



Mobilitäts Daten Marktplatz

# **Containerformat Spezifikation**

**Version 1.1 - 21.02.2014**

# Inhaltsverzeichnis

<b>0</b>	<b>Einführung .....</b>	<b>4</b>
0.1	Referenzierte Dokumente .....	4
0.2	Abkürzungen .....	4
<b>1</b>	<b>Containerformat.....</b>	<b>5</b>
1.1	Aufbau des Container-Headers .....	6
1.2	Aufbau des Container-Body .....	9
1.3	XML-Beispiel .....	12

# Tabellenverzeichnis

Tabelle 1: Strukturinformationen im Containermodell.....	9
Tabelle 2: Ausprägungen type-Attribut V1.0 .....	10

# Abbildungsverzeichnis

Abbildung 1: Übersicht Containermodell .....	5
Abbildung 2: Containermodell - Aufbau header Element .....	6
Abbildung 3: Aufbau Timestamp-Element .....	7
Abbildung 4: Aufbau des Signature-Elementes .....	8
Abbildung 5: Aufbau des xml-Elementes.....	9
Abbildung 6: Aufbau des binary-Elementes .....	10
Abbildung 7: Containermodell - Aufbau body Element .....	11

# 0 Einführung

## 0.1 Referenzierte Dokumente

[Quelle]	Herausgeber
[DatexIISchema]	Schema der Datex II V2.0 Spezifikation
[WS Security Core Specification 1.1]	OASIS Standard 1.1: WS-Security Core Specification 1.1 <a href="http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf">http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf</a>

## 0.2 Abkürzungen

Abkürzung	Beschreibung
MDM	Mobilitätsdaten-Marktplatz
MDM-Plattform	Mobilitätsdaten-Marktplatz-Plattform
XML	Extensible Markup Language
XSD	XML Schema Definition

# 1 Containerformat

Das Containerformat des Mobilitäts Daten Marktplatzes ist ein XML-Datenaustauschformat, das mittels einer XML Schema Definition (XSD) definiert ist. Es unterteilt sich in den Bereich der MDM-Plattform spezifischen Angaben (header-Element) und in den Nutzdatenbereich (body-Element). Diese Formatdefinition erlaubt es, beliebig strukturierte Verkehrsdaten über die MDM-Plattform zu transportieren.

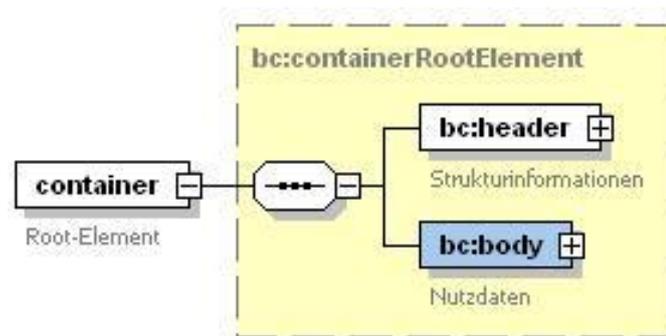


Abbildung 1: Übersicht Containermodell

Hierzu enthält das Containerformat im header-Element je nach Verwendungsart die Publikations-ID der MDM-Plattform, unter der die enthaltenen Daten der MDM-Plattform zur Verfügung gestellt werden, bzw. die Subskriptions-ID der MDM-Plattform aus der die Daten stammen. Des Weiteren kann das Header-Element optional für enthaltene Nutzdaten digitale Signaturen und Zeitstempel enthalten. Im Containerformat ist genau ein header-Element zulässig.

Die Nutzdaten werden im body-Element des Containers abgelegt. Um das Modell flexibel zu halten, werden Format und Inhalt des body Elements nicht vorgegeben. Die Kinder des Body Elements enthalten eine zusätzliche Kennzeichnung des Inhaltstyps. In einem Container gibt es genau ein body-Element.

So können nicht nur Daten im XML-Format im Container transportiert werden, sondern auch Binärdaten. Des Weiteren ist es möglich, über dieses Modell mehr als nur ein Datenpaket (d. h. mehrere Kinder im Body-Element) in einem Container zu versenden.

## 1.1 Aufbau des Container-Headers

Im header Element werden Metadaten zu den übermittelten Nutzdaten gespeichert. Der Aufbau dieser Informationen orientiert sich an der [WS Security Core Specification 1.1] für den Aufbau eines SOAP Headers.

Innerhalb des Headers sind Elemente definiert, die Informationen über den Ursprung der Daten, die Gültigkeit der Nutzdaten und die Signatur der Nutzdaten enthalten. Zusätzlich kann ein Statuscode übermittelt werden.

Der Ursprung einer Nachricht wird über die Publikations-ID, unter der die Publikation des Datengebers registriert ist, beziehungsweise die Subskriptions-ID, unter der die Subskription des Datennehmers registriert ist, gekennzeichnet.

Die Elemente zur Gültigkeit und zur Signatur der Datenpakete können 0 bis n mal vorkommen. Sind im body-Bereich mehrere Datenpakete untergebracht, kann jedes einzelne Paket mit einer Gültigkeit und einer Signatur versehen werden. Detaillierte Angaben hierzu sind in Kapitel 1.2 nachzulesen.

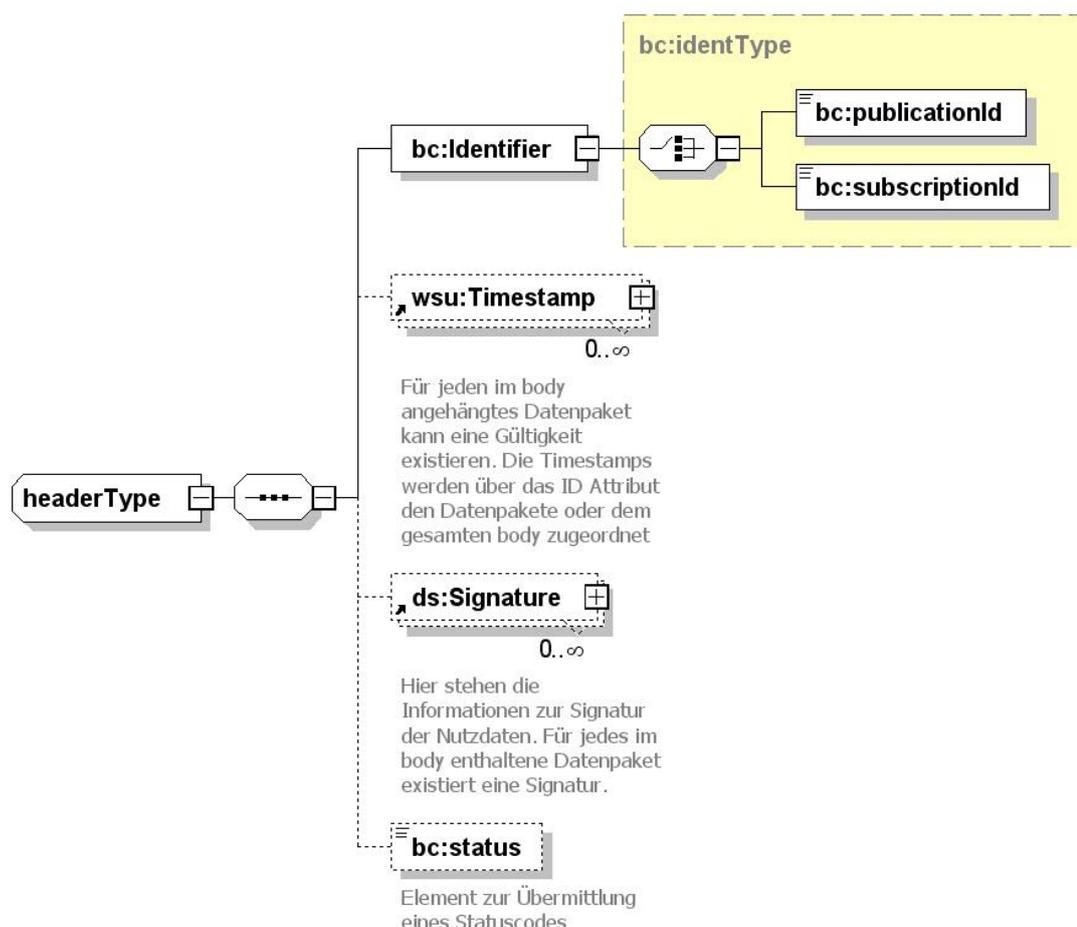


Abbildung 2: Containermodell - Aufbau header Element

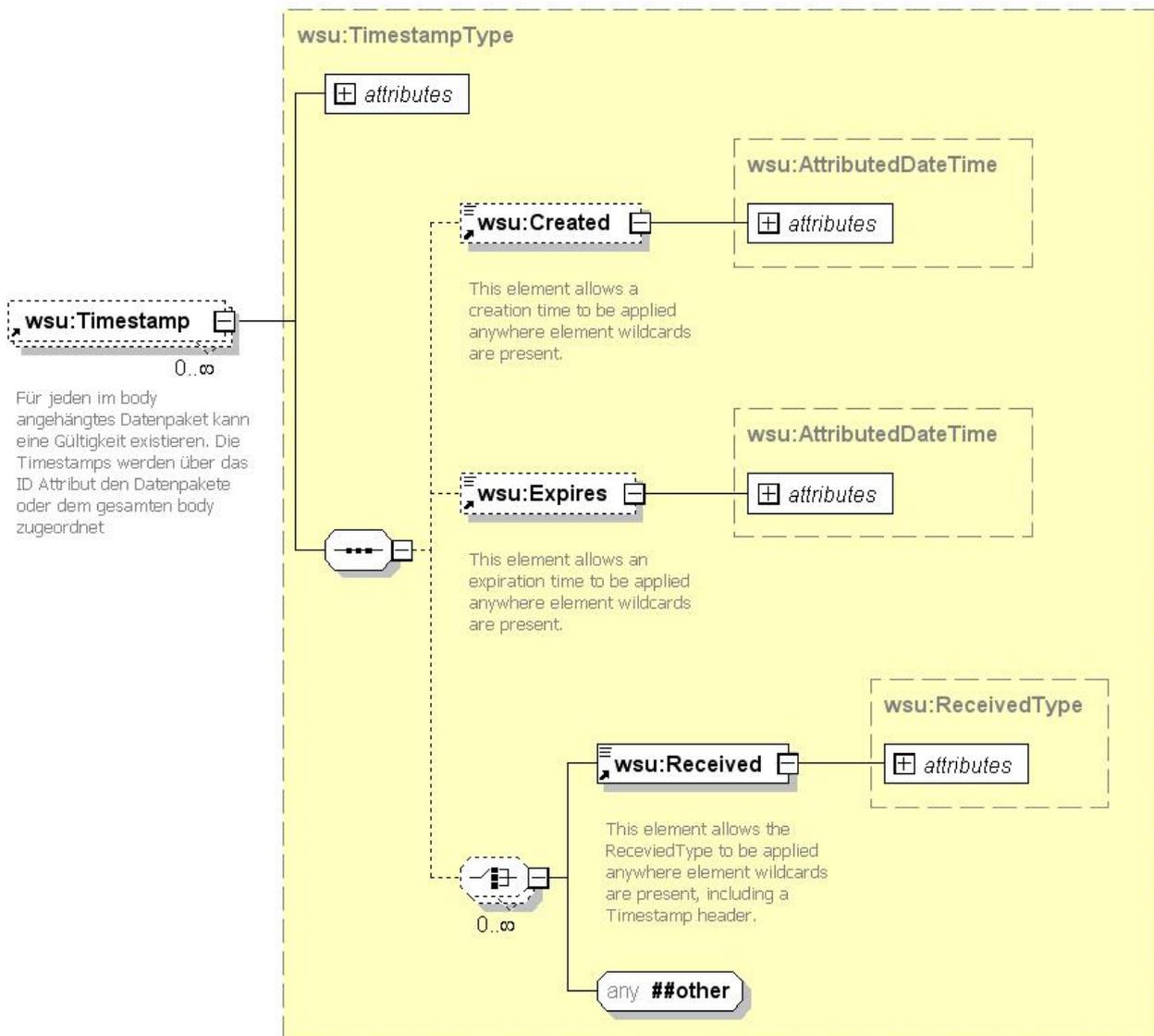


Abbildung 3: Aufbau Timestamp-Element

Das Timestamp-Element ist Bestandteil der OASIS Spezifikation [WS Security Core Specification 1.1] des OASIS Standards 1.1 und definiert den Zeitpunkt der Erstellung sowie den Gültigkeitszeitraum von Daten.

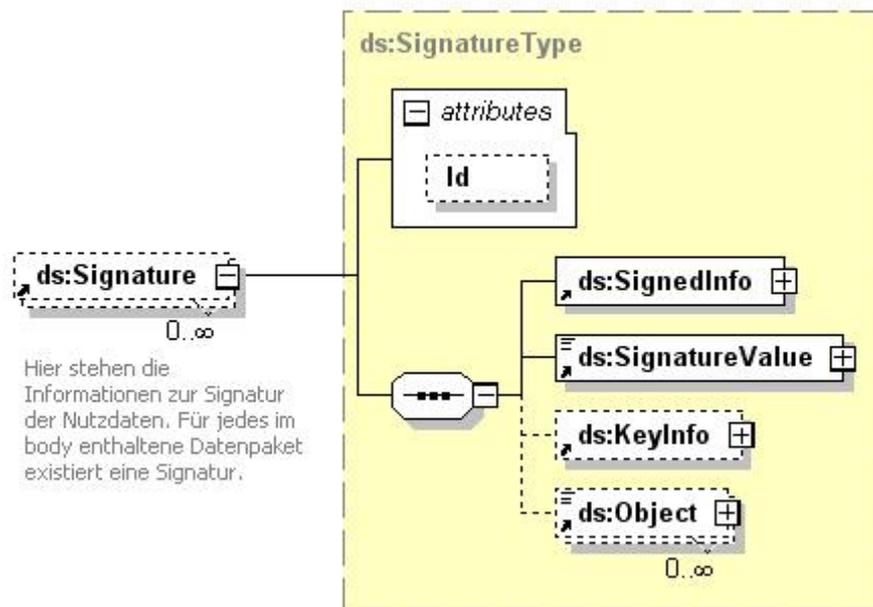


Abbildung 4: Aufbau des Signature-Elementes

Das Signature-Element ist ebenfalls Bestandteil der OASIS Spezifikation [WS Security Core Specification 1.1] des OASIS Standards 1.1 und definiert den Urheber der Nutzdaten. Darüber hinaus stellt die Signatur sicher, dass die Nutzdaten nicht manipuliert wurden (Datenintegrität).

Als Statuscode ist, falls das Element vorhanden ist, in der Version 1.0 nur der Wert "OK" zulässig. Weitere Statuscodes werden ggf. in zukünftigen Versionen ergänzt.

Eine Verschlüsselung des Nachrichteninhaltes ist nicht vorgesehen, da die Nachricht über den gesamten Transportweg SSL-verschlüsselt übertragen wird. Eine asymmetrische Verschlüsselung des Nachrichteninhaltes ist darüber hinaus nicht sinnvoll, da sie erfordern würde, dass der Datengeber beim Erstellen und Senden eines Datenpaketes zur Plattform für alle Datennehmer deren öffentliche Schlüssel ermitteln kann sowie die MDM-Plattform das erforderliche Schlüsselmanagement übernimmt. Innerhalb der MDM-Plattform darf davon ausgegangen werden, dass die Daten nur autorisierten Datennehmern zur Verfügung gestellt werden.

Information	Beschreibung
Identifizier	Enthält Angaben zur Identifizierung des Herkunft der Daten <ul style="list-style-type: none"> <li>• Publikations-ID</li> <li>• Subskriptions-ID</li> </ul>
Gültigkeit	Informationen über die Gültigkeitsdauer der Daten <ul style="list-style-type: none"> <li>▪ Zeitpunkt des Erstellens</li> <li>▪ Zeitpunkt des Verfallens</li> <li>▪ Referenz auf die Daten, für die diese Gültigkeit bestimmt ist</li> </ul>
Signatur	Informationen über die Signatur der Daten <ul style="list-style-type: none"> <li>▪ Art der Signatur</li> <li>▪ Referenz auf die signierten Daten</li> </ul>
Status	Zur Übermittlung eines Statuscodes

Tabelle 1: Strukturinformationen im Containermodell

## 1.2 Aufbau des Container-Body

In einem Container gibt es genau ein body Element. Unterhalb dieses Elements können beliebig viele xml- und binary-Elemente gruppiert werden, in denen die eigentlichen Nutzdaten untergebracht sind.

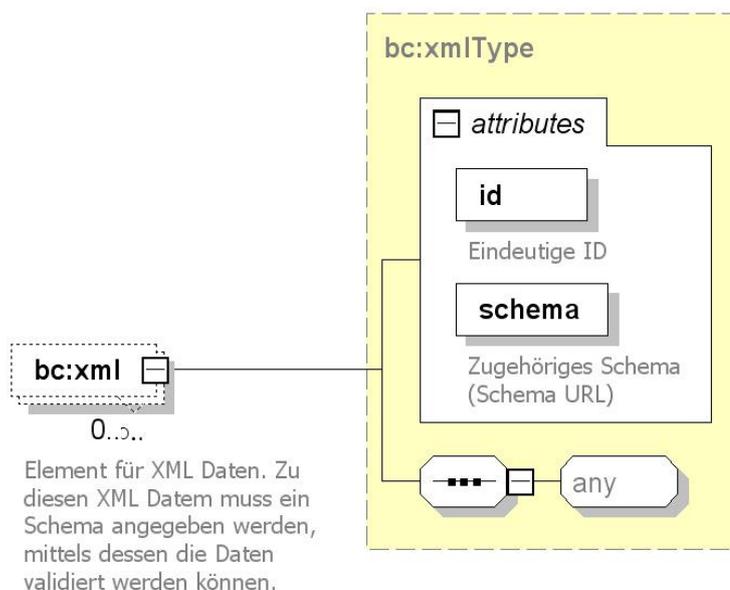


Abbildung 5: Aufbau des xml-Elementes

- Liegen die Nutzdaten in einem auf XML basierendem Format vor, so werden die Nutzdaten unterhalb des *xml*-Elements untergebracht. Das Attribut *schema* verweist dabei auf ein zur XML-Struktur gehöriges XSD-Schema, welches zur Validierung der Daten herangezogen werden kann. Das Attribut *id* weist den Nutzdaten eine eindeutige ID zu, die von den Header-Elementen *Timestamp* und *Signature* referenziert wird.

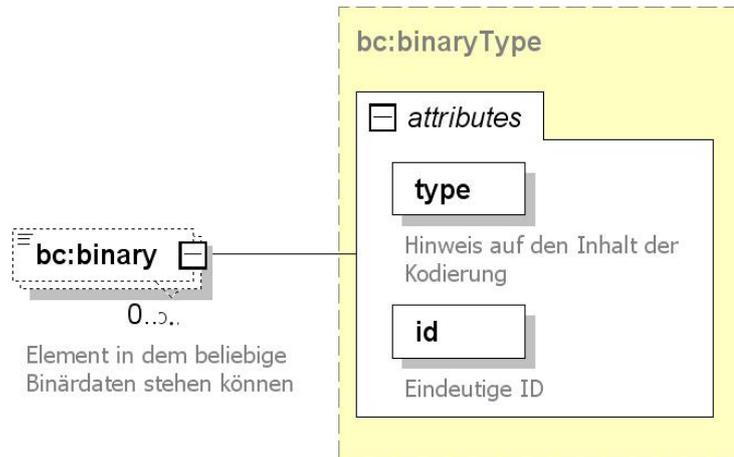


Abbildung 6: Aufbau des binary-Elementes

- Liegen die Nutzdaten als Binärdaten vor, werden diese unterhalb des `binary`-Elements in BASE64-kodierter Form abgelegt. Das Attribut `type` kennzeichnet die Art der Daten. Das Attribut `id` weist den Nutzdaten eine eindeutige Id zu, die von den Header-Elementen `Timestamp` und `Signature` referenziert wird.

type	Beschreibung
base64BinaryDatex2	Base64 kodierte Daten gemäß [DatexIISchema].
base64BinaryOTS2	Base64 kodierte Daten gemäß Open Transport System 2.
hexBinary	Binärdaten in hexadezimaler Schreibweise. Die Art der Daten muss Bestandteil der kodierten Daten sein.
Todo	Platzhalter für weitere Formate

Tabelle 2: Ausprägungen type-Attribut V1.0

Zu jedem der `xml`- und `binary`-Elemente können eine Signatur und eine Gültigkeit gehören, die im header Element definiert sind. Dazu wird dem entsprechenden `xml`- / `binary`-Element eine ID zugeordnet, auf die im `Signature` / `Timestamp` Element des Headers verwiesen wird.

Soll der gesamte Inhalt des *body*-Elements mit einer Signatur oder einer Gültigkeit versehen werden, muss dies durch Verwendung der ID "body" in den Header-Elementen codiert werden. Dem *body*-Element selbst darf dann keine ID zugewiesen werden.

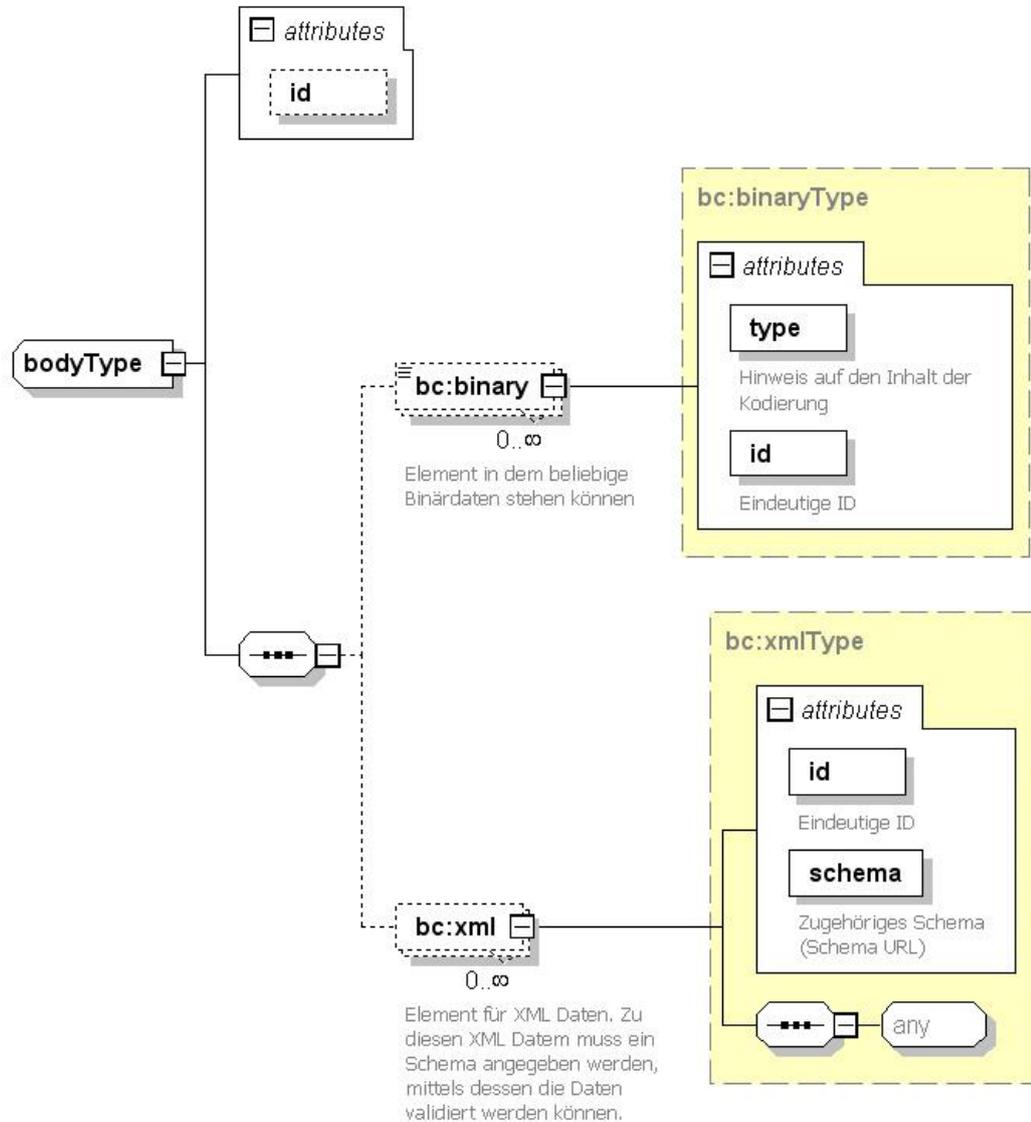


Abbildung 7: Containermodell - Aufbau body Element

## 1.3 XML-Beispiel

Das Beispiel zeigt einen Container mit zwei Datenpaketen im Body und einem Header mit Zeitstempel für den gesamten Body. Alle Informationen sind fiktiv.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:container xmlns:ns1="http://schemas.xmlsoap.org/ws/2002/07/utility"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#"
xmlns:ns3="http://ws.bast.de/container/TrafficDataService">
  <ns3:header>
    <ns3:Identifizier>
      <ns3:publicationId>1734007</ns3:publicationId>
    </ns3:Identifizier>
    <ns1:Timestamp ns1:Id="body">
      <ns1:Created>2014-01-20T12:28:01.874Z</ns1:Created>
      <ns1:Expires>2014-01-27T12:28:01.874Z</ns1:Expires>
    </ns1:Timestamp>
  </ns3:header>
  <ns3:body>
    <ns3:binary type="base64BinaryDatex2" id="B1">
H4sIAAAAAAAAAAN19245dx5HluwH/A8EXPwxCjMiMvBVkNF1604A1Y0xrBvM2oMwymmiKNCi6bf/9&#xD;
RBYtdW225e4d3EzmHMOWi6oL1z61TtxjxZf/80fvXz351/u3P7x88/rXv6Iv8FdP719/++bFy9ff&#xD;
/fpX/+ubAFFX//DVL3/x5Qvzmzffvfz2+auv37y4f/VEvuv1D79++s/v3v3h7tmz79/98C//+Pr3&#xD;
b95+//yd/JxnP3z7z/ffP3+GBPO/+PSXv3j/DXd//uH1T9/0pz/96Ys/2S/evP3umUGKZ//n69/8&#xD;
08P3wcvXP7x7/vrbe/k++Ya79z/tN2++ffjh/9m/9Mn44/2r3759+e39b//4u1cv33
    </ns3:binary>
    <ns3:binary type="base64BinaryDatex2" id="B2">
RBYtdW225e4d3EzmHMOWi6oL1z61TtxjxZf/80fvXz351/u3P7x88/rXv6Iv8FdP719/++bFy9ff&#xD;
H4sIAAAAAAAAAAN19245dx5HluwH/A8EXPwxCjMiMvBVkNF1604A1Y0xrBvM2oMwymmiKNCi6bf/9&#xD;
/fpX/+ubAFFX//DVL3/x5Qvzmzffvfz2+auv37y4f/VEvuv1D79++s/v3v3h7tmz79/98C//+Pr3&#xD;
08P3wcvXP7x7/vrbe/k++Ya79z/tN2++ffjh/9m/9Mn44/2r3759+e39b//4u1cv33/z//3XHz/9&#xD;
b95+//yd/JxnP3z7z/ffP3+GBPO/+PSXv3j/DXd//uH1T9/0pz/96Ys
    </ns3:binary>
  </ns3:body>
</ns3:container>
```