

## Umsetzung der Auftragsprüfung in GeKo

PROJEKTARBEIT

2. FACHSEMESTER

als Prüfungsleistung für das Modul „Projekt I“

im Studiengang Informations- und Kommunikationstechnik  
der Fakultät Digitale Transformation  
der Hochschule für Technik, Wirtschaft und Kultur Leipzig

Vorgelegt von: **Niels Fabien Häußler**, Matrikel-Nr.: **79153**

geboren am: **13. August 2002**

in: **Ulm**

eingereicht am: **26.08.2022**

Erstprüfer: **Dipl.-Math. Benjamin Schmidt**  
Hochschule für Technik, Wirtschaft und Kultur  
Leipzig

Zweitprüfer: **Dipl.-Ing. Mario Heidrich**  
Deutsche Telekom Technik GmbH  
Ulm

# Inhaltsverzeichnis

1	Abbildungsverzeichnis.....	1
2	Abkürzungsverzeichnis.....	2
3	Einleitung.....	3
4	Abteilungsbeschreibung.....	4
5	Konzeptionelle Grundlagen.....	5
5.1	XML-Grundlagen.....	5
5.1.1	Elemente.....	5
5.1.2	Attribute.....	6
5.1.3	Kommentare.....	6
5.1.4	Prolog.....	6
5.2	XSD-Grundlagen.....	7
5.2.1	SimpleType.....	7
5.2.2	ComplexType.....	8
5.2.3	Einschränkungen.....	10
5.3	XPath.....	11
6	Umsetzung der Auftragsprüfung.....	13
6.1	Analyse der internen und externen XSD.....	13
6.2	Regeln und Validierungen in GeKo.....	15
6.2.1	Regel.....	15
6.2.2	Validierung.....	16
7	Abschluss und Blick in die Zukunft.....	18
8	Anhang.....	19
9	Literaturverzeichnis.....	28
10	Selbstständigkeitserklärung.....	30

# 1 Abbildungsverzeichnis

Abbildung 1-Interne(links)-vs Externe(rechts)-XSD (1.Fall); Quelle: Eigene Aufnahme.....	13
Abbildung 2-Interne(links)-vs Externe(rechts)-XSD (2.Fall); Quelle: Eigene Aufnahme.....	13
Abbildung 3-Interne(links)-vs Externe(rechts)-XSD (3.Fall); Quelle: Eigene Aufnahme.....	14
Abbildung 5: Anlegen einer neuen Rege; Quelle: Eigene Aufnahme .....	15
Abbildung 6: Anlegen einer neuen Validierung; Quelle: Eigene Aufnahme .....	17

## 2 Abkürzungsverzeichnis

KoSy-IP	Konfigurationssystem für IP-Produkte
GeKo	Generisches Konfigurationssystem
XML	Extensible Markup Language
XSD	XML Schema Definition
OM-IP	Order Management IP
T-CNF-IPC_CbC	Telekom Core Networks und First-Line Maintenance IP-Core Chapter Opearation IP-Backbone 2
T-CNF-IPC BBC CMC	Telekom Core Networks und First-Line Maintenance IP-Core Squad Changemanagement Carrier und GK-Connectivity
T-CNF-IPC NNA KIC	Telekom Core Networks und First-Line Maintenance IP-Core Squad Konfigurationsysteme IP Core
LER	Label Edge Router
CPE	Customer Premises Equipment
XPath	XML Path Language
W3C	World Wide Web Consortium
RegEx	Regular Expression
QoS	Quality of Service
TCT	Projekt Test Center Technik

### 3 Einleitung

Bei der Deutschen Telekom AG wird für Router-Konfigurationen das Bestandssystem KoSy-IP (Konfigurationssystem für IP-Produkte) verwendet. Dieses soll durch GeKo (Generisches Konfigurationssystem), welches sich noch in der Entwicklung befindet, abgelöst werden. Der Vorteil von GeKo ist die weitreichende Automation der Auftragsbearbeitung.

Eingehende XML-Aufträge (Extensible Markup Language) zu Konfigurationen müssen auf Richtigkeit geprüft werden, damit es im späteren Verlauf der Bearbeitung zu keinen Komplikationen kommt. Hierzu verwendet KoSy-IP eine externe und eine interne XSD (XML-Schema). Die externe XSD überprüft die grundlegenden und die interne XSD die detaillierteren Eigenschaften. So überprüft die externe-XSD zum Beispiel oftmals nur, ob ein Wert vorhanden ist. Die interne hingegen ob der Wert, zum Beispiel eine IP-Adresse, syntaktisch richtig aufgebaut ist. Damit im Falle eines Fehlers dieser an OM-IP (Order Management IP), welche verantwortlich für die Aufträge sind, oder den Sachbearbeitenden zurückgemeldet wird, wird die interne XSD in den Regel- und Validierungsmechanismus von GeKo implementiert. Damit kann die interne XSD entfernt werden, wodurch die Auftragsprüfung einfacher und übersichtlicher gestaltet wird.

Die folgende Projektarbeit befasst sich zunächst mit den Grundlagen von XML, XSD und XPath sowie mit der Übernahme und Implementierung der detaillierteren Auftragsprüfung und dem Regel- und Validierungsmechanismus von GeKo. Dabei wird eine Analyse und ein Vergleich der internen und externen XSD von KoSy-IP angefertigt. Durch diese Analyse können die notwendigen Regeln und Validierungen für eine sichere Auftragsprüfung und Bearbeitung in GeKo hinzugefügt werden. Abschließend wird dieser Mechanismus getestet und beurteilt.

## 4 Abteilungsbeschreibung

Das Chapter T-CNF-IPC\_CbC (Chapter Operation IP-Backbone 2), geleitet von Herrn Artur Schultz, besteht aus den beiden Squads T-CNF-IPC BBC CMC sowie T-CNF-IPC NNA KIC, welche insgesamt 26 Mitarbeiter besitzen (Stand: 11.08.2022).

Hierbei ist das Squad T-CNF-IPC CMC für die Konfiguration von IP-Produkten auf der IP-Plattform der Deutschen Telekom AG zuständig und verwendet dazu unter anderem das Bestandssystem Kosy-IP. Das Aufgabengebiet des Squad ist sehr breitgefächert. Dazu gehören: Automatisierte Konfigurationserstellungen für Router, sowohl für Label Edge Router (LER) als auch für Customer Premises Equipment (CPE), das Erstellen der Router Konfigurationen und dem Einbringen dieser Konfigurationen auf den LERs, Schleifentest, Pingtest, Inbetriebnahme mit dem Kunde, Bearbeiten von Bereitstellungsstörungen und das Erstellen der Konfiguration bei betriebsbedingten Umschaltungen.

Das Squad T-CNF-IPC NNA KIC hingegen kümmert sich um KoSy-IP sowie auch um die Automatisierung und Implementierung der Aufgaben des Squads CMC im Generischen Konfigurationssystem (GeKo). Das Ziel ist es, so viele Arbeitsschritte des Squads T-CNF-IPC CMC wie möglich zu automatisieren und das Bestandssystem KoSy-IP abzulösen. Dazu wird die Open Source Software „Xyna Factory“<sup>1</sup>, bei welchem es sich um eine modellorientierte und auftragsgetriebene Workflow-Plattform handelt, der Firma GIP Exyr GmbH verwendet. Hierbei handelt es sich um eine Konfigurationsplattform für Automatisierungsprozesse. Komplexe Prozesse können mithilfe der Xyna Factory in technische Teilprozesse aufgegliedert werden, um so große Infrastrukturen zu entwickeln<sup>2</sup>. Intern wird dazu die Programmiersprache Java verwendet.

Auf Seite der Deutschen Telekom AG arbeiten vier Entwickler an der Umsetzung der Automatisierung in GeKo. Zusätzlich stellt die Firma GIP Exyr GmbH Mitarbeiter, um die Entwickler der Deutschen Telekom AG zu unterstützen. Da das Team nach Scrum, eine Methode zur Agilen Softwareentwicklung,<sup>3</sup> arbeitet, kommen weitere Personen hinzu, welche Tätigkeiten wie Scrum Master oder Product Owner ausüben.

---

<sup>1</sup> (github -, kein Datum)

<sup>2</sup> (xyna-factory, 2022)

<sup>3</sup> (atlassian - Was ist Scrum?, kein Datum)

## 5 Konzeptionelle Grundlagen

Im Nachfolgenden werden die Themen XML (Extensible Markup Language), XSD (XSD Schema) sowie XPath (XML Path Language ) vorgestellt, da dieses Wissen zur Umsetzung der späteren Auftragsprüfung in GeKo erforderlich ist. Da diese drei Themen sehr ausgeweitet betrachtet werden könnten und dies dann den Rahmen der Probearbeit sprengen würde, werden pro Thema nur die wichtigsten und grundlegendsten Themen behandelt. Eine Vertiefung finden hierbei nicht statt.

### 5.1 XML-Grundlagen

XML ist ein von W3C (World Wide Web Consortium), bei welchem es sich um ein Gremium zu Standardisierung der Techniken im World Wide Web handelt<sup>4</sup>, eingeführter Standard zur Dokumentenauszeichnung. Es handelt sich dabei um eine Methode zur Speicherung und Verarbeitung von Strukturierten und textbasierten Daten. Strukturierte Daten sind zum Beispiel Adressbücher, Vektorgrafikprogrammen oder auch wie im Beispiel GeKo die Auftragsdokumente. Zur Datenspeicherung von Dateien mit großer Bitfolge, wie zum Beispiel Bilder, Ton oder Videos, ist XML ungeeignet.<sup>5</sup>

Das Wissen, wie ein XML-Dokument aufgebaut ist und wie es im Falle von GeKo verwendet wird, ist zwingend erforderlich, da genau dies der „Grundstein“ für die Auftragsprüfung ist. Um Validierungen zu erzeugen, muss genau bekannt sein, wie ein Auftrag für GeKo aufgebaut sein kann und welche Wertebereiche erlaubt sind. Ebenfalls dienen XML-Dokumente schlussendlich dazu, um die Auftragsprüfung zu testen.

#### 5.1.1 Elemente<sup>6</sup>

Jedes XML-Dokument hat genau ein einziges Stamm-, bzw. Root-Element. Dies beschreibt den Anfang und das Ende (Start- und Ende-Tag) eines Dokumentes und kann wie folgt aussehen:

```
<Personen>...</Personen>
```

Das Element besteht dabei immer aus einem Start-Tag (<Personen>) und einem Ende-Tag (</Personen>). Der einzige Unterschied ist dabei das „/“-Slash-Zeichen. Leere Elemente besitzen jedoch kein Ende-Tag, sondern den Slash am Ende des Namens (<Personen/>).

Zwischen Start- und Ende-Tag können dann weitere Elemente aufgeführt werden. Dadurch entsteht eine hierarchische Baumstruktur:

```

<Personen>
  <Person>
    <Vorname></Vorname>
    <Nachnamen></Nachnamen>
  </Person>
  <Person>
    <Vorname></Vorname>
    <Nachnamen></Nachnamen>
  </Person>
</Personen>

```

<sup>4</sup> (W3C, kein Datum)

<sup>5</sup> (Weber, 2002/03), Seite 1 - Was ist XML

<sup>6</sup> (Grupe, 2018) Seite 12 – XML-Grundlagen

Dies kann beliebig ausgebaut werden. Ebenfalls kann zwischen einem Start- und Ende-Tag Text hinzugefügt werden:

```
<Vorname>Niels</Vorname>
<Nachnamen>Häußler</Nachnamen>
```

### 5.1.2 Attribute<sup>7</sup>

Zusätzlich zu der oben beschriebenen Elementen-Schreibweise können Attribute verwendet werden. Ein Attribut wird dabei in das Start-Tag eines Elementes geschrieben. Die Informationen werden dabei dann den Attributen zugeordnet.

```
<Personen>
  <Person Vorname="Niels" Nachnamen="Häußler"/>
  <Person Vorname="Benedikt" Nachname="Häußler"/>
</Personen>
```

Die Vorteile sind, dass nicht für jeden Wert ein eigenes Element erstellt werden muss sowie die bessere Darstellung der hierarchischen Strukturen.

Der Nachteil wiederum ist, dass jedes Attribut, welches unter einem hierarchisch angeordneten Element steht, nur einmal vorkommen darf. Es können dadurch zum Beispiel nicht mehrere „Vornamen“ unter einem Element vorkommen.

### 5.1.3 Kommentare

In XML-Dokumenten ist es möglich, Kommentare zu verfassen, welche bei einer Auswertung nicht beachtet werden. Kommentare beginnen dabei immer mit <!-- und enden mit -->.<sup>8</sup> Hierbei stehen oftmals, so auch in der internen- und externen-XSD von KoSy-IP, wichtige Informationen, um den Aufbau oder bestimmte Regeln besser zu verstehen.

### 5.1.4 Prolog<sup>9</sup>

Am Anfang eines XML-Dokumentes kann ein sog. Prolog verwendet werden. Dieser ist jedoch nicht zwingend nötig und kann unterschiedlich groß ausfallen. Hier werden Zusatz-Informationen wie die Version oder der im Dokument verwendete Zeichensatz aufgeführt. Dies kann wie folgt aussehen:

```
<?xml version="1.0"
  Encoding="ISO-8859-1"?>
<Personen>
  <Person/>
</Personen>
```

Zusätzlich kann es noch Verarbeitungshinweise (Processing Instruction) geben. Diese werden von einem Programm benötigt, welches das XML-Dokument auswertet. Dies kann zum Beispiel so aussehen:

```
<?versandperMail verarbeitung@telekom.de"?>
```

<sup>7</sup> (Grupe, 2018), Seite 13 – XML-Grundlagen

<sup>8</sup> (XML - Kommentare, kein Datum)

<sup>9</sup> (Grupe, 2018), Seite 15 – XML-Grundlagen

## 5.2 XSD-Grundlagen

Die XSD ist für die Umsetzung der Regeln und Validierungen das Wichtigste. In ihr befinden sich die jeweiligen Regeln, welche die Attribute und Elemente überprüfen. Aus diesen Regeln können die unterschiedlichen Validierungen für GeKo erzeugt werden.

Eine XSD ist eine XML-basierte Schemasprache, welche den Aufbau sowie den Inhalt von XML-Dokumenten anhand von eigenen Regeln überprüft und validiert<sup>10</sup>.

### 5.2.1 SimpleType

SimpleType, eine einfacher Typ, sind Datentypen, welche nur einen Wert, zum Beispiel eine Zeichenkette (string) oder eine Zahl (integer, float, double, ...) annehmen können.<sup>11</sup> Die Deklaration eines einfachen Datentyps beginnt mit „simpleType“. Das Attribut „name“ legt den Namen des Datentyps an. Der Name kann dann für Elemente und Attribute innerhalb der XSD-Datei verwendet werden. Innerhalb des „simpleType“ folgt das Element „restriction“ mit seinem Attribut „base“, welches den Datentyp angibt:

```
<simpleType name="Vorname">
  <restriction base="string" />
</simpleType>
```

Ein deklarierter Datentyp kann dann für ein Element verwendet werden. Hierbei ist „name“ wieder der name des Elements und das Attribut „type“ der dazugehörige Datentyp, welcher innerhalb des Dokumentes festgelegt wurde:

```
<element name="Person" type="Vorname" />
```

Natürlich hätte das Definieren des Datentypes auch weggelassen werden können und anstatt des Wertes type="vorname", type="string" gewählt werden. Jedoch empfiehlt sich dies nicht. Einerseits verbessert es die Lesbarkeit eines XSD-Dokumentes, wenn alle Datentypen nicht im Element sondern einzeln deklariert sind und andererseits können somit mehrfach vorkommende Elemente mit demselben Datentype einfacher deklariert werden. Änderungen an einem Datentyp müssen damit nur an einer Stelle, nämlich dem deklarierten Datentype, und nicht an allen einzelnen vorkommenden Elementen geändert werden. Diese Art von Deklaration ist von enormer Bedeutung, wenn es um Einschränkungen geht. Dies wird später im Kapitel 5.2.3 behandelt.

---

<sup>10</sup> (XSD (XML Schema Definition), kein Datum)

<sup>11</sup> (homepage-webhilfe- Einfache Typen, kein Datum)

## 5.2.2 ComplexType

ComplexType, ein komplexer Typ, sind Datentypen, welche lediglich bei Elementen vorkommen und untergeordnete Elemente oder Attribute besitzen.<sup>12</sup> Die Deklaration der komplexen Typen erfolgt ebenfalls nach der Syntax eines einfachen Datentyps, wie sie im Kapitel 5.2.1 (SimpleType) beschrieben werden. Der Unterschied besteht jedoch darin, dass die Deklaration mit dem Element „complexType“ erfolgt und mehrere Attribute oder Elemente besitzt. Ebenso gibt es verschiedene Arten, welche innerhalb des komplexen Datentyps vorkommen:

### Sequenz<sup>13</sup>

Sequenzen bestehen aus mehreren Elementen. Innerhalb der Sequenz befinden sich dann die einzelnen unterschiedlichen Elemente, welche entweder wieder Elemente oder aber auch Attribute (Kapitel 0) besitzen:

Das folgende XML-Dokument dient uns als Beispiel:

```
<Person>
  <Vorname >Niels </Vorname >
  <Nachname >Häußler </Nachname >
  <Email >niels@telekom.de </Email >
</Person>
```

Das obere XML-Dokument wird von der folgenden XSD überprüft:

```
<element name="Person ">
  <complexType>
    <sequence>
      <element name=" Vorname" type="string" />
      <element name=" Nachname" type="string" />
      <element name="Email" type="string" />
    </sequence>
  </complexType>
</element>
```

Um festzulegen wie oft ein Element der Sequenz vorkommen darf, können die Attribute „minOccurs=“...“ “ und „maxOccurs=“...“ “, welche in die jeweiligen Zeilen der Elemente niedergeschrieben werden, verwendet werde. Diese geben die minimale und maximale Anzahl eines auftretenden Elements an. Wenn dies nicht angegeben wird, muss jedes Element genau einmal auftreten:

```
<element name="Vorname" type="string" maxOccurs="unbounded" minOccurs="1" />
```

<sup>12</sup> (homepage-webhilfe - Komplexe Typen, kein Datum)

<sup>13</sup> (w3schools - XML Schema sequence Element, kein Datum)

## Auswahl<sup>14</sup>

Das Element „choice“ definiert eine Auswahl an Möglichkeiten, welche der Datentyp annehmen kann. Es ist nicht auf einen Datentyp festgeschrieben:

```
<element name="Version" >
  <complexType>
    <choice>
      <element name="Version_5.1" type="string" />
      <element name="Version_5.2" type="string" />
    </choice>
  </complexType>
</element>
```

Hierbei werden beide Elemente Versionsarten, 5.1 sowie 5.2, angenommen. Andere hingegen abgelehnt.

## Attribute<sup>15</sup>

Attribute werden mit dem Element „attribut“ definiert. Dieses findet man im Regelfall direkt im „complexType“ niedergeschrieben:

```
<complexType>
  <attribut name="Vorname" type="string"/>
  <attribut name="Nachname" type="string"/>
  <attribut name="Alter" type="string"/>
</complexType>
```

Wenn dabei ein Attribut vorkommen muss und nicht nur optional ist, kann in der geforderten Zeile ein „use="required"“ hinzugefügt werden:

```
<attribut name="Vorname" type="string" use="required" />
```

Andere Punkte, welche mitgegeben werden können, sind:

- default="..."      Dieser Wert wird verwendet, wenn kein Wert gesetzt ist.
- fixed="..."      Es wird ein fester Wert angegeben. Andere sind ungültig.
- Use="optional"      Das attribute ist optional.

<sup>14</sup> (homepage-webhilfe - Komplexe Typen, kein Datum), Auswahl

<sup>15</sup> (pagina - DEKLARATION VON ATTRIBUTEN IN XSD, kein Datum)

### 5.2.3 Einschränkungen<sup>16</sup>

Mit Hilfe von einem „restriction“-Element können Einschränkungen definiert werden. Restrictions beziehen sich dabei immer auf einen vorhandenen Datentyp. Diese können auch selbst definierte Datentypen sein. Der Datentyp wird dabei als „base“ Attribut festgelegt. Dem Restriktion-Element sind dann die jeweiligen Einschränkungen untergeordnet.

#### Länge:

Mit Hilfe von „minLength“, „maxLength“ und „length“ können die Länge von Zeichenketten oder Listen festgelegt werden. Dabei geben minLength und maxLength den Minimal- bzw. den Maximalwert und length den exakten Wert einer Zeichenkette an.

```
<element name="Vorname">
  <simpleType>
    <restriction base="string">
      <minLength value="3"/>
      <maxLength value="30"/>
    </restriction>
  </simpleType>
</element>
```

#### Wertebereich:

Zahlen können mithilfe von „minInclusive“ und „maxInclusive“ eingeschränkt werden. Zahlen müssen hierbei kleiner bzw. größer oder gleich dem definierten Wert sein. Wenn nur kleiner oder größer eingeschränkt werden soll, dann müssen „minExclusive“ und „maxExclusive“ verwendet werden.

```
<element name="Zahl ">
  <simpleType>
    <restriction base="integer">
      <minInclusive value="10"/>
      <maxInclusive value="99"/>
    </restriction>
  </simpleType>
</element>
```

#### Dezimalstellen:

Mit Hilfe von „fractionDigits“ kann die maximale Anzahl von Nachkommastellen festgelegt werden. Jedoch nur für das Element „decimal“. Mit „totalDigits“ hingegen kann die maximale Anzahl von Dezimalstellen definiert werden.

```
<element name="Zahl ">
  <simpleType>
    <restriction base="decimal">
      <fractionDigits value="5"/>
    </restriction>
  </simpleType>
</element>
```

<sup>16</sup> (homepage-webhilfe. - Einschränkungen, kein Datum)

### Listenwerte:

Listenwerte sind sogenannte Aufzählungen von strings welche als Wert wiederum in einem string Attribut erlaubt sind. Es handelt sich somit um eine Auswahl von erlaubten Werten. Dazu wird das Element „enumeration“ verwendet. Dieses kann beliebig oft unterhalb eines restriction Elements aufgeführt werden.

```
<element name="Vertragsergaenzung ">
  <simpleType>
    <restriction base="string">
      <enumeration value="Vlan"/>
      <enumeration value="BUNDLE"/>
      <enumeration value="WLAN"/>
    </restriction>
  </simpleType>
</element>
```

### Reguläre Ausdrücke:

Reguläre Ausdrücke, also die Überprüfung eines Wertes auf bestimmte Merkmale, können mit einem „pattern“ überprüft werden. Dabei können auch mehrere „pattern“ unterhalb eines „restriction“-Element aufgeführt werden.

```
<element name="IPv6 ">
  <simpleType>
    <restriction base="string">
      <pattern value="([0-9A-F]){4}:([0-9A-F]){4}:([0-9A-F]){4}"/>
    </restriction>
  </simpleType>
</element>
```

In allen hier dargestellten Fällen, hätte die Sequenz <simpleType>...</simpleType> als eigener definierter Datentyp ausgegliedert werden können, wie es im Kapitel 5.2.1 (SimpleType) beschrieben wird.

## 5.3 XPath

Mit XPath-Ausdrücken können auf Strukturen und Inhalte innerhalb eines XML-Dokuments zugegriffen werden. Somit können die darin beinhalteten Werte abgefragt und ausgelesen werden.<sup>17</sup> Das Ansprechen von Werten ist für eine Validierung zwingend notwendig.

Um XPath-Ausdrücke besser zu verstehen, werden diese anhand des folgenden XML-Beispiel erläutert:

```
<Personen>
  <PERSON Nachname="Häußler" Alter="20">
    <Vorname>Niels</Vorname>
    <Vorname>Fabien </Vorname>
  </PERSON >
  <PERSON Nachname="Müller" Alter="42">
    <Vorname>Otto </Vorname>
  </PERSON>
</Personen>
```

<sup>17</sup> (data2type - XPath – Einführung in die Technik des Zugriffs auf die XML-Strukturen, 2007)

### 1. Beispiel

Wenn der Vorname der ersten Person ermittelt werden soll, kann dieser mithilfe folgendes XPath angesprochen werden:

```
/Personen/PERSON[1]/Vorname[1]/text()
```

Der Teil „/text()“ könnte jedoch auch weggelassen werden. Als Ergebnis liefert der XPath „Niels“ zurück, was der ersten Vornamen der ersten Person ist. Die Prädikate [1] bedeuten hierbei immer das erste Vorkommen der Elemente gleichen Namens im XML-Dokument.

### 2. Beispiel

Wenn jedoch das Alter der Person, welche den Nachnamen Müller besitzt, ausgelesen werden soll, benötigt man folgenden XPath-Ausdruck:

```
/Personen/PERSON[@Nachname="Müller"]/@Alter
```

Als Ergebnis wird „42“ zurückgeleitet. Das @-Zeichen wird hier ausdrücklich benötigt, da ein Attribut und nicht wie im oberen Beispiel ein Element angesprochen wird.

Aus diesen beiden Beispielen lässt sich erkennen, dass, um einen Wert innerhalb eines XML-Dokuments anzusprechen, ein XPath mit der hierarchischen Struktur, in welchem der Wert liegt, gebaut werden muss. Dabei beginnt der XPath immer mit einem Slash (/). Ebenfalls trennen Slashes die einzelnen hierarchischen Strukturen bis zum letzten Element oder einem Attribut. Um einen Wert eines Elementes anzusprechen, wird nur der Name des Elementes benötigt. Bei Ansprechen eines Attributs hingegen wird ein @-Zeichen sowie der Name des Attributes gebraucht.<sup>18</sup> Sollen Bedingungen eingebaut werden, können diese innerhalb des Pfades mit eckigen Klammern [...] sowie der darin enthaltenen Bedingung dargestellt werden.<sup>19</sup>

Vorsicht ist jedoch geboten, wenn ein XPath nicht genau deklariert wird. Dann kann er mehrere Werte zurückliefern, da er mehrere Stellen im XML-Dokument anspricht. Dies kann aber auch gewollt sein: Wenn zum Beispiel alle Nachnamen aus einem Dokument gefunden werden sollen, kann der XPath Ausdruck so aufgebaut werden:

```
//@Nachname
```

Hierbei werden alle Werte der Attribute, welche den Namen „Nachname“ besitzen, zurückgeliefert. Die beiden anfänglichen Slashes bedeuten hierbei, dass zwischen diesen beiden Slashes jegliche untergeordneten hierarchischen Strukturen gelten sollen.<sup>20</sup>

<sup>18</sup> (Grupe, 2018), Seite 138 - XPath

<sup>19</sup> (ionos - XPath-Tutorial für Einsteiger, 2020)

<sup>20</sup> (w3schools - XPath Syntax, kein Datum)

## 6 Umsetzung der Auftragsprüfung

Die Umsetzung der Auftragsprüfung erfolgt in drei Schritten: Analyse, Implementierung im Regel- und Validierungsmechanismus und das Testen. Hierbei werde ich nur auf die Analyse so wie die Funktionalität des Regel- und Validierungsmechanismus eingehen, da die schlussendliche Implementierung so wie das Testen nur einen kleinen und wenig beschreibbaren Anteil besitzt.

### 6.1 Analyse der internen und externen XSD

Die interne sowie externe XSD müssen gegeneinander verglichen und analysiert werden. Grund dafür ist, dass beide zwar gleich aufgebaut sind aber sowohl unterschiedliche als auch gleiche Inhalte besitzen. Da jedoch nur die Inhalte, welche ausschließlich in der internen und nicht auch in der externen XSD vorhanden sind als Regeln und Validierungen in GeKo implementiert werden sollen, muss diese Analyse stattfinden.

Bei der Analyse stellte sich heraus, dass die interne XSD oftmals deutlich genauere restrictions (beschrieben in Kapitel: 5.2.3) als die externe XSD besitzt.

So sind oftmals in der externen XSD nur Einschränkungen mit „pattern=“.+“ “. Der Punkt (.) steht hierbei für „Jedes Zeichen“ und das Plus (+) für „mindestens ein oder mehr Zeichen“<sup>21</sup>. Das bedeutet, dass der Wert, welcher innerhalb des XML-Dokumentes überprüft werden soll, vorhanden sein und mindestens ein Zeichen beinhalten muss. Andere Anforderungen werden hierbei nicht überprüft. Die interne XSD hingegen verwendet dann entweder eine genauere pattern-Funktion, welche den vorhandenen Inhalt detaillierter überprüft oder aber auch eine enumerations-Funktion welche diversen Aufzählungen vorgibt (Kapitel 5.2.3). Diese Art von restrictions müssen später in GeKo implementiert werden.

<pre>&lt;xs:simpleType name="Auftrags_ID_SSt_stype"&gt;   &lt;xs:restriction base="xs:string"&gt;     &lt;xs:maxLength value="84"/&gt;     &lt;xs:pattern value=".\+-\p{N}+-\p{N}+/"&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>	<pre>&lt;xs:simpleType name="Auftrags_ID_SSt_stype"&gt;   &lt;xs:restriction base="xs:string"&gt;     &lt;xs:maxLength value="84"/&gt;     &lt;xs:pattern value=".\+/"&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>
---	---

Abbildung 1-Interne(links)-vs Externe(rechts)-XSD (1.Fall)  
Quelle: Eigene Aufnahme

Es gibt aber auch andere Varianten. So können beide Einschränkungen, in der internen sowie auch in der externen XSD gleich sein. Da es keine Unterschiede gibt, müssen diese Art von Einschränkungen nicht implementiert werden.

<pre>&lt;xs:simpleType name="Sender_Name_stype"&gt;   &lt;xs:restriction base="xs:string"&gt;     &lt;xs:maxLength value="60"/&gt;     &lt;xs:pattern value=".\+/"&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>	<pre>&lt;xs:simpleType name="Sender_Name_stype"&gt;   &lt;xs:restriction base="xs:string"&gt;     &lt;xs:maxLength value="60"/&gt;     &lt;xs:pattern value=".\+/"&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>
---	---

Abbildung 2-Interne(links)-vs Externe(rechts)-XSD (2.Fall)  
Quelle: Eigene Aufnahme

<sup>21</sup> (regexe, 2017)

Eine weitere Variante, welche nur sehr selten auftritt, ist, dass die externe XSD eine genauere Prüfung als die interne XSD durchführt. Bei diesem Fall muss ebenfalls nichts implementiert werden, da schlussendlich nur die interne XSD entfernt werden soll und somit die detaillierte Prüfung bestehen bleibt. Eine Implementierung dieser Art von Regeln würde eine Doppelung darstellen und keinen Mehrwert erbringen.

```

<xs:simpleType name="Routernummer_stype">
  <xs:restriction base="xs:unsignedLong">
    <xs:maxInclusive value="9999999999999999"/>
  </xs:restriction>
</xs:simpleType>

```

```

<xs:simpleType name="Routernummer_stype">
  <xs:restriction base="xs:unsignedLong">
    <xs:maxInclusive value="9999999999999999"/>
    <xs:pattern value="\-+]?[0-9]+"\>
  </xs:restriction>
</xs:simpleType>

```

Abbildung 3-Interne(links)-vs Externe(rechts)-XSD (3.Fall)  
Quelle: Eigene Aufnahme

Zu jeder Regel werden XPath-Ausdrücke benötigt, welche ebenfalls herausgearbeitet werden müssen. Die XPath-Ausdrücke sind notwendig, um später die Validierungen erzeugen zu können. Dazu müssen alle Elemente im Auftragsdokument, welche von den herausgearbeiteten Regeln überprüft werden sollen, gefunden werden. Zu jedem einzelnen Wert muss dann ein XPath-Ausdruck erstellt werden und seiner Regel zugeordnet werden. Mehr zur Funktion und Bildung von XPath-Ausdrücken im Kapitel 5.3.

Ebenfalls musste herausgearbeitet werden, ob ein Wert mit seinem XPath verpflichtend ist (Kapitel 5.2.2) oder nicht. Also, ob ein Wert definitiv im Auftrag gefunden werden muss oder ob dieser Wert optional ist und auch fehlen darf.

Die aus der Analyse herausgearbeiteten Regeln und XPath-Ausdrücke können im Anhang (Kapitel 8) betrachtet werden.

## 6.2 Regeln und Validierungen in GeKo

Regeln und Validierungen werden in GeKo dazu verwendet, um Auftragsdokumente auf Richtigkeit zu überprüfen. Fehlerhafte oder syntaktisch falsche Aufträge können durch diesen Mechanismus nicht mehr konfiguriert werden, wodurch Komplikationen und Fehler vermieden werden.

Der Mechanismus ist wie folgt aufgebaut:

1. **Eine Regel**, welche zum Beispiel die korrekte Syntax einer IP-Adresse überprüft
2. **Eine oder mehrere Validierungen**, welche zum Beispiel einen XPath-Ausdruck zu einer IP-Adresse im Auftragsdokument und den Verweis zu der jeweiligen Regel (1.) besitzt bzw. besitzen.

Auf diese Weise sind Regeln und Validierungen unabhängig, sodass Regeln mehrfach verwendet werden können, wie z.B. den Betriebsstatus von verschiedenen Netzelementen, welcher mehrfach in einem Auftrag vorkommen kann. Die Prüfung eines Auftrages kann nur in Kombination mithilfe einer Regel sowie einer Validierung erfolgen. Da jedoch eine Validierung nur gezielt einen XPath-Ausdruck besitzen kann, müssen zu jedem Wert, welcher von einer Regel geprüft werden soll, eine eigene Validierung angelegt werden.

Eine Regel an und für sich reicht nicht aus, um eine Überprüfung eines Wertes im Auftragsdokument zu realisieren. Dasselbe gilt für eine Validierung, jedoch mit dem Unterschied, dass eine Validierung ohne Regel technisch gar nicht angelegt werden kann. Eine Regel kann jedoch ohne Validierung existieren.

### 6.2.1 Regel

Da unterschiedliche Inhaltsarten in einem Auftrag vorkommen können, muss es verschiedene Regel Varianten geben. In der GeKo GUI gibt es deshalb insgesamt drei verschiedenen Arten von Regel-Typen welche bei der Anlegung ausgewählt werden können:

Abbildung 4: Anlegen einer neuen Rege  
Quelle: Eigene Aufnahme

- Syntax-Regel:

Ein textueller Wert wird syntaktisch auf Korrektheit überprüft. Dies erfolgt anhand eines Regex-Ausdruckes (Regular expression). Dies entspricht einem „pattern“ aus der XSD.

Beispiele:

- IP4-Adressen (vier Zahlen zwischen 0 und 255, durch Punkte getrennt)<sup>22</sup>
- Hostname eines Netzelements (besteht nur aus zulässigen Zeichen)

- Aufzählungstyp:

Mit dieser Regel wird geprüft, ob ein Wert aus dem Auftragsdokument in einer Liste von zulässigen Werten enthalten ist. Dies entspricht einem „enumeration“ aus der XSD.

Beispiele:

- Pflege der zulässigen Vertragsgegenstandsnummern
- Pflege von zulässigen QoS-Klassen (Quality of Service)

- Referenz-Regel:

Referenz-Regel Mit der Referenzregel kann geprüft werden, ob ein bestimmter Wert im Auftragsdokument an einer anderen Stelle im Auftragsdokument erneut auftaucht.

Beispiele:

- Ein zugewiesenes QoS-Profil muss an einer anderen Stelle im Auftrag definiert sein

Die Punkte „Name“ und „Beschreibung“ haben keinen Einfluss auf die Regel und sind frei wählbar.

## 6.2.2 Validierung

Damit eine Validierung greift, muss eine Validierung einen XPath-Ausdruck besitzen, welcher an eine Stelle im XML-Auftragsdokument verweist.

Beispiel:

**`/Auftrag/Vertragsgegenstand/@Vertragsgegenstandsnummer`**

Das Attribut „Vertragsgegenstandsnummer“ im XML-Element „Vertragsgegenstand“, welches sich unterhalb des XML-Elements „Auftrag“ befindet. Durch diesen Ausdruck kann das Objekt Vertragsgegenstandsnummer anhand einer gegebenen Regel überprüft werden.

Es ist hierbei zu beachten, dass ein XPath-Ausdruck mehrere Ergebnisse liefern kann. Dies tritt zum Beispiel auf, wenn mehrere Netzelement im Auftrag vorkommen. Es werden alle zurückgelieferten Werte validiert.

---

<sup>22</sup> (xplore-dna - IP-Adressen, 2021)

The screenshot shows a dialog box titled "Validierung hinzufügen". It has the following fields and controls:

- XPath-Ausdruck:** A text input field.
- Pflichtfeld:** A checkbox.
- Art:** Two radio buttons, with "Validierung" selected.
- Regel:** A dropdown menu showing "<Keine ausgewählt>".
- Beschreibung:** A text input field.
- Aktiv:** A checkbox.
- Buttons:** "Abbrechen" (with a close icon) and "Sichern" (with a checkmark icon) at the bottom right.

Abbildung 5: Anlegen einer neuen Validierung  
Quelle: Eigene Aufnahme

Jedoch sind Aufträge von unterschiedlicher Größe und Art, was dazu führt, dass nicht immer alle Validierungen greifen können, da der gegebene XPath-Befehl ins Leere zeigt, da bestimmte Daten in diesem Auftrag nicht gegeben sind. Deshalb können Validierungen als Pflicht (Pflichtfeld) betitelt werden. Dies bedeutet, dass die Validierung fehlschlägt, wenn entweder der Wert nicht der Regel entspricht oder überhaupt nicht vorhanden ist. Wenn die Validierung nicht als Pflicht deklariert ist, dann schlägt diese nur an, wenn ein Wert gefunden ist und dieser nicht der Syntax der Regel entspricht.

Ebenfalls kann die Validierung auf zwei verschiedene Arten durchgeführt werden, einer Sperrung oder einer Validierung:

1. **Sperrung:** Wenn eine Sperr-Regel greift, kann der Auftrag nicht konfiguriert werden. Dies kann dazu verwendet werden, um z.B. temporäre Aufträge mit bestimmten Vertragsgegenstandsnummern zu sperren. Jegliche Aufträge, welche die Sperr-Regel erfasst, werden gesperrt und können nicht konfiguriert werden.
2. **Validierung:** Diese überprüft bei Auftragseingang und beim Starten der Konfiguration den Auftrag. Wenn ein falscher Wert anhand der gegebenen Regel gefunden wurde, wird der Auftrag abgelehnt. Wenn die Regel nicht anschlägt, kann der Auftrag konfiguriert werden

Den Unterschied zwischen Sperrung und Regel erkennt man sehr gut an einem Beispiel:

*Ein Attribut soll nur Zahlen zwischen 1 und 60 besitzen:*

*Sperrung:* Syntax-Regel mit einem Regex [-∞ bis 0 und 61 bis +∞]

*Validierung:* Syntax-Regel mit einem Regex [1-60]

Im Falle der Umsetzung der Auftragsprüfung werden nur Validierungen und keine Sperren eingesetzt. Sperren sollten nur temporär und nicht dauerhaft verwendet werden. Deshalb sind diese für die Auftragsdokumentenprüfung ungeeignet.

Die Punkte „Name“ so wie „Beschreibung“ sind frei wählbar und haben keinen Einfluss auf die Validierung. Empfehlenswert ist hier in der Beschreibung eine Fehlermeldung sowie den XPath-Ausdruck anzugeben. Somit ist dem Sachbearbeiter sofort erkenntlich, wo im Auftrag ein Fehler vorhanden ist, und kann diesen gezielt beheben.

## 7 Abschluss und Blick in die Zukunft

Im Rahmen dieser Projektarbeit konnte die Auftragsdokumentenprüfung analysiert und vereinfacht werden. Mithilfe der Analyse so wie der Implementierung in der Regel und Validierungsmechanismus von GeKo konnte die interne XSD entfernt werden und somit die Auftragsüberprüfung übersichtlicher und einfacher gestaltet werden. Um die Funktionalität des Mechanismus zu überprüfen, wurde dieser mit gezielt manipulierten XML-Aufträgen, welche somit bestimmte Fehler aufwiesen, getestet. Dadurch konnte der Validierungsmechanismus als korrekt funktionierend eingestuft werden.

Insgesamt ist der Regel und Validierungsmechanismus von GeKo positiv und effektiv zu bewerten. Er ermöglicht das schnelle Einbauen, Ändern und Deaktivieren von Validierungen und Regeln ohne umständlich XSD-Dateien zu verändern und neu einspielen zu müssen. Durch die weitere Funktion der "Sperrung" kann der Mechanismus gut im Wirkbetrieb eingesetzt werden.

Jedoch besitzt der Regel- und Validierungsmechanismus auch Schwachstellen. So lassen sich nicht alle Auftragsprüfungen, welche für den Wirkbetrieb von GeKo benötigt werden, darin umsetzen. Nur Referenzen, Werte und RegExe, wie sie auch in der XSD vorkommen, können abgefragt und überprüft werden. Komplexe Prüfungen, wie zum Beispiel den Aufbau von bestimmten Aufträgen je nach bestimmten Werten, zum Beispiel nach einer Vertragsgegenstandsergänzung, sind nicht möglich und müssen über die Xyna Factory weiterhin abgebildet werden. Es kann gesagt werden, dass das Ziel, nämlich die Implementierung der internen XSD, erfolgreich war, jedoch der Mechanismus nicht für eine komplette Auftragsprüfung von GeKo geeignet ist.

Die Implementierungen in GeKo wurden von den Auftraggebern inklusiv Tests im TCT (Projekt Test Center Technik) vorgestellt und angenommen. Mit dem Beginn des Wirkbetriebes ab Ende Juli 2022 sind die Regeln und Validierungen fester Bestandteil der aktiven Auftragsüberprüfung.

## 8 Anhang

### Regeln

Name	Regelart	Regel
Negotiation	enum	Auto,Nein,Ja
OMS_Prozess	enum	Ungekoppelte Neueinrichtung VPN, Änderung Kundenrouting VPN, Bitratenänderung ohne Router-Tausch VPN, Kündigung Zweibegeanbindung, Neueinrichtung_Doppelte_Anbindung_VPN, Ungekoppelte Kündigung VPN, Interne Router Konfiguration, Software Update, Router-Tausch VPN, Pop-Schwenk, Änderung Zweibegeanbindung VPN, Produktwechsel VPN, Standortänderung VPN
Auftragsart	syntax	( manuell,)(Alles AutomatischeKonfiguration Nur_CPE Nur_Plattform Kuendigung Bitratenaenderung Aenderung_mit_Router-Tausch Aenderung Aenderung_Protokollhub Umschaltung Softwareupdate Stornierung)
Beziehung_zu_QoS_Profilname	reference	/Auftrag/Netzelement/QoS/Profil/@Profilname
Auftragsart Kündigung	syntax	Kuendigung
Betriebsstatus	enum	Planned,Active,Shutdown-Planned,Planned (Folgeversion)
Verbindung_Attribut_Typ	enum	nominelle_Bitrate,LSZ,LSZ_Erg,EndSZ_A,EndSZ_B,OrdNr,Leistungsbezeichnung,Access_EVC_ID,Line_ID,Anschlussnummer
NE_Funktion	enum	LER,CPE,MSAN,RD_Remote
Netzmanagement	enum	managed,unmanaged
Hersteller	enum	Cisco,Juniper,OneAccess,Unbekannt
IFC_Funktion	enum	WAN,LAN,Management,Alternativ_LAN,SLA_Probe,CROSS_LAN,Looopback,Dokumentation,GL_WAN,Physik,Tunnel
VRF_Attribut_Typ	enum	VRF_Routen_Maximum,Threshold,eiBGP_Multipath,BGP_SET
Interface_Attribut_Typ	enum	NNI-Typ,MTU_Size,InnerTag,InnerTagEtherType,OuterTag,OuterTagEtherType,L2-Produktvariante,fControlWord,SDSL-Variante,L3-Produktvariante,RateLimit,VlanSubbitrate,BFD_Min_Interval,BFD_Multiplier,Negotiation,MediaType,EthernetSpeed,Scramble,FlowControl,Framing,FrameRelayType,NationalBit,NutzBits,DSUMode,PDHMode,ClockSource,LineCode,T1_speed,LACP_MinActiveLinks,LACP_Rate,DLCI,LMIType,LanMode,Typ,Trust_DSCP,ARP_Timeout,Routingvariante,AddDropInnerTag,AddDropOuterTag,DropFramesInbound,DropFramesOutbound,IPMB_ServiceName,NNI_Management,CFM_Interval,CFM_Domainname,MPLS_Label,Remote_VLAN_ID,IPX_Hub_Service,IPX_Servicekante,iACL,GRE_Tunnel_source_Adresse,GRE_Tunnel_destination_Adresse,3rd_Party_NNI_Uebergang,IPX_Customer_Service,IPX_Tunnel_Accounting_Key
BGP_Attribut_Typ	enum	Password,Community_High,Community_Low,Community_Delete,Export_Filter_Prefix,Import_Filter_Prefix,Maximum_Prefix,Site_Of_Origin,AS_Override,Remove_Private_AS,Hop_Count,Behind_ASN,Multihomed_at_ASN,Community_Steuerung,Export_Variant

		e_IPMB,Export_Variante_IPX_Customer,Export_Variante_IPX_Hub,Export_Variante_IPX_Provider,Export_Variante_WIA,Import_Variante,Allow_Community,Allow_AS_In,Replace_AS,IPv4_Export_Variante,IPv6_Export_Variante,Restart_Timer,Peer_FQDN,Servicekante,set_localpreference,set_community,community2MED,Export_Filter_mark_Prefixset,Import_Filter_mark_Prefixset,Export_Filter_block_Prefixset,export_as_path_prepend,Peerlock_Upstream_ASN,BGP_TTL_Security
Virtual_Switch_Attribut_Typ	enum	VPLS_Typ,Encapsulation,MAC_aging_time,MAC_table_size
Pseudowire_Attribut_Typ	enum	PW_Typ,Encapsulation
BGP_Variante_stype	enum	eBGP_VPN_UNI,eBGP_VPN_NNI,eBGP_VPN_Betriebsnetz,eBGP_Label,eBGP_VPN_IPX,eBGP_IPX_LAN_Customer,eBGP_IPX_LAN_Provider,eBGP_IPX_LAN_Hub,iBGP_IPX_RR,eBGP_IPX_Port_Customer,eBGP_IPX_Port_Provider,eBGP_IPMB,eBGP_WIA,eBGP_Public,eBGP_Public_IPMB,eBGP_10A_NNI,eBGP_Public_PFS,eBGP_Public_OAM,eBGP_Public_VOICE,eBGP_NICAs,eBGP_Public_CDN,eBGP_VPN_CDN,eBGP_Public_SEC_SA,eBGP_VPN_SEC_SA,eBGP_RAN_sharing,eBGP_IPX_Port_Hub,eBGP_IPX_IPSec_Customer,eBGP_NGVI,eBGP_Public_ADPP
Rolle_Beziehung_Verbindung_uebergeordnet	enum	Grundleitung,LACP Channel Verbindung,Basisverbindung,Verbindung zum Kunden,Verbindung zum NNI
Rolle_Beziehung_Verbindung_untergeordnet	enum	Anschluss Basisleitung,LACP Member Verbindung,Teilstrecke Kunde,Teilstrecke NNI,Subverbindung
Rolle_Beziehung_Interface_uebergeordnet	enum	Crossconnect CE WAN,IP Adress Lieferant (IPv4),IP Adress Lieferant (IPv6),LACP Channel Interface,Basisinterface,IP-Routing CE WAN,Basislfc_parser_gen
Rolle_Beziehung_Interface_untergeordnet	enum	Crossconnect CE LAN,IP Adress Empfaenger (IPv4),IP Adress Empfaenger (IPv6),LACP Member Interface,Subinterface,IP-Routing CE LAN,Basislfc_parser_gen
Ja_Nein_stype	enum	Ja,Nein
NNI_Klasse	enum	CIS_NNI_Provider,CIS_NNI_3rd_Party
Typ	enum	Loopback0,Loopback1,Transitnetz,Transit
Auftrag/Netzelement/QoS/Klasse/@Subklasse	enum	BestellwertBE,Flooding
Auftrag/Verbindung/@Wegeart3	enum	Erstweg,Zweitweg,Einzelweg
Auftrag/Netzelement/Doppelte_Anbindung/@Wegeart2	enum	Erstweg,Zweitweg

Auftrags_ID_St	syntax	.\p{N}+\p{N}+
Auftrags_ID_OMS	syntax	^\{1,60\}\$
Projekt_Kurzbezeichnung	syntax	[A-Za-z\p{N}\p{Pd}]+
Hostname	syntax	[A-Za-z_0-9\p{Pd}]+
Domainname	syntax	[A-Za-z.\p{N}\p{Pd}]+
FQDN	syntax	[A-Za-z._\p{N}\p{Pd}]+
Netzabschnitt	syntax	[A-Z]{1}
Filter_Makro	syntax	MC CD QG VLAN_\d+ VLAN_\d+-\d+
Auftrag/Netzelement/Interface/Statische_Route/@Metrik	syntax	^\{1-9\} \{1-9\}\{0-9\} 1\{0-9\}\{0-9\} 2\{0-5\}\{0-5\}\}\$
Projekt_Nummer	syntax	^\{1-9\}\{0-9\}\{0-9\}\{0-9\}\{0-9\}\}\$
Routercode	syntax	^\{0-9\} \{0-9\}\{0-9\} \{0-9\}\{0-9\}\{0-9\} \{0-9\}\{0-9\}\{0-9\}\{0-9\}\}\$
IPv6_Adresse-link-local	syntax	(?!FE80)(([0-9A-F]){4}:){7}([0-9A-F]){4}
Auftrag_Anschlusskurzname	syntax	[0-9]+-[0-9]+-[0-9]+
Vertragsgegenstandsergaenzung	syntax	((ADPP_(P V)(_* )) IPX_Customer_GRE RouteReflector BUNDLE (Multi-VPN(_Port _PORT _VLAN))) VLAN GATEF ZISPF ngIPsec IPsec-old bei_KoSy_Admin_zu_erfragen)
IP_Adresse_v4_v6_stype	syntax	((([25[0-5] 2[0-4][0-9] 1[0-9][0-9] [1-9][0-9] 0[0-9])\.)\{3\}25[0-5] 2[0-4][0-9] 1[0-9][0-9] [1-9][0-9] 0[0-9]) ([0-9A-F]){4}:){7}([0-9A-F]){4})
Route_Attribut_Typ_stype	enum	Routepolicy

## Validierungen

XPath	Name	Pflichtfeld
/Auftrag/@OMS_Prozess	OMS_Prozess	true
/Auftrag/@Auftragsart	Auftragsart	true
/Auftrag/Netzelement/Interface/Beziehung_zu_QoS_Profil/@Profilname	Beziehung_zu_QoS_Profil	false
/Auftrag/@Auftragsart	Sperre Auftragsart Kuendigung	false
/Auftrag/Vertragsgegenstand/@Vertragsgegenstandsergaenzung	Vertragsgegenstandsergaenzung	false
/Auftrag/Netzelement/Firmware/@Betriebsstatus	Betriebsstatus-Firmware	true
/Auftrag/Netzelement/Interface/Interface_Attribut/@Betriebsstatus	Betriebsstatus-Interface_Attribut	false

/Auftrag/Netzelement/Interface/IPv4_Adresse/@Betriebsstatus	Betriebsstatus-IPv4_Adresse	false
/Auftrag/Netzelement/Interface/IPv6_Adresse/@Betriebsstatus	Betriebsstatus-IPv6_Adresse	false
/Auftrag/Netzelement/Interface/HSRP/IPv4_virtuell/@Betriebsstatus	Betriebsstatus-IPv4_virtuell	false
/Auftrag/Netzelement/Interface/HSRP/IPv6_virtuell/@Betriebsstatus	Betriebsstatus-IPv6_virtuell	false
/Auftrag/Netzelement/Interface/HSRP/@Betriebsstatus	Betriebsstatus-HSRP	false
/Auftrag/Netzelement/Interface/Statische_Route/IPv4_Route/@Betriebsstatus	Betriebsstatus-IPv4_Route	false
/Auftrag/Netzelement/Interface/Statische_Route/IPv6_Route/@Betriebsstatus	Betriebsstatus-IPv6_Route	false
/Auftrag/Netzelement/Interface/BGP/IPv4_BGP_session/@Betriebsstatus	Betriebsstatus-IPv4_BGP_session	false
/Auftrag/Netzelement/Interface/BGP/IPv6_BGP_session/@Betriebsstatus	Betriebsstatus-IPv6_BGP_session	false
/Auftrag/Netzelement/Interface/Virtual_Switch/Virtual_Switch_Attribut/@Betriebsstatus	Betriebsstatus-Virtual_Switch_Attribut	false
/Auftrag/Netzelement/Interface/Virtual_Switch/Virtual_Switch_Gegenstelle/@Betriebsstatus	Betriebsstatus-Virtual_Switch_Gegenstelle	false
/Auftrag/Netzelement/Interface/Virtual_Switch/@Betriebsstatus	Betriebsstatus-Virtual_Switch	false
/Auftrag/Netzelement/Interface/Pseudowire/Pseudowire_Attribut/@Betriebsstatus	Betriebsstatus-Pseudowire_Attribut	false
/Auftrag/Netzelement/Interface/Pseudowire/@Betriebsstatus	Betriebsstatus-Pseudowire	false
/Auftrag/Netzelement/Interface/Tunnel/@Betriebsstatus	Betriebsstatus-Tunnel	false
/Auftrag/Netzelement/Interface/Beziehung_zu_Interface_uebergeordnet/@Betriebsstatus	Betriebsstatus-Beziehung_zu_Interface_uebergeordnet	false
/Auftrag/Netzelement/Interface/Beziehung_zu_Interface_untergeordnet/@Betriebsstatus	Betriebsstatus-Beziehung_zu_Interface_untergeordnet	false
/Auftrag/Netzelement/Interface/Beziehung_zu_QoS_Profil/@Betriebsstatus	Betriebsstatus-Beziehung_zu_QoS_Profil	false
/Auftrag/Netzelement/Interface/Beziehung_zu_VRF/@Betriebsstatus	Betriebsstatus-Beziehung_zu_VRF	false
/Auftrag/Netzelement/Interface/@Betriebsstatus	Betriebsstatus-Interface	true
/Auftrag/Netzelement/iBGP_Neighbors/@Betriebsstatus	Betriebsstatus-iBGP_Neighbors	false
/Auftrag/Netzelement/VRF/@Betriebsstatus	Betriebsstatus-VRF	false
/Auftrag/Netzelement/VRF/VRF_Attribut/@Betriebsstatus	Betriebsstatus-VRF_Attribut	false
/Auftrag/Netzelement/VRF/Route_Target_Import/@Betriebsstatus	Betriebsstatus-Route_Target_Import	false
/Auftrag/Netzelement/VRF/Route_Target_Export/@Betriebsstatus	Betriebsstatus-Route_Target_Export	false
/Auftrag/Netzelement/SNMP/@Betriebsstatus	Betriebsstatus-SNMP	false
/Auftrag/Netzelement/ACL/Listenposition/@Betriebsstatus	Betriebsstatus-Listenposition	false

/Auftrag/Netzelement/Doppelte_Anbindung/@Betriebsstatus	Betriebsstatus-Doppelte_Anbindung	false
/Auftrag/Netzelement/@Betriebsstatus	Betriebsstatus-Netzelement	true
/Auftrag/Netzaggregate/IPv4_Aggregat/@Betriebsstatus	Betriebsstatus-IPv4_Aggregat	false
/Auftrag/Netzaggregate/IPv6_Aggregat/@Betriebsstatus	Betriebsstatus-IPv6_Aggregat	false
/Auftrag/Verbindung/Verbindung_Attribut/@Betriebsstatus	Betriebsstatus-Verbindung_Attribut	false
/Auftrag/Verbindung/Beziehung_zu_Verbindung_uebergeordnet/@Betriebsstatus	Betriebsstatus-Beziehung_zu_Verbindung_uebergeordnet	false
/Auftrag/Verbindung/Beziehung_zu_Verbindung_untergeordnet/@Betriebsstatus	Betriebsstatus-Beziehung_zu_Verbindung_untergeordnet	false
/Auftrag/Verbindung/@Betriebsstatus	Betriebsstatus-Verbindung	true
/Auftrag/Netzelement/Interface/BGP/IPv4_BGP_session/Import_AS_Set/AS_Sets/@Betriebsstatus	Betriebsstatus-BGP/IPv4_BGP_session/Import_AS_Set/AS_Sets	false
/Auftrag/Netzelement/Interface/BGP/IPv6_BGP_session/Import_AS_Set/AS_Sets/@Betriebsstatus	Betriebsstatus-BGP/IPv6_BGP_session/Import_AS_Set/AS_Sets	false
/Auftrag/Netzelement/Interface/Generische_Parameter/@Betriebsstatus	Betriebsstatus-Interface/Generische_Parameter	false
/Auftrag/Generische_Parameter/@Betriebsstatus	Betriebsstatus-Generische_Parameter	false
/Auftrag/Verbindung/Verbindung_Attribut/@Typ	Verbindung_Attribut_Typ	false
/Auftrag/Netzelement/@NE_Funktion	NE_Funktion-Netzelement	true
/Auftrag/Netzelement/@Netzmanagement	Netzmanagement-Netzelement	true
/Auftrag/Netzelement/Hardware/@Hersteller	Hersteller-Hardware	true
/Auftrag/Netzelement/Interface/@IFC_Funktion	IFC_Funktion-Interface	true
/Auftrag/Netzelement/VRF/VRF_Attribut/@Typ	VRF_Attribut_Typ	false
/Auftrag/Netzelement/Interface/Interface_Attribut/@Typ	Interface_Attribut_Typ	false
/Auftrag/Netzelement/Interface/BGP/IPv4_BGP_session/BGP_Attribut/@Typ	BGP_Attribut_Typ-IPv4_BGP_session	false
/Auftrag/Netzelement/Interface/BGP/IPv6_BGP_session/BGP_Attribut/@Typ	BGP_Attribut_Typ-IPv6_BGP_session	false
/Auftrag/Netzelement/Interface/Virtual_Switch/Virtual_Switch_Attribut/@Typ	Virtual_Switch_Attribut_Typ-Virtual_Switch	false
/Auftrag/Netzelement/Interface/Pseudowire/@Typ	Pseudowire_Typ-Interface	false
/Auftrag/Netzelement/Interface/BGP/IPv4_BGP_session/@BGP_Variante	BGP_Variante-IPv4_BGP_session	false
/Auftrag/Netzelement/Interface/BGP/IPv6_BGP_session/@BGP_Variante	BGP_Variante-IPv6_BGP_session	false
/Auftrag/Verbindung/Beziehung_zu_Verbindung_uebergeordnet/@Rolle	Rolle-Beziehung_zu_Verbindung_uebergeordnet	false

/Auftrag/Verbindung/Beziehung_zu_Verbindung_untergeordnet/@Rolle	Rolle-Beziehung_zu_Verbindung_untergeordnet	false
/Auftrag/Netzelement/Interface/Beziehung_zu_Interface_ubergeordnet/@Rolle	Rolle-Beziehung_zu_Interface_ubergeordnet	false
/Auftrag/Netzelement/Interface/Beziehung_zu_Interface_untergeordnet/@Rolle	Rolle-Beziehung_zu_Interface_untergeordnet	false
/Auftrag/Netzelement/QoS/Profil/@Wiederverwendbar	Wiederverwendbar-QoS_Profil	false
/Auftrag/Netzelement/QoS/Klasse/@Wiederverwendbar	Wiederverwendbar-QoS_Klasse	false
/Auftrag/Netzelement/Interface/Beziehung_zu_VRF/NNI_Uebergang/@NNI_Klasse	NNI_Klasse-NNI_Uebergang	false
/Auftrag/Netzaggregate/IPv4_Aggregat/@Typ	IPv4_Aggregat_Typ-Netzaggregate	false
/Auftrag/Netzaggregate/IPv6_Aggregat/@Typ	IPv6_Aggregat_Typ-Netzaggregate	false
/Auftrag/Netzelement/QoS/Klasse/@Subklasse	Subklasse-Klasse	false
/Auftrag/Verbindung/@Wegeart	Wegeart-Verbindung	true
/Auftrag/Netzelement/Doppelte_Anbindung/@Wegeart	Wegeart-Doppelte_Anbindung	false
/Auftrag/@Auftrags_ID_SSt	Auftrags_ID_SSt	true
/Auftrag/@Storno_zu_Auftrags_ID_SSt	Storno_zu_Auftrags_ID_SSt	false
/Auftrag/Auftragsgruppe/Zur_Gruppe_gehoerende_Auftraege/@Auftrags_ID_OMS	Auftrags_ID_OMS-Zur_Gruppe_gehoerende_Auftraege	false
/Auftrag/@Auftrags_ID_OMS	Auftrags_ID_OMS	true
/Auftrag/Netzelement/Interface/Beziehung_zu_VRF/NNI_Uebergang/@NNI_Projektkurzbezeichnung	NNI_Projektkurzbezeichnung-NNI_Uebergang	false
/Auftrag/Netzelement/VRF/@Projekt_Kurz_Bezeichnung	Projekt_Kurz_Bezeichnung-VRF	false
/Auftrag/@Projekt_Kurzbezeichnung	Projekt_Kurzbezeichnung	true
/Auftrag/Auftragsgruppe/Zur_Gruppe_gehoerende_Auftraege/@Auftrag_Hostname	Auftrag_Hostname-Zur_Gruppe_gehoerende_Auftraege	false
/Auftrag/Netzelement/Interface/HSRP/@Hostname_Partner	Hostname_Partner-HSRP	false
/Auftrag/Netzelement/Interface/Virtual_Switch/Virtual_Switch_Gegenstelle/@Hostname_Gegenstelle	Hostname_Gegenstelle-Virtual_Switch_Gegenstelle	false
/Auftrag/Netzelement/Interface/Pseudowire/@Hostname_Gegenstelle	Hostname_Gegenstelle-Pseudowire	false
/Auftrag/Netzelement/Interface/Tunnel/Tunnel_Gegenstelle/@Hostname_Anschluss_Gegenstelle	Hostname_Anschluss_Gegenstelle-Tunnel_Gegenstelle	false
/Auftrag/Netzelement/Interface/Beziehung_zu_VRF/NNI_Uebergang/@NNI_PE_Hostname	NNI_PE_Hostname-NNI_Uebergang	false
/Auftrag/Netzelement/iBGP_Neighbors/@Neighbor_Hostname	Neighbor_Hostname-iBGP_Neighbors	false

/Auftrag/Netzelement/Doppelte_Anbindung/Partner_CPE/ @Hostname	Hostname-Partner_CPE	false
/Auftrag/Netzelement/@Hostname	Hostname-Netzelement	true
/Auftrag/Verbindung/Quelle/@Hostname	Hostname-Quelle	true
/Auftrag/Verbindung/Senke/@Hostname	Hostname-Senke	true
/Auftrag/@Auftrag_Hostname	Auftrag_Hostname	true
/Auftrag/Netzelement/Interface/HSRP/@Domainname_Par tner	Domainname_Partner- HSRP	false
/Auftrag/Netzelement/Interface/Virtual_Switch/Virtual_Swi tch_Gegenstelle/@Domainname_Gegenstelle	Domainname_Gegenstelle- Virtual_Switch_Gegenstelle	false
/Auftrag/Netzelement/Interface/Pseudowire/@Domainna me_Gegenstelle	Domainname_Gegenstelle- Pseudowire	false
/Auftrag/Netzelement/Interface/Tunnel/Tunnel_Gegenstell e/@Domainname_Anschluss_Gegenstelle	Domainname_Anschluss_G egenstelle- Tunnel_Gegenstelle	false
/Auftrag/Netzelement/Interface/Beziehung_zu_VRF/NNI_U ebergang/@NNI_PE_Domainname	NNI_PE_Domainname- NNI_Uebergang	false
/Auftrag/Netzelement/iBGP_Neighbors/@Neighbor_Domai nname	Neighbor_Domainname- iBGP_Neighbors	false
/Auftrag/Netzelement/Doppelte_Anbindung/Partner_CPE/ @Domainname	Domainname-Partner_CPE	false
/Auftrag/Netzelement/@Domainname	Domainname-Netzelement	true
/Auftrag/Verbindung/Quelle/@Domainname	Domainname-Quelle	true
/Auftrag/Verbindung/Senke/@Domainname	Domainname-Senke	true
/Auftrag/Netzelement/Interface/HSRP/@FQDN_Partner	FQDN_Partner-HSRP	false
/Auftrag/Netzelement/Interface/Virtual_Switch/Virtual_Swi tch_Gegenstelle/@FQDN_Gegenstelle	FQDN_Gegenstelle- Virtual_Switch_Gegenstell e	false
/Auftrag/Netzelement/Interface/Pseudowire/@FQDN_Geg enstelle	FQDN_Gegenstelle- Pseudowire	false
/Auftrag/Netzelement/Interface/Tunnel/Tunnel_Gegenstell e/@FQDN_Anschluss_Gegenstelle	FQDN_Anschluss_Gegenst elle-Tunnel_Gegenstelle	false
/Auftrag/Netzelement/Interface/Beziehung_zu_VRF/NNI_U ebergang/@NNI_PE_FQDN	NNI_PE_FQDN- NNI_Uebergang	false
/Auftrag/Netzelement/iBGP_Neighbors/@Neighbor_FQDN	Neighbor_FQDN- iBGP_Neighbors	false
/Auftrag/Netzelement/Doppelte_Anbindung/Partner_CPE/ @FQDN	FQDN-Partner_CPE	false
/Auftrag/Netzelement/@FQDN	FQDN-Netzelement	true
/Auftrag/Verbindung/Quelle/@FQDN	FQDN-Quelle	true
/Auftrag/Verbindung/Senke/@FQDN	FQDN-Senke	true
/Auftrag/Netzelement/@Netzabschnitt	Netzabschnitt- Netzelement	false
/Auftrag/Netzelement/Interface/QoS/Klasse/IPv4_Filter/@F ilter_Makro	Filter_Makro-IPv4_Filter	false
/Auftrag/Netzelement/Interface/QoS/Klasse/IPv6_Filter/@F ilter_Makro	Filter_Makro-IPv6_Filter	false
/Auftrag/Netzelement/Interface/Statische_Route/IPv4_Rou te/@Metrik	Metrik-IPv4_Route	false
/Auftrag/Netzelement/Interface/Statische_Route/IPv6_Rou te/@Metrik	Metrik-IPv6_Route	false

/Auftrag/Auftragsgruppe/Zur_Gruppe_gehoerende_Auftraege/@Projekt_Nummer	Projekt_Nummer-Zur_Gruppe_gehoerende_Auftraege	false
/Auftrag/Netzelement/Interface/Beziehung_zu_VRF/NNI_Uebergang/@NNI_Projektnummer	NNI_Projektnummer-NNI_Uebergang	false
/Auftrag/@Projekt_Nummer	Projekt_Nummer	true
/Auftrag/Netzelement/@Routercode	Routercode-Netzelement	false
/Auftrag/Netzelement/Interface/IPv6_Adresse/@IP_Adresse	IPv6-Adresse-Interface	false
/Auftrag/Netzelement/Interface/HSRP/@virtuelle_IP_Adresse	IPv6-Adresse-HSRP-virtuelle_IP_Adresse	false
/Auftrag/Netzelement/Interface/HSRP/@Ifc_IP_Adresse	IPv6-Adresse-HSRP-Ifc_IP_Adresse	false
/Auftrag/Netzelement/Interface/Statische_Route/IPv6_Route/@IP_Netz	IPv6-Adresse-IPv6_Route-Statische_Route	false
/Auftrag/Netzelement/@Erreichbarkeit_IPv6_Adresse	IPv6-Adresse-Erreichbarkeit_IPv6_Adresse-Netzelement	false
/Auftrag/Netzelement/Interface/BGP/IPv6_BGP_session/@Local_IP	IPv6-Adresse-Local_IP-IPv6_BGP_session	false
/Auftrag/Netzelement/Interface/BGP/IPv6_BGP_session/@Remote_IP	IPv6-Adresse-Remote_IP-IPv6_BGP_session	false
/Auftrag/Netzelement/Interface/Virtual_Switch/Virtual_Switch_Gegenstelle/@Loopback0_Gegenstelle_IPv6	IPv6-Adresse-Loopback0_Gegenstelle_IPv6 - Virtual_Switch_Gegenstelle	false
/Auftrag/Netzelement/Interface/Pseudowire/@Loopback0_Gegenstelle_IPv6	IPv6-Adresse - Loopback0_Gegenstelle_IPv6 - Pseudowire	false
/Auftrag/Netzelement/Interface/Beziehung_zu_VRF/NNI_Uebergang/@NNI_Ifc_IPv6_Adresse	IPv6-Adresse - NNI_Ifc_IPv6_Adresse - Beziehung_zu_VRF	false
/Auftrag/Netzelement/QoS/Klasse/IPv6_Filter/@Quell_IP	IPv6-Adresse - Quell_IP - Klasse	false
/Auftrag/Netzelement/QoS/Klasse/IPv6_Filter/@Ziel_IP	IPv6-Adresse - Ziel_IP - Klasse	false
/Auftrag/Netzelement/SNMP/IPv6_Accesslist/@IP_Netz	IPv6-Adresse - IP_Netz - SNMP	false
/Auftrag/Netzelement/Doppelte_Anbindung/Partner_CPE/@Loopback1_IPv6	IPv6-Adresse - Loopback1_IPv6 - Doppelte_Anbindung	false
/Auftrag/Netzelement/Doppelte_Anbindung/Cross_LAN/@Cross_LAN_IPv6_Partner	IPv6-Adresse - Cross_LAN_IPv6_Partner - Doppelte_Anbindung	false
/Auftrag/Netzelement/@Management_IPv6_Adresse	IPv6-Adresse - Management_IPv6_Adresse - Netzelement	false
/Auftrag/Netzaggregate/IPv6_Aggregat/@IP_Netz	IPv6-Adresse - IP_Netz-IPv6_Aggregat	false
/Auftrag/@Auftrag_Anschlusskurzname	Auftrag_Anschlusskurzname	true

/Auftrag/Netzelement/Interface/Beziehung_zu_ACL/@Betriebsstatus	Betriebsstatus-Beziehung_zu_ACL	false
/Auftrag/Netzelement/ACL/@Betriebsstatus	Betriebsstatus-ACL	false
/Auftrag/Netzelement/Interface/Statische_Route/IPv4_Route/Route_Attribut/@Betriebsstatus	Betriebsstatus-Route_Attribut_IPv4	false
/Auftrag/Netzelement/Interface/Statische_Route/IPv6_Route/Route_Attribut/@Betriebsstatus	Betriebsstatus-Route_Attribut_IPv6	false
/Auftrag/Netzelement/Interface/Statische_Route/IPv4_Route/Route_Attribut/@Typ	Typ-Route_Attribut_IPv4	false
/Auftrag/Netzelement/Interface/Statische_Route/IPv6_Route/Route_Attribut/@Typ	Typ-Route_Attribut_IPv6	false
/Auftrag/@OMS_Prozess	OMS_Prozess	true

## 9 Literaturverzeichnis

- atlassian - Was ist Scrum?* (kein Datum). Abgerufen am 18. August 2022 von <https://www.atlassian.com/de/agile/scrum#:~:text=Scrum%20wird%20oft%20als%20ein,und%20Managen%20der%20Teamarbeit%20unterst%C3%BCtzen.>
- data2type - XPath – Einführung in die Technik des Zugriffs auf die XML-Strukturen.* (2007). Abgerufen am 19. Juni 2022 von <https://www.data2type.de/xml-xslt-xslfo/xpath/xpath-einfuehrung>
- github -* (kein Datum). Abgerufen am 18. August 2022 von <https://github.com/GIP-SmartMercial/>
- Grupe, W. (2018). *XML Grundlagen | Technologie | Validierung | Auswertung*. Frechen: mitp Verlags GmbH & Co.KG.
- homepage-webhilfe - Komplexe Typen.* (kein Datum). Abgerufen am 10. Juni 2022 von <https://www.homepage-webhilfe.de/XML/XSD/komplexetypen.php>
- homepage-webhilfe- Einfache Typen.* (kein Datum). Abgerufen am 08. Juli 2022 von <https://www.homepage-webhilfe.de/XML/XSD/einfachetypen.php>
- homepage-webhilfe. - Einschränkungen.* (kein Datum). Abgerufen am 15. April 2022 von <https://www.homepage-webhilfe.de/XML/XSD/einschraenkungen.php>
- ibm - XPath overview.* (29. Juli 2022). Abgerufen am 12. August 2022 von <https://www.ibm.com/docs/en/app-connect/11.0.0?topic=xpath-overview>
- ionos - XPath-Tutorial für Einsteiger.* (28. September 2020). Abgerufen am 12. August 2022 von <https://www.ionos.de/digitalguide/websites/web-entwicklung/xpath-tutorial/>
- liquid-technologies - XML Tutorials.* (kein Datum). Abgerufen am 12. August 2022 von <https://www.liquid-technologies.com/tutorials>
- pagina - DEKLARATION VON ATTRIBUTEN IN XSD.* (kein Datum). Abgerufen am 16. April 2022 von <https://www.pagina.gmbh/xml-hintergruende/pagina-das-kompendium/themenkomplex-ii-xml/xml-schema-als-dtd-alternative/deklaration-von-attributen-in-xsd/>
- regexe.* (2017). Abgerufen am 23. Mai 2022 von <http://regexe.de/hilfe.jsp>
- tutorialspoint - XML - Schemas.* (kein Datum). Abgerufen am 04. Juli 2022 von [https://www.tutorialspoint.com/de/xml/xml\\_schemas.htm#:~:text=XML%2DSchema%20wir d%20allgemein%20als,Schema%2DElement%20unterst%C3%BCtzt%20Namespaces.](https://www.tutorialspoint.com/de/xml/xml_schemas.htm#:~:text=XML%2DSchema%20wir d%20allgemein%20als,Schema%2DElement%20unterst%C3%BCtzt%20Namespaces.)
- W3C.* (kein Datum). Abgerufen am 13. April 2022 von [https://www.lgln.niedersachsen.de/startseite/online\\_angebote\\_services/lgln\\_lexikon/w/w3c-190566.html](https://www.lgln.niedersachsen.de/startseite/online_angebote_services/lgln_lexikon/w/w3c-190566.html)
- w3schools - XML Schema sequence Element.* (kein Datum). Abgerufen am 10. Juni 2022 von [https://www.w3schools.com/xml/el\\_sequence.asp](https://www.w3schools.com/xml/el_sequence.asp)
- w3schools - XPath Syntax.* (kein Datum). Abgerufen am 28. Juli 2022 von [https://www.w3schools.com/xml/xpath\\_syntax.asp](https://www.w3schools.com/xml/xpath_syntax.asp)
- Weber, C. (2002/03). *XML-Grundlagen*. Abgerufen am 17. Mai 2022 von <http://www.lgis.informatik.uni-kl.de/archiv/wwwdvs.informatik.uni-kl.de/courses/seminar/WS0203/ausarbeitung1.pdf>

*XML - Kommentare.* (kein Datum). Abgerufen am 21.. Mai 2022 von  
[https://www.tutorialspoint.com/de/xml/xml\\_comments.htm](https://www.tutorialspoint.com/de/xml/xml_comments.htm)

*xplore-dna - IP-Adressen.* (15. Dezember 2021). Abgerufen am 25. August 2022 von  
<https://www.xplore-dna.net/mod/page/view.php?id=1298>

*XSD (XML Schema Definition).* (kein Datum). Abgerufen am 19.. Mai 2022 von <https://www.torsten-horn.de/techdocs/java-xsd.htm>

*xyna-factory.* (2022). Abgerufen am 12.. April 2022 von <https://www.xyna.com/xyna-factory/>

## 10 Selbstständigkeitserklärung

Hiermit erkläre ich, dass die von mir an der Hochschule für Technik, Wirtschaft und Kultur Leipzig eingereichte Abschlussarbeit zum Thema

### **Umsetzung der Auftragsprüfung in GeKo**

vollkommen selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht. Die Abbildungen in dieser Arbeit sind von mir selbst erstellt oder mit einem entsprechenden Quellennachweis versehen. Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Hochschule/Universität eingereicht worden.

Ulm, den 25. August 2022

.....

## Bestätigung des Praxisbetriebs

Hiermit wird bestätigt, dass ich die vorliegende Arbeit zur Kenntnis genommen habe.

Der/Die Studierende hat die Projektarbeit selbstständig im Rahmen seiner Praxistätigkeit erarbeitet, sich dabei in angemessenem Maße persönlich engagiert und eingebracht und die Zwischenergebnisse stets mit mir abgestimmt.

Über das Projekt wurde ein Tätigkeitsnachweis geführt und vom Praxisbetreuer freigegeben. In diesem ist mindestens der laut Modulbeschreibung erforderliche Arbeitsumfang dokumentiert (200 Stunden in den Projekten I, II und III sowie 436 Stunden im Praxisprojekt).

Aus unserer Sicht erfüllt die Arbeit damit die betrieblichen Anforderungen.

(Bitte ankreuzen, falls zutreffend)

- Ich habe Ihnen weitere Hinweise per E-Mail zugesandt.
- Ein gemeinsamer Termin, bspw. für eine Präsentation der Projektergebnisse wird gewünscht.
- Sonstige Hinweise:

Kontaktdaten Praxisbetreuer*in	
<i>Name:</i>	Heidrich, Mario
<i>Firmenanschrift:</i>	Olgastraße 67, 89073 Ulm
<i>Telefon:</i>	+49 731 10034651
<i>E-Mail:</i>	mario.heidrich@telekom.de

.....

Ort, Datum

Unterschrift Praxisbetreuer\*in

<b>Bachelorstudiengang</b> <input checked="" type="checkbox"/> Informations- und Kommunikationstechnik (IKB) <input type="checkbox"/> Telekommunikationsinformatik (TIB)		 <b>HIT</b> Hochschule für Wirtschaft und	
<b>TÄTIGKEITSNACHWEIS</b> zu einem praktischen Studienabschnitt			
<b>Name, Vorname:</b>	<b>Häußler, Niels Fabien</b>		
<b>Matrikelnummer:</b>	<b><u>79153</u></b>		
<b>Seminargruppe:</b>	<b>21IKB</b>		
<b>Praxisstelle:</b>	<b>Deutsche Telekom Technik GmbH - Chapter Operator IP Backbone 2</b>		
<b>Modulnummer:</b>		<b>Modulname:</b>	
<b>Praxisanleiter</b>	Name	<b>Heidrich, Mario</b>	
	höchster akademischer oder Berufsabschluss	<b>Dipl. Ing. (FH) Technische Informatik</b>	
	E-Mail	<b><u>mario.heidrich@telekom.de</u></b>	
	Telefon	<b>+49 731 10034651</b>	
<b>Verlaufsdarstellung</b>			
Zeitraum (von – bis)	Umfang des praktischen Einsatzes in Stunden	Praxiseinrichtung	Beschreibung der Tätigkeiten und Aufgabenstellung
07-21.03	35		Erlernen der Grundlagen von XML/XSD + Regel/Validierungen GeKo
16.03-1.04	60	T-CNF-IPC NNA KIC	Analyse XSD + XPath herausarbeiten
25.-29.04	20	T-CNF-IPC NNA KIC	Implementierung Regel/Validierung in GeKo
02.04.05	15	T-CNF-IPC NNA KIC	Testen (Manipulation XML-Aufträge)
27-29.07	15	T-CNF-IPC NNA KIC	Update/Verbesserung Regel/Validierungen + Testen
21.03-15.08	75		Schreiben der Projektarbeit
<b>Summe:</b>	<b>220</b>		

Hiermit wird bestätigt, dass der praktische Studienabschnitt wie oben ausgewiesen absolviert wurde.

Bemerkungen der Praxisstelle:

Name und Funktion in der Praxisstelle:

Datum:

Unterschrift