Koordinierungsstelle
für IT-Standards

OSCI-Transport 1.2
- 01/2025-
Status: Entwurf

Korrigenda Nr. 9

Gültig ab: 15.06.2025

Koordinierungsstelle für IT-Standards (KoSIT)

Bremen, 26.05.2025

Inhaltsverzeichnis

Copyright

Änderungshistorie

Datum	**Änderung**
05.05.2025	Bereitstellung zur internen Kommentierung
09.05.2025	Breitstellung zur Kommentierung durch Mitglieder der Expertengruppe Sicherer Transport
26.05.2025	Einarbeitung der Kommentierungen. Neben redaktionellen Änderungen wurden Gültigkeit und Übergangsfrist angepasst.

1. Einleitung

1.1 Anlass der Korrigenda

Die Notwendigkeit dieser Korrigenda ergab sich aus zwei im Rahmen der Arbeiten mit dem Standard festgestellten dringenden Problemen sowie dem Bedarf, die Interaktion zwischen XÖV-Standards und OSCI-Transport präziser zu gestalten.

1.2 Gültigkeit und Übergangsfrist

Diese Korrigenda tritt ab 15.06.2025 in Kraft.

1.3 Konventionen zur Textauszeichnung

Es gelten die gleichen Konventionen wie für die zugrundeliegende Spezifikation:

- Normative Absätze sind hellgrau unterlegt. Beispiel:

  Dieser Absatz ist normativ.

  In Zweifelsfällen gelten die Festlegungen in Schemata dieser Spezifikation vor normativen Textpassagen dieser Spezifikation. Diese gelten wiederum vor normativen Teilen referenzierter Dokumente und diese schließlich vor nicht-normativen Teilen dieser Spezifikation.
- Änderungen der Korrigenda zur Spezifikation sind innerhalb der normativen Textpassagen fett gesetzt.
- Jede Art von Code ist in `Schreibmaschinenschrift` gesetzt.

2. Fortschreibung der Spezifikation

Im Folgenden werden die einzelnen Änderungen mit Bezug auf die entsprechenden Kapitel der Spezifikation OSCI Transport 1.2 [OSCI12] detailliert beschrieben. Über einer geänderten Zeile befindet sich ein Kommentar mit einer kurzen Beschreibung der Änderung.

3. Fehlendes `minOccurs="0"` in ResponseToFetchProcessCard.xsd

In dem Schema ResponseToFetchProcessCard.xsd existiert es den `ComplexType` `responseToFetchProcessCardType`. In dieser Antwort wird die ursprüngliche Anfrage aus dem `FetchProcessCard`- Auftrag wiederholt.

Laut Kapitel „Laufzettelabholauftrag – `FetchProcessCard`“ der Spezifikation gilt:

"4. Ist keines der Elemente `osci:ReceptionOfDelivery`, `osci:RecentModification` oder `osci:MessageId` angegeben, so werden alle für den Benutzer vorliegenden Laufzettel gesendet."

Der Typ `responseToFetchProcessCardType` erlaubt bisher nicht, dass eine leere `fetchProcessCard` zurückgegeben wird.

Dieser Fehler kann durch Einfügen eines `minOccurs="0"` in dem `any`-Element korrigiert werden. Folgende Struktur ergibt sich dadurch in der ResponseToFetchProcessCard.xsd:

```
<xsd:complexType name="responseToFetchProcessCardType">
  <xsd:complexContent>
    <xsd:extension base="osci:DefaultBodyBlockTemplate">
```

```
      <xsd:sequence>
        <xsd:element name="Feedback" type="osci:FeedbackType" />
        <xsd:element name="fetchProcessCard">
          <xsd:complexType>
            <xsd:sequence>
<!-- KoSIT 2025-03-26 Patch start
Laut Spezifikation (OSCI incl. Korr1-7) Kapitel "6.7.15
Laufzettelabholauftrag - FetchProcessCard" gilt:
4. Ist keines der Elemente osci:ReceptionOfDelivery,
osci:RecentModification oder osci:MessageId angegeben, so werden alle für
den Benutzer vorliegenden Laufzettel gesendet.

Dies kann durch Einfügen von  minOccurs="0"  in der Response erreicht
werden.
-->
              <xsd:any namespace="http://www.osci.de/2002/04/osci"
                   processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
<!-- Patch 2025-03-26 ende -->

            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="ProcessCardBundle"
          type="osci:ProcessCardBundleType" minOccurs="0"
          maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

4. Aktualisierung kryptographischer Verfahren

Das Deutsche Patent- und Markenamt verwendet ECC-basierte D-Trust5.x Karten, die ausschließlich den Signaturalgorithmus SHA-384withECDSA unterstützen. Um diese Karten für Signaturen bei der OSCI-Kommunikation verwenden zu können, wird der Algorithmus in die Liste der erlaubten Algorithmen im Kapitel „Unterstützte Signieralgorithmen“ und „Ausprägung von XML-Signature“ für OSCI aufgenommen (in der folgenden Tabelle wurde der Eintrag **FETT** hervorgehoben).

4.1 Änderung an Kapitel „Unterstützte Signieralgorithmen“

Signatur-Algorithmus	**Algorithmus-Identifier (nach [RFC6931])**
Signaturschema RSASSA-PKCS1-v1_5 [PKCS_1] mit Hash-Algorithmus SHA-256. Modullänge des RSA-Schlüsselpaares mind. 2048 Bit	`http://www.w3.org/2001/04/xmldsig-more#rsa-sha256`
Signaturschema RSASSA-PKCS1-v1_5 [PKCS_1] mit Hash-Algorithmus SHA-512. Modullänge des RSA- Schlüsselpaares mind. 2048 Bit	`http://www.w3.org/2001/04/xmldsig-more#rsa-sha512`
Signaturschema RSASSA-PKCS1-PSS [PKCS_1] (ohne Parameter) mit Hash-Algorithmus SHA-256 gemäß RFC 6931 [RFC6931]. Modullänge des RSA-Schlüsselpaares mind. 2048 Bit	`http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1`
Signaturschema RSASSA-PKCS1-PSS [PKCS_1] (ohne Parameter) mit Hash- Algorithmus SHA-512 gemäß RFC 6931 [RFC6931]. Modullänge des RSA-Schlüsselpaares mind. 2048 Bit	`http://www.w3.org/2007/05/xmldsig-more#sha512-rsa-MGF1`
Signaturschema ECDSA-SHA* gemäß RFC 6931 [RFC6931] mit Hash-Algorithmus SHA-256	`http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256`
Signaturschema ECDSA-SHA* gemäß RFC 6931 [RFC6931] mit Hash-Algorithmus SHA-384	**`http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha384`**
Signaturschema ECDSA-SHA* gemäß RFC 6931 [RFC6931] mit Hash-Algorithmus SHA-512	`http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha512`
Signaturschema RSASSA-PKCS1-PSS [PKCS_1] (ohne Parameter) mit Hash-Algorithmus SHA3-256. Modullänge des RSA-Schlüsselpaares mind. 2048 Bit	`http://www.w3.org/2007/05/xmldsig-more#sha3-256-rsa-MGF1`
Signaturschema RSASSA-PKCS1-PSS [PKCS_1] (ohne Parameter) mit Hash-Algorithmus SHA3-384. Modullänge des RSA-Schlüsselpaares mind. 2048 Bit	`http://www.w3.org/2007/05/xmldsig-more#sha3-384-rsa-MGF1`
Signaturschema RSASSA-PKCS1-PSS [PKCS_1] (ohne Parameter) mit Hash-Algorithmus SHA3-512. Modullänge des RSA-Schlüsselpaares mind. 2048 Bit	`http://www.w3.org/2007/05/xmldsig-more#sha3-512-rsa-MGF1`

4.2 Ausprägung von XML-Signature

An dieser Stelle lediglich der veränderte Teil in oscisig.xsd:

```
<xsd:complexType name="SignatureMethodType">
  <xsd:complexContent>
    <xsd:restriction base="ds:SignatureMethodType">
```

```
      <xsd:attribute name="Algorithm" use="required">
        <xsd:simpleType>
          <xsd:restriction base="xsd:anyURI">
          <!-- [KORR2] Zeilen neu fuer rsa-sha256 und rsa-sha512-->
            <xsd:enumeration
              value="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256" />
            <xsd:enumeration
              value="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha512" />
          <!-- [KORR3] Zeilen neu fuer ecdsa-sha256 und ecdsa-sha512
-->
            <xsd:enumeration
              value="http://www.w3.org/2001/04/xmldsig-more#ecdsa-
sha256" />
          <!-- [KORR9] Zeile neu fuer ecdsa-sha384 -->
            <xsd:enumeration
              value="http://www.w3.org/2001/04/xmldsig-more#ecdsa-
sha384" />
            <xsd:enumeration
              value="http://www.w3.org/2001/04/xmldsig-more#ecdsa-
sha512" />
          <!-- [KORR4] Zeilen neu fuer sha256-rsa-MGF1 und sha512-
rsa-MGF1-->
            <xsd:enumeration
              value="http://www.w3.org/2007/05/xmldsig-more#sha256-
rsa-MGF1" />
            <xsd:enumeration
              value="http://www.w3.org/2007/05/xmldsig-more#sha512-
rsa-MGF1" />
          <!-- [KORR5] Zeilen neu fuer sha3-256-rsa-MGF1, sha3-384-
rsa-MGF1, sha3-512-rsa-MGF1 -->
            <xsd:enumeration
              value="http://www.w3.org/2007/05/xmldsig-more#sha3-
256-rsa-MGF1" />
            <xsd:enumeration
              value="http://www.w3.org/2007/05/xmldsig-more#sha3-
384-rsa-MGF1" />
            <xsd:enumeration
              value="http://www.w3.org/2007/05/xmldsig-more#sha3-
512-rsa-MGF1" />
          <!-- [KORR3] Zeile entfernt
            xsd:enumeration
              value="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/> -
->
          <!-- [KORR4] Zeile entfernt
            xsd:enumeration
              value="http://www.w3.org/2001/04/xmldsig-more#rsa-
ripemd160"/> -->
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:restriction>
```

```
  </xsd:complexContent>
</xsd:complexType>
```

5. Redaktionelle Änderung der Bezeichner der Kommunikationsverfahren

In XÖV-Standards gibt es häufig ein Kapitel mit Vorgaben für OSCI-Transport, das Bezeichnungen für Kommunikationsszenarien verwendet, die nicht direkt in der OSCI-Spezifikation enthalten sind, sondern in DVDV-WSDL-Templates und XÖV-Standards zum Einsatz kommen. Um den Zusammenhang zwischen den Bezeichnern eindeutig zu machen, wird der in den XÖV-Dokumenten vorhandenen Bezeichner an den entsprechenden Stellen der OSCI-Spezifikation angehängt.

Folgende Kapitel werden um den Teil in Klammern erweitert:

- One-Way-Message, aktiver Empfänger, Protokollierung **(one-way-active)**
- One-Way-Message, passiver Empfänger, Protokollierung **(one-way-passive)**
- Request-Response, passiver Empfänger, Protokollierung **(request-response)**
- Request-Response, passiver Empfänger, keine Protokollierung **(request-response-noprotocol)**

6. Literaturverzeichnis

[OSCI12] OSCI Transport 1.2 – Spezifikation; OSCI Leitstelle 2002, http://www.xoev.de/detail.php?gsid=bremen83.c.2472.de

[EFFI] Ergänzung zur Spezifikation OSCI 1.2 - Effiziente Übertragung großer Datenmengen, KoSIT 2017-10-25, https://www.xoev.de/sixcms/media.php/13/OSCI-1.2_mit_Korrigenda.pdf

[Korr] Korrigenda 1-8 zu OSCI-Transport 1.2, KoSIT 2024, https://www.xoev.de/sixcms/media.php/13/OSCI_Korrigenda.zip