätespezifisch)	Org., Station, Gerät, Komp.
CurrentSetting	variabel	Aktueller Wert einer Konfigurationseinstellung	Konfig.
CurrentTime	String (ISO-Zeit)	Aktuelle Zeit laut interner Uhr des Geräts	Gerät
DataType	String	Datentyp einer Konfigurationseinstellung	Konfig.
Description	String	Beschreibung eines Status oder Modus	Modus, Status
Descriptors	Array	Liste der vom Gerät unterstützten Standard-Deskriptoren	Gerät
Device	String	Gerätetyp	Gerät
Devices	Array	Liste der Daten pro Gerät	Station
DevURL	String	URL, unter der die UIDEP-Dienste erreicht werden können	Gerät
Documents	Array<String>	Liste der verfügbaren Dokumente	Station

ErrorStatuses	Array	Liste der möglichen Fehlerstati	Gerät
ErrSts	Array<String>	Liste der angefallenen Fehlerstati	Wert
EventText	String	Ereignistext einer Eigenmeldung	EM
EventType	String	Typ einer Eigenmeldung	EM
Features	Object	Angabe der unterstützten Features	Gerät
FunctionChecks	Array	Liste der Funktionskontrollen (Kalibrierungen)	Komp.
ID	String	Komponenten-ID	Komp.
IntSts	Array<String>	Liste der angefallenen internen Stati	Wert
LastCalibration	String (ISO-Zeit)	Datum der letzten „richtigen“ Kalibrierung	Gerät
MaxLength	Integer	Maximale Länge einer Konfig-Einstellung vom Typ String	Konf.
MaxSetting	Gleitkomma	Maximal erlaubter Wert einer Einstellung	Konf.
MaxValue	variabel	Größter Messwert im Mittelungsintervall	Wert
MeasuredValues	Array	Liste der Messwerte	Komp.
MinSetting	Gleitkomma	Minimal erlaubter Wert einer Einstellung	Konf.
MinValue	variabel	Kleinster Messwert im Mittelungsintervall	Konf.
NominalValue	Gleitkomma	Sollwert der Funktionskontrolle**	FK-Ref
NotificationURL	String	URL der Zentrale (für Eigenmeldungen)	Org.
OperatingMode	String	Betriebsmodus („M“=Messen, „0“=Nullen, „1“=Span, ...)	Gerät
OperatingModes	Array	Liste der möglichen Betriebsmodi	Gerät
OperatingStatuses	Array	Liste der möglichen Betriebsstati	Gerät
OpSts	Array<String>	Liste der angefallenen Betriebsstati	Wert
Organisation	String	Name der Organisation	Org.
RefNo	Integer	Nummer des Referenzpunktes (0=Zero, 1=Span, ...)**	FK-Ref
Refs	Array	Liste der Referenzpunkte einer Funktionskontrolle	FK
Services	Array	Liste der unterstützten Dienste (HTTP-Aufrufe)	Gerät, Station
SN	String	Seriennummer	Gerät
Station	String	Stationsname (Schlüsselbegriff)	Station
StationURL	String	URL, unter der die UIDEP-Dienste erreicht werden können	Gerät
Status	String	Statuscode für Selbstbeschreibung	Status
StdDev	Gleitkomma	Standardabweichung im Mittelungsintervall	Wert
StoragePeriod	Integer	Zeitraum, in dem das Gerät zurückliegende Messwerte speichert, in Sekunden	Aggr.
Time	String (ISO-Zeit)	Zeitpunkt	Wert
TimeSynchronized	Boolean	True, wenn das Gerät über eine mit offizieller Zeit synchronisierte Echtzeituhr enthält (per NTP, Funkuhr o.ä.)	Gerät

Unit	String	Einheit	Komp.
Valid	Boolean	Gültigkeitskennzeichen	Wert
Value	Gleitkomma	Messwert	Wert
Version	String	Software-Version	Station

\*\* ... bei mehreren FK-Vorgaben bzw in Messungen als „lack of fit“-Test bestimmt RefNo welcher Messpunkt betroffen ist

Hinweis: neue Descriptoren der Protokollversion 1.5 wurden in obiger Liste **gelb** hinterlegt

## 5. Geräteerkennungsprotokoll

Das Geräteerkennungsprotokoll dient dazu, im lokalen Netz befindliche Messgeräte, welche UIDEP unterstützen, automatisch erkennen und einbinden zu können.

Der Hauptzweck ist der, dass ein MSR automatisch die Messgeräte in der Station auffinden und konfigurieren kann. Es ist aber auch möglich, dass sich ein Messgeräte auf diese Weise mit einem anderen Messgerät verbindet, um von diesem Messwerte zur zB. Kompensation bestimmter, aufgrund von Querempfindlichkeit sich ergebender Fehlmessungen, zu erhalten.

### Abgrenzung: Auffinden von MSR durch die MNZ, IPv6

Für das unten beschriebene Verfahren ist es notwendig, dass sich Sender und Empfänger der UDP-Broadcast im gleichen Netzwerksegment befinden. Dies ist im Fall von MSR und MG regelmässig der Fall (stationsinternes Netz), aber normalerweise nicht im Fall von MNZ und MSR. Durch eine Umstellung auf IPv6 würde diese Limitierung obsolet werden; die dann einzusetzenden Multicasts (statt Broadcasts) könnten alle interessierten Geräte erreichen. Daher ist unten beschrieben, wie eine MSR-Antwort auf eine derartige Anfrage auszusehen hat.

Dies wird durch einen UDP-Broadcast auf Port 4120 realisiert, der folgenden Inhalt (ASCII) hat:

UIDEP-Geräte, die das Erkennungsprotokoll unterstützen, reagieren darauf ihrerseits mit einer UDP-Nachricht an den Absender des Broadcasts, ebenfalls Port 4120; in dieser Nachricht werden (UTF-8 codiert) im JSON-Format die wichtigsten Daten des Geräts bekannt gegeben.

Folgende Descriptoren können dabei zur Anwendung kommen:

Component

Ein MSR, der auf eine derartige Anfrage reagiert, kann darüber hinaus auch folgende Descriptoren liefern:

Station, StationURL, AmsType, Version, DeviceList

Beispiel für ein MG, das sich meldet:

```
{
  "Device": "Horiba APNA/370",
  "SN": "12345678",
  "DevURL": "http://192.168.11.15:8080/uidep",
  "Components": [
    {
      "ID": "178",
      "Component": "NO2"
    },
    {
      "ID": "179",
      "Component": "NO"
    }
  ]
}
```

Beispiel für einen MSR, der sich meldet:

```
{
  "Station": "AIP-Teststation",
  "AmsType": "GEMI IOX",
  "Version": "3.02",
  "StationURL": "http://192.168.11.2/foobar/uidep",
  "Devices": [
    {
      "Device": "Horiba APNA/370",
      "SN": "12345678",
      "DevURL": "http://192.168.11.15:8080/uidep",
      "Components": [
        {
          "ID": "178",
          "Component": "NO2"
        },
        {
          "ID": "179",
          "Component": "NO"
        }
      ]
    }
  ]
}
```

Die Antwort muss gültiges JSON sein; Leerzeichen und Zeilenumbrüche sind wie üblich im obigen Beispiel der Lesbarkeit wegen enthalten, aber nicht obligatorisch.

Bitte beachten: Port 4120 ist fix für diesen Erkennungsdienst festgelegt; im Gegensatz dazu können die in den vorigen Kapiteln beschriebenen Webdienste jeden beliebigen Port verwenden – es ist nur wichtig, dass dieser im **DevURL**-Feld der Antwort auf den Geräteerkennungsbroadcast angegeben wird. (Im obigen Beispiel wäre das Port 8080)

## 6. Abkürzungen und Acronyme

... in Arbeit / unvollständig

### 6.1 generell verwendete IT-

JSON	JavaScript Object Notation
URL	Uniform Resource Locator; hier: bereitgestellter Webdienst (bzw. dessen Adresse)

### 6.2 Fach-Begriffe

APR	ArbeitsPlatzRechner
BHP	Bayer-Hessen-Protokoll
EU-RL	Europäische Richtlinie
MG	Messgerät
MNZ	MessNetzZentrale
MSR	MessStationsRechner
NWG	NachWeisGrenze

### 6.3 Projektinterne Begriffe

UDAP	UBIS Detached Acquisition & Plausibility
UIDAP / UIDEP	Unverselles ImmissionsDaten Austausch Protokoll / Universal Immission Data Exchange Protocol

### 6.4 Herstellerbezogene Begriffe