Container Format Specification

Version 1.2 - 27.05.2015

Table of contents

List of tables

List of figures

1 Introduction

1.1 Referenced documents

[Source]	Publisher
[DatexIISchema]	Schema of Datex II V2.0 specification
[WS Security Core Specification 1.1]	OASIS Standard 1.1: WS-Security Core Specification 1.1 http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf

1.2 Abbreviations

Abbreviation	Description
MDM	Mobility Data Marketplace
MDM Platform	Mobility Data Marketplace platform
XML	eXtensible Markup Language
XSD	XML Schema Definition

2 Container format

The container format of the mobility data marketplace is an XML data exchange format that is defined using an XML Schema Definition (XSD). It is divided into the area of MDM platform-specific information (header element) and the payload area (body element). This format definition allows to carry any structured traffic data by using the MDM platform.

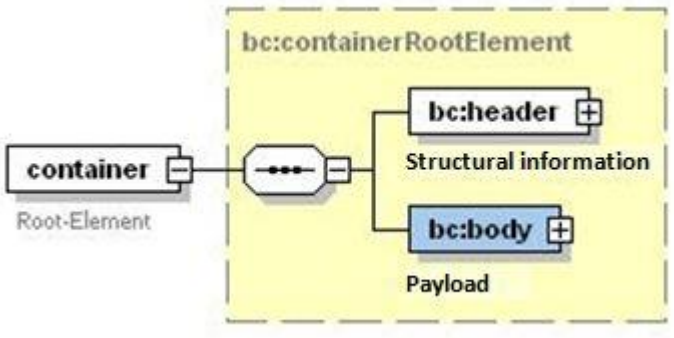

Figure 1: Overview Container Model

In the header element the container format includes - depending on the type of use - the ID of the publication under which the data is made available to the MDM platform, or the ID of the subscription under which the data are sent to the data client. Furthermore, the header element optionally contains digital signatures and timestamps for the included payload. In the container format, exactly one header element is permitted.

The payload is stored in the body element of the container. In order to keep the model flexible, the format and content of the body element is not specified. The children of the body element contain an additional mark of the content type. In a container, there is exactly one body element.

Thus, not only data in XML format can be transported in containers, but also binary data. It is also possible to transport more than just one data packet (i.e. several children in the body element) in a container using this model.

2.1 Construction of the container header

The metadata for the transmitted payload is stored in the header element. The structure of this information is based on the [WS Security Core Specification 1.1] for the construction of a SOAP header.

The header includes elements consisting of information about the data origin, the validity and the signature of the payload. In addition, a status code can be transmitted.

The origin of a message is characterized by the publication ID under which the publication of the data supplier is registered, or the subscription ID under which the subscription of the data client is registered.

The elements on the validity and the signature of the data packets may occur 0 to n times. If multiple data packets are contained in the body, each package can then be provided with a validity and a signature. For detailed information, please refer to chapter 2.2.

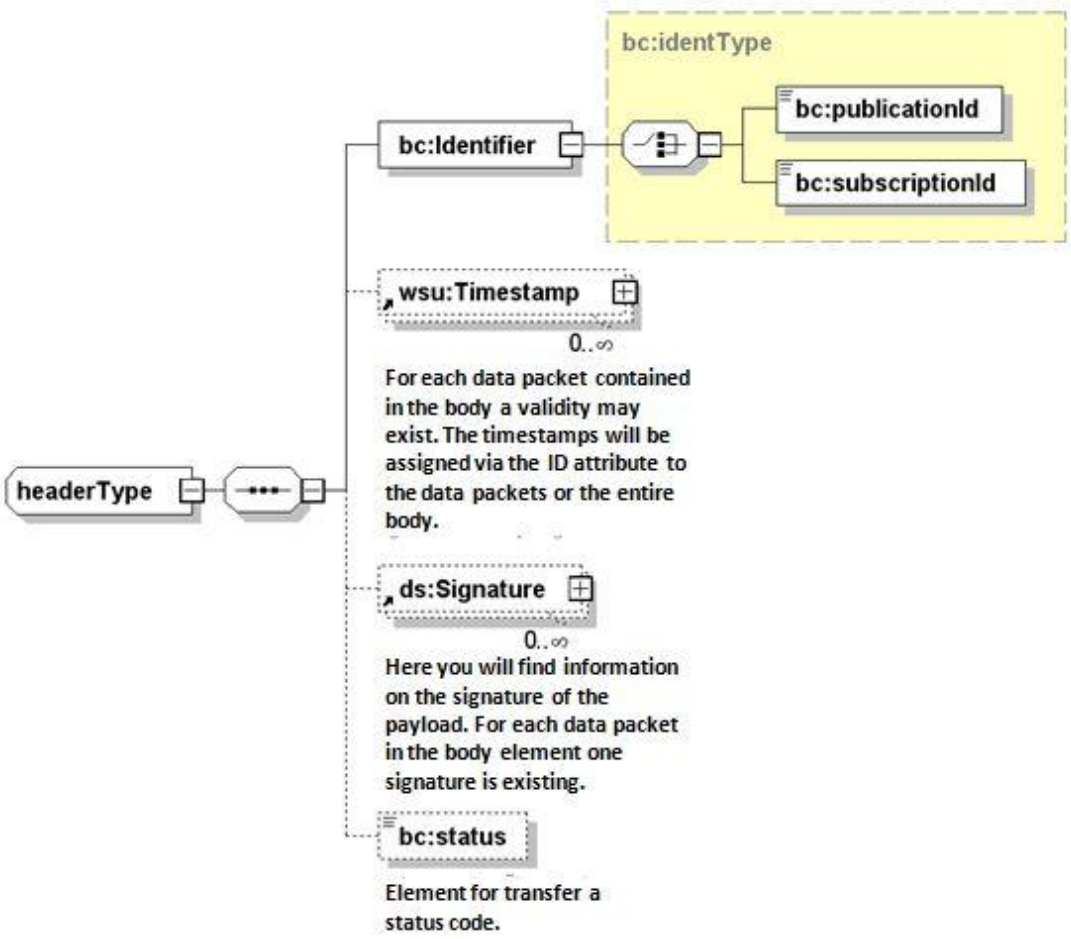

Figure 2: Container model - Header Element Structure

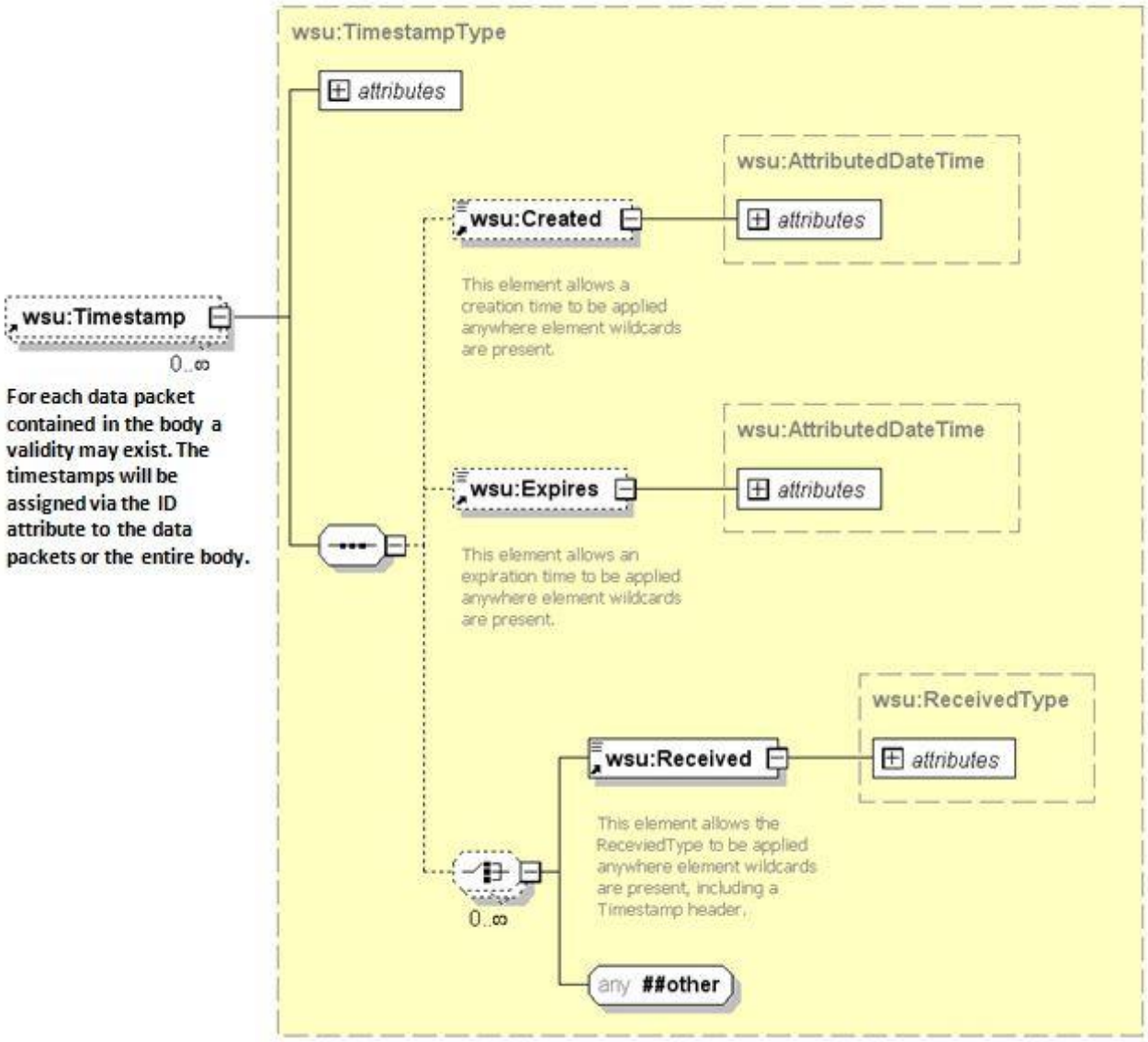

Figure 3: Construction Timestamp Element

The timestamp element is part of the OASIS specification [WS Security Core Specification 1.1] of the OASIS standard 1.1 and defines the time of creation as well as the validity period of data.

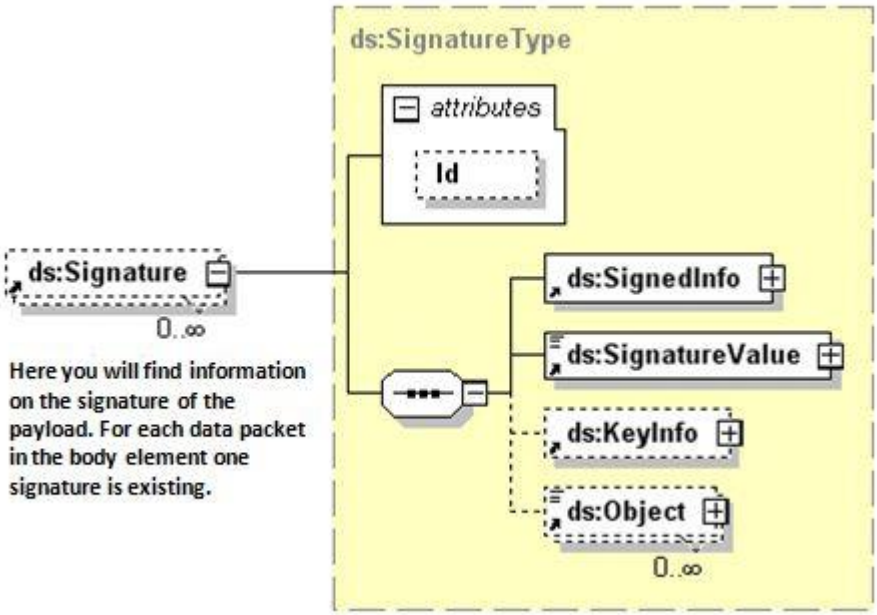

Figure 4: Structure of the signature element

The signature element is also part of the OASIS specification [WS Security Core Specification 1.1] of the OASIS standard 1.1 and defines the originator of the payload. In addition, the signature ensures that the payload has not been manipulated (data integrity).

If the element exists, only the value "OK" will be allowed as status code in version 1.0. Other status codes may be supplemented in future releases.

The encryption of the message content is not provided, as the message is sent SSL-encrypted across the entire route of transmission. An asymmetric encryption of the message content is not useful, as it would require an extensive key management from the MDM platform, to enable the data suppliers, when creating and sending a data packet to the platform, to identify the public keys of the data clients. Within the MDM platform, it must be assumed that the data is made available to authorized data clients only.

Information	Description
Identifier	Contains information to identify the origin of data • Publication ID • Subscription ID
Validity	Information on the validity period of data Date and time of creation Date and time of expiry Reference to the data for which the validity is determined
Signature	Information on the signature of data Type of signature Reference to the signed data
Status	For transferring a status code

Table 1: Structural information in the container model

2.2 Construction of the container body

In a container, there is exactly one body element. Below this element, as many xml and binary elements can be grouped. They host the actual payload.

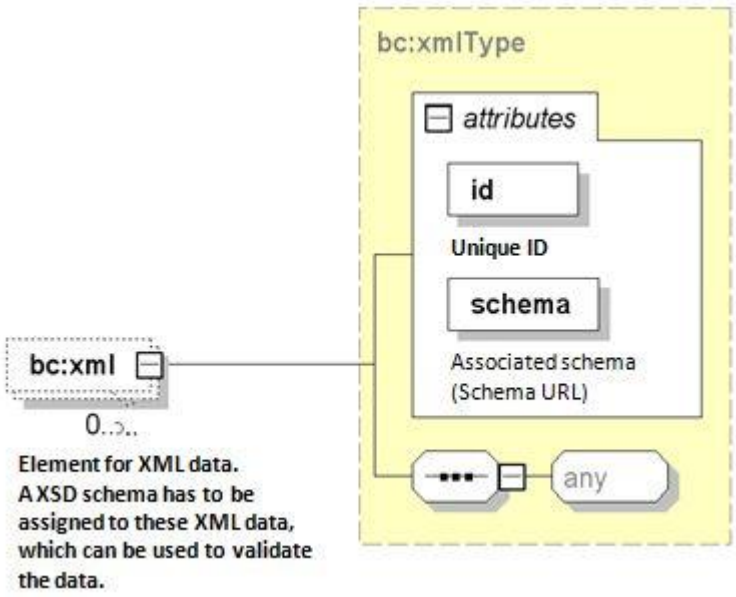

Figure 5: Structure of the XML element

- If the payload is provided in an XML-based format, they will then be installed below the XML element. The *scheme* attribute makes reference to an XSD schema belonging to the XML structure. This XSD schema can be used to validate the data. The *id* attribute assigns the payload to a unique ID that is referenced by the header elements *timestamp* and *signature*.

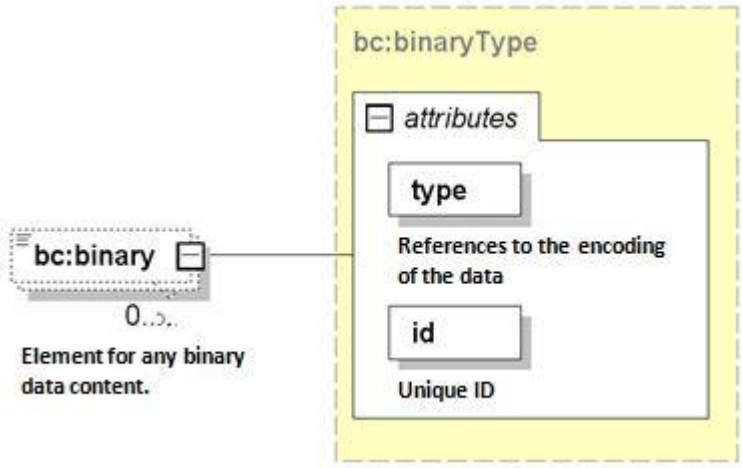

Figure 6: Structure of the binary element

- If the payload is provided as binary data, the latter will be installed under the binary element in BASE64-encoded form. The *type* attribute indicates the type of data. The *id* attribute assigns the payload to a unique ID that is referenced by the header elements *timestamp* and *signature*.

Type	Description
base64BinaryDatex2	Base64-encoded data according to [DatexIISchema].
base64BinaryOTS2	Base64-encoded data according to Open Transport System 2.
hexBinary	Binary data in hexadecimal notation. The type of data must be part of the coded data.
Todo	Placeholder for further formats

Table 2: Characteristics type attribute V1.0

A signature and a validity that are defined in the header element may belong to each of the *xml* and *binary* elements. For this purpose, the corresponding *XML / binary* element is assigned an ID that is referenced in the *signature / timestamp* element of the header.

If the entire content of the *body* element is provided with a signature or a validity, it must be encoded by using the ID "body" in the header elements. The *body* element itself must then be assigned no ID.

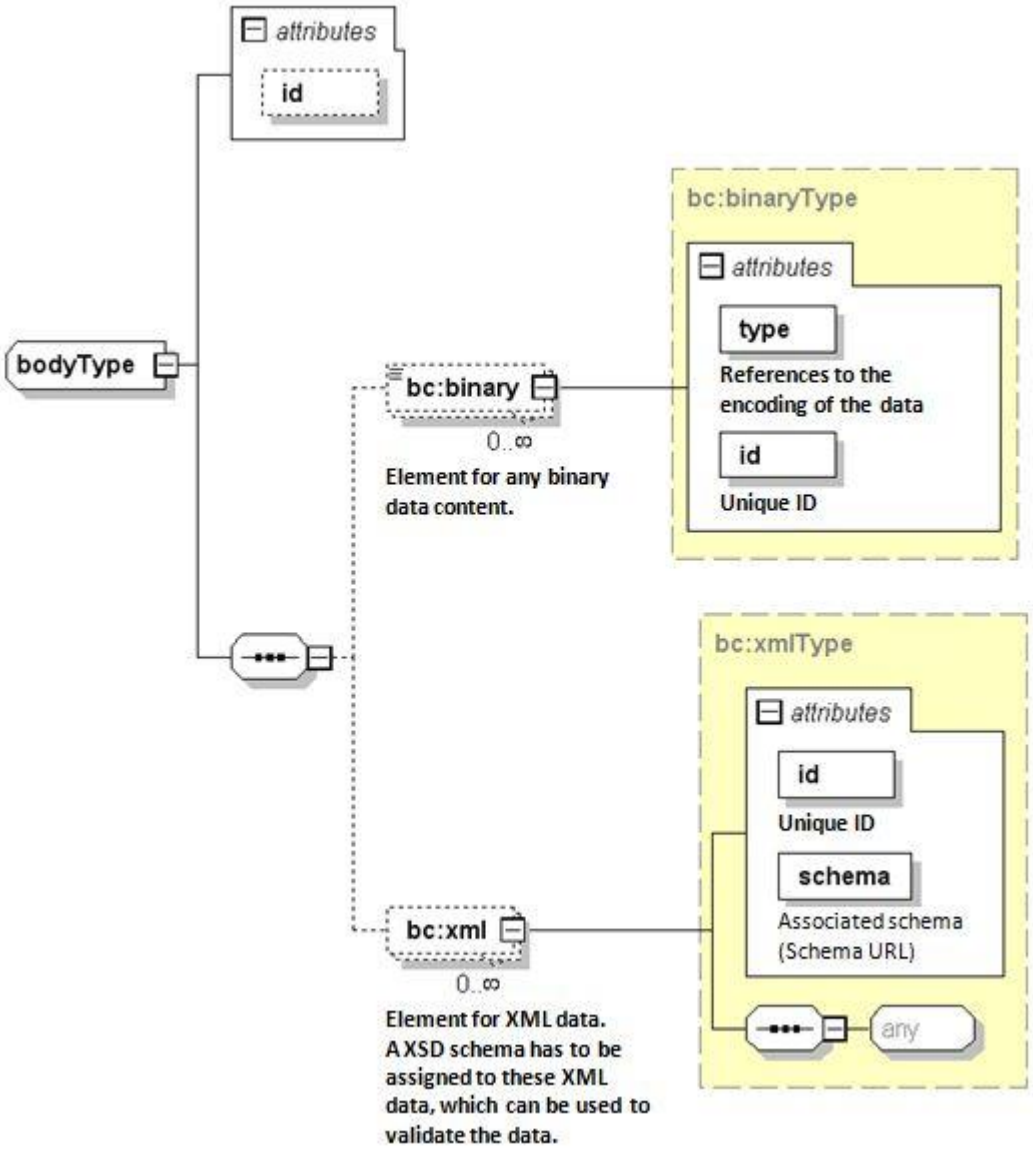

Figure 7: Container model - Body Element Structure

2.3 XML example

The example shows a container with two data packets in the body and a header with a time stamp for the entire body. All information is fictional.


```xml
<?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:Identifier>
            <ns3:publicationId>1734007</ns3:publicationId>
        </ns3:Identifier>
        <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">
H4sIAAAAAAAAAN19245dx5HluwH/A8EXPwxCjMiMvBVkNfI604A1Y0xrBvM2oMWymmiKNCi6bf/9&#xD;
RBYtdW225e4d3EzmHMOWi6oL1z61TtxjxZf/8OfvXz351/u3P7x88/rXv6Iv8FdP719/++bFy9ff&#xD;
/fpX/+ubAfFX//DVL3/x5Qvzmzffvfz2+auv37y4f/VEvuv1D79++s/v3v3h7tmz79/98C//+Pr3&#xD;
b95+//yd/JxnP3z7z/ffP3+GBPO/+PSXv3j/DXd//uHlT9/0pz/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/8OfvXz351/u3P7x88/rXv6Iv8FdP719/++bFy9ff&#xD;
H4sIAAAAAAAAAN19245dx5HluwH/A8EXPwxCjMiMvBVkNfI604A1Y0xrBvM2oMWymmiKNCi6bf/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//uHlT9/0pz/96Ys
        </ns3:binary>
    </ns3:body>

</ns3:container>
```