



Koordinierungsstelle
für IT-Standards



Ergänzung zu Spezifikation OSCI 1.2: Neuer Laufzettel Status: ENTWURF

Gültig ab: dd.mm.yyyy

Koordinierungsstelle für IT-Standards (KoSIT)

Bremen, 28.04.2026

1. Inhalt

1.	Inhalt.....	2
	Copyright	3
	Änderungshistorie	4
2.	Einleitung.....	5
2.1	Anlass der Ergänzung	5
2.2	Gültigkeit und Übergangsfrist	5
2.3	Konventionen zur Textauszeichnung	5
2.4	Anmerkungen.....	6
2.4.1	eIDAS-Verordnung	6
2.4.2	Begriffe „Laufzettel“ und „Laufzettel-Bündel“ (aka LaufzettelBundle)	6
3.	Änderungsanforderungen	6
3.1	CR 2017-01 Neue Struktur für einen OSCI 1.2 Laufzettel	6
3.1.1	Die Anforderungen	7
3.1.2	Umsetzung der Anforderungen	7
3.2	CR 2018-01 eIDAS Prüfergebnisse für Signaturen und Zertifikate.....	12
3.2.1	Die Anforderungen	13
3.2.2	Umsetzung der Anforderungen	13
3.2.3	Hinweis zum Umgang mit ETSI/ eIDAS	14
3.3	CR 2018-03 Laufzettel eigenständig signieren.....	14
3.3.1	Die Anforderungen	14
3.3.2	Umsetzung der Anforderungen	15
3.4	CR 2018-02 Qualitätsanforderungen an Aufträge an den OSCI-Intermediär	16
3.4.1	Die Anforderungen	16
3.4.2	Umsetzung der Anforderungen	17
4.	Geänderte Dateien	18
5.	Neue Dateien.....	18
6.	Quellcode	20
6.1	ProcessCardExtendedOldNS.xsd	20
6.2	ProcessCardExtended.xsd	21
7.	Literaturverzeichnis	36

Copyright

Die vorliegende Ergänzung der Spezifikation OSCI-Transport 1.2 wurde durch die Koordinierungsstelle für IT-Standards in Zusammenarbeit mit dem Expertengremium Sicherer Transport erarbeitet.

Diese Ergänzung ist urheberrechtlich geschützt. Alle Nutzungsrechte liegen beim Herausgeber. Herstellern wird zur Implementation von Bürger-, Kommunal-, Intermediär- oder Dienstleistungssystemen unentgeltlich ein einfaches Nutzungsrecht eingeräumt. Im Rahmen des genannten Zwecks darf dieses Dokument in unveränderter Form vervielfältigt und zu den nachstehenden Bedingungen verbreitet werden.

Umgestaltungen, Bearbeitungen, Übersetzungen und jegliche Änderungen sind nur nach Rücksprache mit dem Herausgeber zulässig. Kennzeichnungen, Copyright-Vermerke und Eigentumsangaben sind beizubehalten.

Haftung für Mängel dieses Dokuments wird nur bei Vorsatz und grober Fahrlässigkeit übernommen. Hersteller der oben genannten Systeme sind gebeten, Fehler, Unklarheiten oder Interpretationsfreiräume dieser Spezifikation, die die ordnungsgemäße Funktion oder die Interoperabilität behindern, dem Herausgeber zu melden.

Eine Weitergabe dieses Dokuments an Dritte darf nur unentgeltlich, in unveränderter Form und zu den vorstehenden Bedingungen erfolgen.

Änderungshistorie

Datum	Änderung
28.03.2024	Erster Entwurf für Rücksprache mit Governikus
16.05.2024	Start Überarbeitung/ Vervollständigung nach EG24/02
14.10.2025	Weitere Überarbeitung nach EG25/03
10.11.2025	Aktualisierung für EG25/04
02.12.2025	Start Überarbeitung nach EG25/04
02.02.2026	Finalisierte Fassung für EG26/01
28.04.2026	<ul style="list-style-type: none">• Einarbeitung Rückmeldungen nach EG26/01• Freigabe zur öffentlichen Kommentierung

2. Einleitung

2.1 Anlass der Ergänzung

Die Anforderungen an die im OSCI-Laufzettel bereitzustellenden Informationen haben sich erhöht. Um den neuen Vorgaben zu entsprechen, sind entsprechende Anpassungen erforderlich. In dieser Ergänzung werden die Anforderungen systematisch beschrieben und ein Vorschlag zur Umsetzung aufgezeigt. Im gesamten Dokument wird der Laufzettel gemäß der gestiegenen Anforderungen als *neuer Laufzettel* bezeichnet. Die Bezeichnung *alter Laufzettel* gilt für die Umsetzung des Laufzettels ohne die in diesem Dokument beschriebenen, erhöhten Anforderungen.

2.2 Gültigkeit und Übergangsfrist

Diese Ergänzung ist nicht unmittelbar verpflichtend anzuwenden, es wird jedoch empfohlen, die Umsetzung zeitnah einzuplanen. Mit der Ergänzung werden rechtliche Vorgaben zur eIDAS-Verordnung der EU umgesetzt, zusätzliche Nachweise für den Transport ermittelt, bereitgestellt und separat verfügbare signierte bzw. gesiegelte Laufzettel für einzelne Nachweise verfügbar gemacht.

Ohne einen Stichtag, an dem alle Sender, Empfänger und Intermediäre umgestellt werden, können 4 Szenarien auftreten, die alle weiterhin funktionieren müssen. Sie werden im Folgenden mit dem daraus folgenden Zustand beschrieben.

Szenarien:

1. Weder Sender/ Empfänger noch der Intermediär unterstützen den neuen Laufzettel. Der alte Laufzettel wird verwendet. *Es entsteht kein Problem.*
2. Sender/ Empfänger unterstützt den neuen Laufzettel, der Intermediär jedoch nicht. Wird ein neuer Laufzettel angefordert, wird nur der alte Laufzettel zurückgeliefert, den alle aktuell existierenden Implementierungen auswerten können. *Es entsteht kein Problem.*
3. Sender/ Empfänger unterstützen den neuen Laufzettel nicht, der Intermediär schon. Der Intermediär könnte den neuen Laufzettel zurückliefern, dieser wird jedoch nicht angefragt. Der Intermediär liefert daher nur den alten Laufzettel zurück. *Es entsteht kein Problem.*
4. Intermediär und nur entweder Sender oder Empfänger unterstützt den neuen Laufzettel. Der Intermediär hält alle Daten bereit, um sowohl den alten als auch neuen Laufzettel zur Verfügung zu stellen. Sender und Empfänger erhalten unterschiedliche Informationen im Laufzettel, da der neue Laufzettel mehr Informationen enthält als der alte. Ein 1:1-Vergleich der Laufzettel durch Sender/ Empfänger ist nicht möglich. Es ist jedoch bisher kein Fall bekannt, bei dem das getan wird. *Es entsteht kein Problem.*

Für jedes beschriebene Szenario gibt es eine Lösung. Daher steht einer Umsetzung ohne Stichtag nichts im Wege. Langfristig betrachtet sollte die Ergänzung in der Infrastruktur flächendeckend umgesetzt werden, um alle rechtlichen Vorgaben zu erfüllen.

2.3 Konventionen zur Textauszeichnung

Für diese Ergänzung 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 Ergänzung zur Spezifikation sind innerhalb der normativen Textpassagen fett gesetzt.
- Jede Art von Code ist in Schreibmaschinenschrift gesetzt.

2.4 Anmerkungen

2.4.1 eIDAS-Verordnung

Mit der eIDAS-Verordnung [EUR-LEX] wird eine klare begriffliche Trennung zwischen Signatur und Siegel vorgenommen: Eine „natürliche Person“ (also ein Mensch) unterzeichnet bzw. signiert, während eine „juristische Person“ (in Deutschland etwa ein Unternehmen ab der Rechtsform GmbH) siegelt.

In der Spezifikation OSCI-Transport 1.2 sowie den dazu vorliegenden Ergänzungen wird diese Unterscheidung bislang nicht getroffen. Nach Abstimmung mit der Expertengruppe Sicherer Transport wurde im EG24/02 beschlossen, dass ein erläuternder Hinweis zu Beginn der OSCI-Spezifikation ausreicht, um klarzustellen, dass der Begriff „Signatur“ dort stellvertretend sowohl für „Signatur“ als auch für „Siegel“ verwendet wird. Auch in diesem Dokument werden die Begriffe „Signatur“ und „Siegel“ in diesem Sinne synonym gebraucht.

2.4.2 Begriffe „Laufzettel“ und „Laufzettel-Bündel“ (aka LaufzettelBundle)

In den Änderungsanträgen wird von „Laufzetteln“ gesprochen. Technisch gesehen ist hiermit die Struktur „ProcessCardBundle“ der OSCI-Spezifikation und nicht „ProcessCard“ gemeint. Dies wurde im EG24/02 klargestellt. In dieser Ergänzung wird der Begriff „Laufzettel“ synonym für „ProcessCardBundle“ verwendet – und für alle weiteren Strukturen die ggf. „ProcessCard“ und „Bundle“ im selben Namen tragen (z.B. „Request**ProcessCardExtendedBundle**Type“).

3. Änderungsanforderungen

Die Änderungsanforderungen wurden im EG Sicherer Transport definiert und eine Umsetzung beschlossen. Sie lagen ursprünglich als Textdokumente vor und wurden im Lauf der Arbeit in Einträge in ein nicht-öffentliches GitLab für die EG-Teilnehmenden übertragen.

Da jeder Änderungsantrag Unterpunkte enthalten kann, werden die vorhandenen Anträge mit Unterpunkten versehen, um eine Umsetzung nachvollziehen zu können.

3.1 CR 2017-01 Neue Struktur für einen OSCI 1.2 Laufzettel

Der Laufzettel wurde seit Erstellung des OSCI-Transport 1.2 Spezifikation nicht verändert. Er entsprach den damaligen Anforderungen, muss nun aber aktuellen Anforderungen, wie zum Beispiel der eIDAS-Verordnung, angepasst werden. Um den laufenden Betrieb nicht zu stören und die Validierbarkeit der alten Laufzettel zu erhalten, darf die Definition gemäß der initialen OSCI-Transport 1.2 Spezifikation nicht sofort entfernt werden.

3.1.1 Die Anforderungen

- Req. 1. : Es braucht eine neue Struktur für einen OSCI-Transport 1.2 Laufzettel.
- Req. 1.1. Alter und neuer Laufzettel müssen nebeneinander existieren können.
- Req. 1.2. Es muss bei Abfrage des Laufzettels die Möglichkeit bestehen, mit der auf den alten oder den neuen Laufzettel zugegriffen werden kann.
- Req. 1.3. Der neue Laufzettel soll erweiterbar sein.
- Req. 1.4. Der neue Laufzettel muss mit extern definierten Strukturen umgehen können.

3.1.2 Umsetzung der Anforderungen

3.1.2.1 Umsetzung Req. 1. : „Neue Struktur für einen OSCI 1.2 Laufzettel“

Die vorhandene `ProcessCardTemplate` und `ProcessCardBundleTemplate` werden in `order.xsd` definiert und liegen im OSCI-Namespacespace `http://www.osci.de/2002/04/osci`. Wenn diese Strukturen in einer Nachricht verwendet werden sollen, dann wird in dem zugehörigen Schema aus dem `ProcessCardBundleTemplate` über eine `Restriction` ein konkreter `ProcessCardBundleType` erstellt. Auf die gleiche Weise wird aus einem `ProcessCardTemplate` mit einer `Restriction` eine `ProcessCardType`. Es gibt aktuell sieben verschiedene `ProcessCardBundleType` und `ProcessCardType`, verteilt auf folgende sechs Schema-Dateien:

- `AcceptDelivery.xsd`
- `ProcessDelivery.xsd`
- `ResponseToFetchDelivery.xsd`
- `ResponseToForwardDelivery.xsd`
- `ResponseToMediateDelivery.xsd` (enthält 2 `ProcessCardType` und `ProcessCardBundleType`)
- `ResponseToStoreDelivery.xsd`

Die Struktur für das neue `ProcessCardExtendedBundleTemplate` und `ProcessCardExtendedTemplate` werden in der Schema-Datei `ProcessCardExtended.xsd` mit dem Namespace `http://xoev.de/transport/osci/osciLauf2026` definiert. Um nicht alle neuen Strukturen auf viele Dateien zu verteilen, werden die über `restriction` erzeugten konkreten Strukturen ebenfalls in dieser Datei hinterlegt und an der benötigten Stelle eingebunden.

3.1.2.2 Umsetzung von Req. 1.1: „Alter und neuer Laufzettel müssen nebeneinander existieren können“

Der alte Laufzettel von OSCI-Transport 1.2 muss vorerst erhalten bleiben, um den Betrieb nicht zu stören. Später soll neben dem neuen Laufzettel ein weiterer, neuerer Laufzettel existieren können, damit auch der neue Laufzettel ohne erneute Änderungen der Schemadateien abgelöst werden kann. Die Verwaltung der unterstützten Versionen geschieht über eine Codeliste mit folgenden Informationen:

Metadaten

- Kennung `urn:xoev-de:kosit:codeliste:osci.validation-reports`
- Name (technisch) `osci.validation-reports`

Felder

- | | |
|-----------------|------------------------------------|
| ○ Code | Eindeutiger Name des Eintrages |
| ○ Bezeichner | Bezeichner des Eintrages |
| ○ Version | Versionsnummer |
| ○ Schema-URL | URL des Schemas |
| ○ NotBefore | Früheste Verwendung des Eintrages |
| ○ NotAfter | Letzte zulässige Verwendung |
| ○ Spezifikation | Link zur Spezifikation des Schemas |

Es wird eine neue Nachricht `FetchProcessCardExtended` zur expliziten Abholung von erweiterten Laufzetteln definiert. Die Antwort wird ebenfalls in der neuen Nachricht `ResponseToFetchProcessCardExtended.xsd` definiert.

Eine Reihe von Nachrichten geben implizit einen Laufzettel zurück. Diese Stellen müssen so modifiziert werden, dass entweder der alte Laufzettel oder der neue Laufzettel zurückgegeben werden kann. Betroffen sind hier folgende Nachrichten:

- `AcceptDelivery.xsd`
- `ProcessDelivery.xsd`
- `ResponseToFetchDelivery.xsd`
- `ResponseToForwardDelivery.xsd`
- `ResponseToMediateDelivery.xsd`
- `ResponseToPartialStoreDelivery.xsd`
- `ResponseToPartialStoreDeliveryOldNS.xsd`
- `ResponseToStoreDelivery.xsd`

Der bisher vorhandene Block in den Dateien

```
<xsd:sequence>
  <xsd:element name="ProcessCardBundle"
    type="osci:ProcessCardBundleType" />
</xsd:sequence>
```

wird durch folgenden Block ersetzt

```
<!-- [OSCILauf] {Req. 1.1} start
  Alter und neuer Laufzettel müssen nebeneinander existieren können
  Es wird entweder eine Sequence von "normalen" Request/Reply-
  ProcessCardBundle zurück gegeben oder eine Sequence von "neuen"
  Request/Reply-ProCardExtBndl.
-->
<xsd:choice>
  <xsd:sequence>
    <xsd:element name="ProcessCardBundle"
      type="osci:ProcessCardBundleType"/>
  </xsd:sequence>
  <xsd:sequence>
    <xsd:element name="SignedProcessCardBundleExtended"
      type="osciLauf2026:SignedProCardExtBndl_RespFetchDelType"/>
  </xsd:sequence>
</xsd:choice>
<!-- [OSCILauf] {Req. 1.1} ende -->
```

Sollte in dem bisherigen Block das `ProcessCardBundle` optional sein über `minOccurs="0"`, dann muss dies auch durch den ersetzenden Block so erhalten bleiben.

Namespace und Import werden über folgende Zeilen eingebunden:

```
xmlns:osciLauf2026="http://xoev.de/transport/osci12/osciLauf2026"

<!-- [OSCIlauf] {Req. 1.1} -->
<xsd:import namespace="http://xoev.de/transport/osci12/osciLauf2026"
  schemaLocation="./ProcessCardExtended.xsd" />
```

3.1.2.3 Umsetzung von Req. 1.2: „Es muss bei Abfrage des Laufzettels die Möglichkeit bestehen, mit der auf den alten oder den neuen Laufzettel zugegriffen werden kann“

Um den neuen Laufzettel abholen zu können, werden eine neue Nachricht zum Abholen (`FetchProcessCardExtended.xsd`) und für den Empfang (`ResponseToFetchProcessCardExtended.xsd`) definiert.

Für Nachrichten die implizit einen Laufzettel zurückgeben muss schon beim Aufruf signalisiert werden, ob ggf. der neue Laufzettel zurückgegeben werden soll. Hierfür wird in aufrufenden Nachrichten und Antworten im SOAP-Header eine `FeatureDescription` mitgeschickt. Dies ist in folgenden Nachrichten der Fall:

- `soapAcceptDelivery.xsd`
- `soapFetchDelivery.xsd`
- `soapForwardDelivery.xsd`
- `soapMediateDelivery.xsd`
- `soapProcessDelivery.xsd`
- `soapResponseToFetchDelivery.xsd`
- `soapResponseToForwardDelivery.xsd`
- `soapResponseToMediateDelivery.xsd`
- `soapResponseToStoreDelivery.xsd`
- `soapStoreDelivery.xsd`

In diesen Nachrichten wird im SOAP-Header zwischen die Elemente `NonIntermediaryCertificates` und `##other` das Element `FeatureDescription` eingefügt. Es ergibt sich folgende Reihenfolge:

```
<xsd:element ref="osci:NonIntermediaryCertificates" minOccurs="0" />

<!-- [OSCIlauf] {Req. 1.2} Ein AcceptDelivery liefert auch einen Laufzettel
zurück. Hier muss die Möglichkeit bestehen, auch den neuen, erweiterten
Laufzettel zu erhalten. -->
<xsd:element ref="osci2017:FeatureDescription" minOccurs="0"/>

<xsd:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
```

Namespace und Import werden über folgende Zeilen eingefügt:

```
xmlns:osci2017="http://xoev.de/transport/osci12/7"
```

```
<!-- [OSCIlauf] {Req. 1.2} namespace und Import um ggf. den erweiterten
Laufzettel zu erhalten -->
<xsd:import namespace="http://xoev.de/transport/osci12/7"
  schemaLocation="./EFFI.xsd"/>
```

Nicht angepasst werden müssen `soapResponseToPartialFetchDelivery.xsd` und `soapResponseToPartialStoreDelivery.xsd` da hier schon eine `FeatureDescription` enthalten ist.

In den passiven Kommunikationsszenarien von OSCI 1.2 schickt der Intermediär eine Nachricht direkt an den Empfänger. In der Nachricht kann direkt ein Laufzettel enthalten sein. Der Empfänger kann in diesem Fall nicht zuerst die Anforderung schicken, dass der neue Laufzettel enthalten sein soll. Für diese Fälle gibt es folgende Lösungen:

1. Der Empfänger schickt in der Response eine `FeatureDescription` mit der Information, dass der neue Laufzettel für ggf. folgende Nachrichten in einem Dialog verwendet werden soll. Den Laufzettel für die zuvor erhaltene Nachricht muss der Empfänger „manuell“ per `FetchProcessCardExtended` abholen.
2. Auf dem Intermediär wird gespeichert, ob ein Sender/ Empfänger den neuen Laufzettel abgeholt hatte und speichert die Konfiguration für diesen Client. Danach wird immer der neue Laufzettel für diesen Client verwendet. Die Möglichkeit dies zu ändern, muss für den Betreiber des Clients möglich sein um hier ggf. steuernd eingreifen zu können.

3.1.2.4 Umsetzung von Req. 1.3: „Der neue Laufzettel soll erweiterbar sein“

Um die Erweiterbarkeit des neuen Laufzettels zu erreichen, wird der Extension-Mechanismus von XTA verwendet. Um eine direkte Abhängigkeit von XTA zu vermeiden, werden die Strukturen kopiert und in OSCI in der Schema-Datei `extensions.xsd` neu definiert.

Quelle: <https://xoev.de/transport/xta/core/3.1.0/XTA-Webservice-Datentypen.xsd>

```
<xsd:schema
  targetNamespace="http://www.osci.de/2002/04/osci12/osciExtensions"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:osciExt="http://www.osci.de/2002/04/osci12/osciExtensions"
  attributeFormDefault="unqualified" elementFormDefault="qualified">
```

```
<!--
  [OSCIlauf] {Req.1.2} Erweiterbarkeit
  Mit ExtensionType und ExtensionsType werden Strukturen für die
  Erweiterbarkeit eingefügt. Sie stammen aus XTA und funktionieren
  dort bereits. Um keine Abhängigkeit in OSCI von XTA zu erhalten,
  werden die Strukturen nach OSCI kopiert.
```

Quelle der XTA-Strukturen:
<https://xoev.de/transport/xta/core/3.1.0/XTA-Webservice-Datentypen.xsd>

Folgende Zeilen müssen bei der Verwendung eingefügt werden, um die Erweiterbarkeit zu Erreichen:

```
Bekanntgabe Namespace:
<xsd:schema targetNamespace=
  ...
  xmlns:osciExt="http://www.osci.de/2002/04/osci12/osciExtensions"
  ...>
```

```
Import Namespace:
  <xsd:import
    namespace="http://www.osci.de/2002/04/osci12/osciExtensions"
    schemaLocation="./extensions.xsd"/>

Element-Zeile in einer Struktur die den Mechanismus erhalten soll:
  <xsd:element name="Extensions" type="osciExt:ExtensionsType"
    minOccurs="0" />
-->

<xsd:annotation>
  <xsd:documentation xml:lang="de">
    OSCI 1.2 - Typen und Strukturen für Extension-Mechanismen
    Dateiname: extensions.xsd
  </xsd:documentation>
</xsd:annotation>

<xsd:simpleType name="NonEmptyURIType">
  <xsd:annotation>
    <xsd:documentation>Eine Ressourcenidentifikation mit der
      Mindestlänge 1.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:anyURI">
    <xsd:minLength value="1"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="NonEmptyStringType">
  <xsd:annotation>
    <xsd:documentation>Eine Zeichenkette mit der
      Mindestlänge 1.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="ExtensionType">
  <xsd:annotation>
    <xsd:documentation>Ein Element zum Hinzufügen einer
      Extension.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:sequence>
    <xsd:element name="Identifizier" type="osciExt:NonEmptyURIType">
      <xsd:annotation>
        <xsd:documentation>Eindeutige Kennung der verwendeten Ergänzung
          Als URI-Angabe. Beispiel:
          urn:xoev-de:kosit:standard:xta.extended-contentcontainer
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>

    <xsd:element name="Version" type="osciExt:NonEmptyStringType">
      <xsd:annotation>
        <xsd:documentation>Verwendete Version der Ergänzung nach dem
          Semantic Versioning Prinzip.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```

```

</xsd:element>

<xsd:element name="Data" type="xsd:anyType">
  <xsd:annotation>
    <xsd:documentation>Enthält Daten die mit einem eigenen Schema
      definiert und nach einer dazugehörigen Spezifikation der
      Ergänzung verwendet werden müssen.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>

<xsd:attribute name="id" type="xsd:ID" use="required">
  <xsd:annotation>
    <xsd:documentation>Hier muss die ID zur eindeutigen Referenzierung
      des XML-Elements eingetragen werden. Ihr Verwendungszweck ist
      außerhalb der Spezifikation beschrieben.
    </xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:complexType>

<xsd:complexType name="ExtensionsType">
  <xsd:annotation>
    <xsd:documentation>Eine Liste von Ergänzungen, welche die Funktionen
      des XTA Kerns nach eigener Spezifikation ergänzen.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:sequence>
    <xsd:element name="Extension" type="osciExt:ExtensionType"
      maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>Eine Ergänzung der Kernparameterinhalte.
      </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

3.1.2.5 Umsetzung von Req. 1.4: „Der neue Laufzettel muss extern definierte Strukturen umgehen können“

Wenn extern definierte Strukturen eingebunden werden sollen, muss hierfür eine Extension gemäß Req. 1.3 erstellt werden.

3.2 CR 2018-01 eIDAS Prüfergebnisse für Signaturen und Zertifikate

In den Laufzetteln sind Validierungsinformationen zu Signaturen und Zertifikaten enthalten. Das Ergebnis wird im Element `InspectionReport` abgelegt. Um eIDAS-konforme Reports zu erhalten, soll eine entsprechende Überprüfung durchgeführt und protokolliert werden. Die neuen Prüfungen und Protokolle ersetzen die alten Reports. In [ETSI] wurden für die Protokollierungen entsprechende Strukturen definiert, welche in Zukunft verwendet werden sollen. Da nicht klar ist wie oft diese extern definierten Strukturen geändert werden, muss die Einbindung in OSCI entsprechend flexibel gestaltet sein.

3.2.1 Die Anforderungen

- Req. 2. : eIDAS Prüfergebnisse für Signaturen und Zertifikate
- Req. 2.1. Das Ergebnis der Überprüfung von Zertifikaten und Signaturen wird in eIDAS-/ETSI-Strukturen abgelegt
- Req. 2.2. Es soll ein Katalog erstellt und gepflegt werden, der Auskunft über unterstützte/empfohlene eIDAS-/ETSI-Versionen gibt

3.2.2 Umsetzung der Anforderungen

3.2.2.1 Umsetzung von Req. 2.1: „Das Ergebnis der Überprüfung von Zertifikaten und Signaturen wird in eIDAS-/ETSI-Strukturen abgelegt“

Das bisher in OSCI verwendete Element `InspectionReport` im Laufzettel wird durch das neue Element `ValidationReport` ersetzt.

Da wir keinen Einfluss auf die eIDAS-/ ETSI-Strukturen haben, müssen diese möglichst flexibel eingebunden werden. Hierfür wird folgende Struktur verwendet.

```

<!-- [OSCIlauf] {Req.2} eIDAS-ValidationReports start -->
<xsd:complexType name="ValidationReportsType">
  <xsd:annotation>
    <xsd:documentation>
      Struktur für Validation Reports.
      Alte Inspections können in dieser neuen Struktur abgelegt werden.
    </xsd:documentation>
  </xsd:annotation>
</xsd:complexType>

<!-- [OSCIlauf] {Req.2.1} flexible Einbindung von eIDAS/ ETSI start -->
<xsd:sequence>
  <xsd:element name="Data" type="xsd:anyType"
    minOccurs="0" maxOccurs="unbounded">
    <xsd:annotation>
      <xsd:documentation>Enthält Daten die mit einem eigenen Schema
        definiert und nach einer dazugehörigen Spezifikation der
        Erweiterung verwendet werden müssen.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:sequence>
<!-- [OSCIlauf] {Req.2.1} flexible Einbindung von eIDAS/ ETSI ende -->

<!-- [OSCIlauf] {Req.2.2} Verwendung eines Katalogs/ einer Codelist
start
-->
<xsd:attribute name="Code" type="osciExt:NonEmptyStringType"
  use="required">
  <!-- Beispiel: https://www.xrepository.de/details/urn:xoev-
de:kosit:codeliste:osci.validation-reports_0.1.0#version -->
  <xsd:annotation>
    <xsd:documentation>Es ist der Eintrag der Spalte "Code" aus der
      aktuell gültigen Codeliste mit der Kennung
      urn:xoev-de:kosit:codeliste:osci.validation-reports
      einzutragen.
    </xsd:documentation>
  </xsd:annotation>
</xsd:attribute>

```

```

<!-- [OSCIlauf] {Req.2.2} Verwendung eines Katalogs/ einer Codelist
  ende
-->

</xsd:complexType>
<!-- [OSCIlauf] {Req.2} eIDAS-ValidationReports ende -->

```

3.2.2.2 Umsetzung von Req. 2.2: „Es soll ein Katalog erstellt und gepflegt werden, der Auskunft über unterstützte/ empfohlene eIDAS-/ETSI-Versionen gibt“

Um die Anzahl der zu unterstützenden Report-Strukturen eingrenzen zu können, soll eine Codelist in der XRepository hinterlegt werden (aktuell liegt eine Version 0.1.0 hier https://www.xrepository.de/details/urn:xoev-de:kosit:codelist:osci.validation-reports_0.1.0#version) mit Informationen hierzu. Diese wird in der zuvor aufgebauten Struktur mit dem Attribut Code referenziert.

Metadaten

- Kennung urn:xoev-de:kosit:codelist:osci.validation-reports
- Name (technisch) osci.validation-reports

Felder

- Code Eindeutiger Name des Eintrages
- Bezeichner Bezeichner des Eintrages
- Version Versionsnummer
- Schema-URL URL des Schemas
- NotBefore Früheste Verwendung des Eintrages
- NotAfter Letzte zulässige Verwendung
- Spezifikation Link zur Spezifikation des Schemas

3.2.3 Hinweis zum Umgang mit ETSI/ eIDAS

Die in dem [ETSI]-Dokument aufgebauten Strukturen und Inhalte sind gemäß dem Dokument zu verwenden. Eine Nachspezifikation innerhalb dieser Erweiterung findet nicht statt, da der Aufwand für die jeweils gültigen Versionen nachgepflegt werden müsste.

3.3 CR 2018-03 Laufzettel eigenständig signieren

Ein Laufzettel wird bisher nicht unabhängig signiert, sondern nur im Zusammenhang mit der gesamten Nachricht, in der er enthalten ist. Soll der Laufzettel extrahiert und gespeichert werden, dann fehlt aktuell die Signatur, was ihn als Beweis nicht-benutzbar macht. Wird an dem Laufzettel selbst eine Signatur angebracht, dann ändert sich dies. Für den neuen Laufzettel ergeben sich daraus folgende Anforderungen:

3.3.1 Die Anforderungen

- Req. 3. : Laufzettel eigenständig signieren
- Req. 3.1. Gibt der Intermediär ein einzelnes `ProcessCardBundle` zurück, dann signiert er dies
- Req. 3.2. Auch einzelne Laufzettel, die in Nachrichten enthalten sind werden signiert
- Req. 3.3. Eine Option Signatur nur bei Bedarf anbringen zu lassen gibt es nicht

3.3.2 Umsetzung der Anforderungen

Die Anforderungen beziehen sich nur auf den neuen Laufzettel und den neuen, signierten Laufzettel (ProcessCardExtendedBundleTemplate und SignedProCardExtBndlTemplate) .

3.3.2.1 Umsetzung von Req. 3.: „Laufzettel eigenständig signieren“

Der neue Laufzettel kann signiert werden.

Für bisher in OSCI enthaltene Laufzettel wurde eine Struktur in order.xsd als Template hinterlegt und in einer ganzen Reihe von Schema-Dateien für den konkreten Anwendungsfall modifiziert. Für den neuen Laufzettel wurden all diese Strukturen weitestgehend zentral in ProcessCardExtended.xsd gesammelt. Für den neuen, signierten Laufzettel ergibt sich folgende Struktur:

```
<!-- [OSCIlauf] {Req.3} start -->
<xsd:complexType name="SignedProCardExtBndlTemplate">
  <xsd:annotation>
    <xsd:documentation>
      Template für die neuen, erweiterten, signierten
      ProcessCardBundle.
      Die Signatur ist optional.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:sequence>
    <xsd:element ref="ds:Signature" minOccurs="0" maxOccurs="1"/>
    <xsd:choice>
      <!-- AcceptDelivery -->
      <xsd:element name="ProCardExtAcceptDel"
        type="osciLauf2026:ProCardExtBndl_AcceptDelType"
        minOccurs="1" maxOccurs="1"/>
      <!-- ProcessDelivery -->
      <xsd:element name="ProCardExtProcessDel"
        type="osciLauf2026:ProCardExtBndl_ProcessDelType"
        minOccurs="1" maxOccurs="1"/>

      <!-- MediateDelivery Reply -->
      <xsd:element name="ProCardExtReply"
        type="osciLauf2026:ProCardExtBndl_ReplyProCardType"
        minOccurs="1" maxOccurs="1"/>
      <!-- MediateDelivery Request -->
      <xsd:element name="ProCardExtReq"
        type="osciLauf2026:ProCardExtBndl_ReqProCardType"
        minOccurs="1" maxOccurs="1"/>

      <!-- ResponseToFetchDelivery -->
      <xsd:element name="ProCardExtRespFetchDel"
        type="osciLauf2026:ProCardExtBndl_RespFetchDelType"
        minOccurs="1" maxOccurs="1"/>
      <!-- ResponseToForwardDelivery -->
      <xsd:element name="ProCardExtRespForwDel"
        type="osciLauf2026:ProCardExtBndl_RespForwDelType"
        minOccurs="1" maxOccurs="1"/>
      <!-- ResponseToStoreDelivery -->
      <xsd:element name="ProCardExtRespStoreDel"
        type="osciLauf2026:ProCardExtBndl_RespStoreDelType"
        minOccurs="1" maxOccurs="1"/>
    </xsd:choice>
  </xsd:sequence>
</complexType>
```

```
</xsd:complexType>  
<!-- [OSCIlauf] {Req.3} ende -->
```

3.3.2.2 Umsetzung von Req. 3.1.: „Gibt der Intermediär einzelne *ProcessCardExtendedBundle* zurück, dann signiert er diese“

Wird in einer Nachricht ein einzelnes *ProcessCardExtendedBundle* zurückgegeben, dann wird dies immer signiert. Wird mehr als ein *ProcessCardExtendedBundle* zurückgegeben, dann wird keine Signatur angebracht. Eine Signatur des SOAP-Bodys findet in beiden Fällen statt.

3.3.2.3 Umsetzung von Req. 3.2.: „Auch einzelne Laufzettel, die in Nachrichten enthalten sind werden signiert“

Es gibt eine Reihe von Nachrichten, die Laufzettel mit übertragen (siehe Kapitel 3.1.2.1). Sofern nur genau ein Laufzettel enthalten ist, wird dies signiert.

3.3.2.4 Umsetzung Req. 3.3.: „Eine Option Signatur nur bei Bedarf anbringen zu lassen gibt es nicht“

Jenseits der zuvor aufgeführten Regelung zur Anbringung einer Signatur gibt es keine Möglichkeit, einzelne Laufzettel nicht signieren zu lassen.

3.4 CR 2018-02 Qualitätsanforderungen an Aufträge an den OSCI-Intermediär

Bisher kann im LaufzettelBundle nicht dokumentiert werden, mit welchen Transporteigenschaften (bspw. Transport-Signatur oder Transport-Verschlüsselung) der Sender einen Auftrag an den OSCI-Intermediär übergeben hat. Dadurch ist es nicht möglich die fachlich geforderten Qualitäten mit den geleisteten abzugleichen und Fehler und Probleme diesbezüglich festzustellen.

Künftig soll der OSCI-Intermediär die Transporteigenschaften zu jeder transportieren Nachricht feststellen, protokollieren und auf dem Laufzettel dokumentieren. Damit wird den Rollen Empfänger und Leser ein Soll-/ Ist-Abgleich ermöglicht, ob die fachlich geforderten Transporteigenschaften von Sender und Intermediär eingehalten wurden.

3.4.1 Die Anforderungen

Req. 4. : Qualitätsanforderungen an Aufträge an den OSCI-Intermediär

Req. 4.1. Zur Signatur des Transportauftrages sollen weitere Informationen enthalten sein

- Wurde die Transportnachricht signiert oder nicht.
- Welcher Signaturalgorithmus wurde verwendet.
- Welcher Digestalgorithmus wurde verwendet.

Req. 4.2. Zur Verschlüsselung des Transportauftrages sollen weitere Informationen enthalten sein

- Wurde die Transportnachricht verschlüsselt oder nicht.
- Welcher symmetrische Verschlüsselungsalgorithmus wurde verwendet.
- Länge des Initialisierungsvektors *IvLength* ().
- Welcher Digestalgorithmus wurde verwendet.
- Welcher asymmetrische Verschlüsselungsalgorithmus wurde verwendet

- SubjectDN¹

3.4.2 Umsetzung der Anforderungen

Das `ProcessCardExtendedBundle` muss um Informationen zu Signatur und Verschlüsselung des Transportauftrags erweitert werden. Hierfür werden die notwendigen Strukturen definiert und dann eingebunden.

3.4.2.1 Ergänzung in `ProcessCardExtended.xsd`

Im `complexType` `name="ProcessCardExtendedTemplate"` wird nach dem Element `ValidationReport` die Elemente `TransportEncryptionInspection` und `TransportSignatureInspection` eingefügt.

```
<!-- [OSCILauf] {Req.4} {Req.4.1} {Req.4.2} start -->
<xsd:element name="TransportEncryptionInspection"
  type="osciLauf2026:TransportEncryptionInspectionType" minOccurs="0"/>

<xsd:element name="TransportSignatureInspection"
  type="osciLauf2026:TransportSignatureInspectionType" minOccurs="0"/>
<!-- [OSCILauf] {Req.4} {Req.4.1} {Req.4.2} ende -->
```

3.4.2.2 Umsetzung Req. 4.1.: „Zur Signatur des Transportauftrages sollen weitere Informationen enthalten sein“

Die geforderten Informationen werden in folgender Struktur festgelegt:

```
<!-- [OSCILauf] {Req.4.1} start -->
<xsd:complexType name="TransportSignatureCheckType">
  <xsd:sequence>
    <xsd:element name="SignatureAlgorithm" type="xsd:anyURI"
      minOccurs="0" maxOccurs="1"/>
    <xsd:element name="DigestAlgorithm" type="xsd:anyURI"
      minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="isTransportSigned" type="xsd:boolean"
    use="required"/>
</xsd:complexType>
<!-- [OSCILauf] {Req.4.1} ende -->
```

3.4.2.3 Umsetzung Req. 4.2.: „Zur Verschlüsselung des Transportauftrages sollen weitere Informationen enthalten“

Die geforderten Informationen werden in folgender Struktur festgelegt:

```
<!-- [OSCILauf] {Req.4.2} start -->
<xsd:complexType name="TransportEncryptionInspectionType">
  <xsd:sequence>
```

¹ SubjectDN

<https://datatracker.ietf.org/doc/html/rfc5280> , Kapitel <https://datatracker.ietf.org/doc/html/rfc5280#section-4.2.1.13> :

„When the LDAP URI scheme [RFC4516] is used, the URI MUST include a field containing the distinguished name of the entry holding the CRL, MUST include a single that contains an appropriate attribute description for the attribute that holds the CRL [RFC4523], and SHOULD include a (e.g., `<ldap://ldap.example.com/cn=example%20CA,dc=example,dc=com?certificateRevocationList;binary>`).“

```
<xsd:element name="SymmetricKeyAlgorithm" type="xsd:anyURI"
  minOccurs="0" maxOccurs="1"/>
<xsd:element name="IvLength" type="osci128:IvLengthType"
  minOccurs="0" maxOccurs="1"/>
<xsd:element name="DigestAlgorithm" type="xsd:anyURI"
  minOccurs="0" maxOccurs="1" />
<xsd:element name="AsymmetricKeyAlgorithm" type="xsd:anyURI"
  minOccurs="0" maxOccurs="1"/>
<xsd:element name="SubjectDN" type="osciExt:NonEmptyStringType"
  minOccurs="1" maxOccurs="1"/>
<!-- Falls für Post-Quantum-Cryptographie weitere Informationen
gespeichert werden müssen, so werden diese hier nachgetragen. -->
</xsd:sequence>

<xsd:attribute name="IsTransportEncrypted" type="xsd:boolean"
  use="required"/>
</xsd:complexType>
<!-- [OSCIlauf] {Req.4.2} ende -->
```

4. Geänderte Dateien

- AcceptDelivery.xsd
- ProcessDelivery.xsd
- ResponseToFetchDelivery.xsd
- ResponseToForwardDelivery.xsd
- ResponseToMediateDelivery.xsd
- ResponseToPartialStoreDelivery.xsd
- ResponseToPartialStoreDeliveryOldNS.xsd
- ResponseToStoreDelivery.xsd

- soapAcceptDelivery.xsd
- soapFetchDelivery.xsd
- soapForwardDelivery.xsd
- soapMediateDelivery.xsd
- soapProcessDelivery.xsd
- soapResponseToFetchDelivery.xsd
- soapResponseToForwardDelivery.xsd
- soapResponseToMediateDelivery.xsd
- soapResponseToStoreDelivery.xsd
- soapStoreDelivery.xsd

5. Neue Dateien

- extensions.xsd

- ProcessCardExtended.xsd
- ProcessCardExtendedOldNS.xsd

- FetchProcessCardExtended.xsd
- soapFetchProcessCardExtended.xsd

- ResponseToFetchProcessCardExtended.xsd
- soapResponseToFetchProcessCardExtended.xsd

6. Quellcode

6.1 ProcessCardExtendedOldNS.xsd

```

<xsd:schema targetNamespace="http://www.osci.de/2002/04/osci"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  xmlns:osci2017="http://xoev.de/transport/osci12/7"
  elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xsd:include schemaLocation="./order.xsd" />
  <xsd:import namespace="http://xoev.de/transport/osci12/7"
    schemaLocation="./EFFI.xsd"/>

  <!--
    [OSCIlauf] Import alter Namespaces und Verfügbarmachung einiger anderen
    Elemente.
  -->

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 2024 - Import bestehender Datentypen
      Dateiname: ProcessCardExtendedOldNS.xsd
    </xsd:documentation>
  </xsd:annotation>

  <!--### global complex types and templates ### -->

  <xsd:complexType name="ControlBlockType">
    <xsd:complexContent>
      <xsd:restriction base="osci:ControlBlockTemplate">
        <xsd:sequence>
          <xsd:element name="Response" type="xsd:string" minOccurs="1" />
          <xsd:element name="Challenge" type="xsd:string" minOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="ConversationId" type="osci:Number"
          use="required" />
        <xsd:attribute name="SequenceNumber" type="osci:Number"
          use="required" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="NonIntermediaryCertificatesType">
    <xsd:complexContent>
      <xsd:restriction
        base="osci:NonIntermediaryCertificatesTemplate">
        <xsd:sequence>
          <xsd:element name="SignatureCertificateOriginator"
            type="osci:CertificateType" minOccurs="0" />
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:element name="MessageId" type="osci:MessageIdType"/>
  <xsd:element name="Feedback" type="osci:FeedbackType" />
  <xsd:element name="InspectionReport" type="osci:InspectionReportType" />

  <xsd:element name="ControlBlock" type="osci:ControlBlockType"/>
  <xsd:element name="NonIntermediaryCertificates"
    type="osci:NonIntermediaryCertificatesType"/>
  <xsd:element name="ClientSignature" type="osci:SignatureBlockType" />
  <xsd:element name="DesiredLanguages" type="osci:DesiredLanguagesType" />

  <xsd:element name="ChunkInformationStore"
    type="osci2017:ChunkInformationTemplate" />
  <xsd:element name="ChunkInformationFetch"

```

```

type="osci2017:ChunkInformationTemplate" />

<!--
  <xsd:element name="SupplierSignature" type="osci"/>
-->
</xsd:schema>

```

6.2 ProcessCardExtended.xsd

```

<xsd:schema targetNamespace="http://xoev.de/transport/osci12/osciLauf2026"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:osci="http://www.osci.de/2002/04/osci"
  xmlns:osci2017="http://xoev.de/transport/osci12/7"
  xmlns:osci128="http://xoev.de/transport/osci12/8"
  xmlns:osciExt="http://www.osci.de/2002/04/osci12/osciExtensions"
  xmlns:osciLauf2026="http://xoev.de/transport/osci12/osciLauf2026"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <xsd:import namespace="http://www.osci.de/2002/04/osci"
    schemaLocation="./ProcessCardExtendedOldNS.xsd"/>
  <xsd:import namespace="http://xoev.de/transport/osci12/8"
    schemaLocation="./order_zusatz.xsd"/>

  <xsd:import namespace="http://www.w3.org/2000/09/xmldsig#"
    schemaLocation="./oscisig.xsd"/>

  <xsd:import namespace="http://xoev.de/transport/osci12/7"
    schemaLocation="./EFFI.xsd"/>
  <xsd:import namespace="http://www.osci.de/2002/04/osci12/osciExtensions"
    schemaLocation="./extensions.xsd"/>

  <!--
    [OSCIlauf] {Req.1} Neue Struktur für einen OSCI 1.2 Laufzettel
    Struktur alter Laufzettel in order.xsd.
    Struktur neuer Laufzettel und LaufzettelBundle mit neuem Namen hier
    definiert.

    [OSCIlauf] {Req.1.1} Alter und neuer Laufzettel müssen nebeneinander
    existieren können.
    Alter und neuer Laufzettel in jeweils eigener Schema-Datei und eigenem
    Namespace.

    [OSCIlauf] {Req.1.2} Erweiterbarkeit
    Zeile für Erweiterbarkeit in LaufzettelBundle eingefügt. Siehe auch in der
    jeweiligen Struktur.
  <xsd:element name="Extensions" minOccurs="0" type="osciExt:ExtensionsType"
  />

  [OSCIlauf] {Req.1.4} Der neue Laufzettel muss extern definierte Strukturen
  umgehen können, wie sie zum Beispiel mit eIDAS existieren.
  Hierfür muss eine Extension erstellt werden, die über die Erweiterbarkeit
  erreicht wird.
  -->

  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      OSCI 1.2 - Erweiterter Laufzettel.
    </xsd:documentation>
  </xsd:annotation>

  <!-- [OSCIlauf] {Req.4} start
    Umsetzung für "CR 2018-02 Qualitätsanforderungen an Aufträge an den OSCI-
    Intermediär"
    https://projekte.kosit.org/transport/osci/-/issues/7

```

```

- Strukturen eingefügt für TransportSignatureCheckType und
TransportEncryptionType
- Referenz auf diese Strukturen eingefügt in ProcessCardExtendedTemplate
(unter element Subject)
-->

<!-- [OSCIlauf] {Req.4.1} Zur Signatur des Transportauftrages sollen weitere
Informationen enthalten sein.
start -->
<xsd:complexType name="TransportSignatureInspectionType">
  <xsd:annotation>
    <xsd:documentation>
      Informationen ob und wie ein Transportauftrag signiert war.
      [OSCIlauf] {Req.4.1} CR 2018-02 Qualitätsanforderungen an Aufträge
an den OSCI-Intermediär
    </xsd:documentation>
  </xsd:annotation>

  <xsd:sequence>
    <xsd:element name="SignatureAlgorithm" minOccurs="0" maxOccurs="1"
type="xsd:anyURI"/>
    <xsd:element name="DigestAlgorithm" minOccurs="0" maxOccurs="1"
type="xsd:anyURI"/>
  </xsd:sequence>
  <xsd:attribute name="IsTransportSigned" type="xsd:boolean" use="required"/>
</xsd:complexType>
<!-- [OSCIlauf] {Req.4.1} ende -->

<!-- [OSCIlauf] {Req.4.2} Hilfsdatentypen start
Quellen:
  https://www.rfc-editor.org/rfc/rfc5280#section-4.1.2.4
  https://en.wikipedia.org/wiki/X.509#Root_certificate

JKS 2025-12-02: nachbauen der Datentypen aus X.509-Quellen erscheint zu
aufwendig,
daher nur Einsatz von nonEmptyString.
-->
<!--
Name ::= CHOICE { - only one possibility for now -
  rdnSequence  RDNSequence }

RDNSequence ::= SEQUENCE OF RelativeDistinguishedName

RelativeDistinguishedName ::=
  SET SIZE (1..MAX) OF AttributeTypeAndValue

AttributeTypeAndValue ::= SEQUENCE {
  type      AttributeType,
  value     AttributeValue }

AttributeType ::= OBJECT IDENTIFIER

AttributeValue ::= ANY - DEFINED BY AttributeType

DirectoryString ::= CHOICE {
  teletexString      TeletexString (SIZE (1..MAX)),
  printableString    PrintableString (SIZE (1..MAX)),
  universalString    UniversalString (SIZE (1..MAX)),
  utf8String         UTF8String (SIZE (1..MAX)),
  bmpString          BMPString (SIZE (1..MAX)) }
-->

<!-- [OSCIlauf] {Req.4.2} Hilfsdatentypen ende -->

<!-- [OSCIlauf] {Req.4.2} Zur Verschlüsselung des Transportauftrages sollen
weitere Informationen enthalten sein.
start -->

```

```

    <xsd:complexType name="TransportEncryptionInspectionType">
      <xsd:annotation>
        <xsd:documentation>
          Informationen ob und wie ein Transportauftrag verschlüsselt war.
          [OSCIlauf] {Req.4.2} CR 2018-02 Qualitätsanforderungen an Aufträge
an den OSCI-Intermediär
        </xsd:documentation>
      </xsd:annotation>

      <xsd:sequence>
        <xsd:element name="SymmetricKeyAlgorithm" type="xsd:anyURI"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="IvLength" type="osci128:IvLengthType" minOccurs="0"
maxOccurs="1"/>
        <xsd:element name="DigestAlgorithm" type="xsd:anyURI" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="AsymmetricKeyAlgorithm" type="xsd:anyURI"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="SubjectDN" type="osciExt:NonEmptyStringType"
minOccurs="1" maxOccurs="1"/>
        <!-- Falls für Post-Quantum-Cryptographie weitere Informationen
gespeichert werden müssen,
so werden diese hier nachgetragen. -->
      </xsd:sequence>
      <xsd:attribute name="IsTransportEncrypted" type="xsd:boolean"
use="required"/>
    </xsd:complexType>
    <!-- [OSCIlauf] {Req.4.2} ende -->

    <!-- [OSCIlauf] {Req.4} ende -->

    <!--
      SelectionRule(s) eigentlich in order.xsd definiert.
      Um Strukturen für den erweiterten Laufzettel zusammen zu halten hier
definiert
-->
    <xsd:complexType name="SelectionRuleFetchExtendedType">
      <xsd:annotation>
        <xsd:documentation>
          Auswahlkriterien beim Abrufen von Laufzettel.
        </xsd:documentation>
      </xsd:annotation>

      <xsd:choice>
        <xsd:element name="ReceptionOfDelivery" type="xsd:dateTime"/>
        <xsd:element name="RecentModification" type="xsd:dateTime"
minOccurs="0"/>
        <xsd:element name="MessageId" type="osci:MessageIdType"
maxOccurs="unbounded"/>
      </xsd:choice>
    </xsd:complexType>

    <!-- [OSCIlauf] {Req.1} und folgende start -->
    <xsd:complexType name="ProCardExtTemplate" abstract="true">
      <xsd:annotation>
        <xsd:documentation>
          Template für die erweiterten ProcessCards.
          Sind bei den alten ProcessCards verteilt auf die entsprechenden
Nachrichten.
        </xsd:documentation>
      </xsd:annotation>

      <xsd:sequence>
        <xsd:element name="Creation" type="osci:TimestampType" minOccurs="1">
          <xsd:annotation>
            <xsd:documentation>
              Datum und Zeit der Erzeugung des Laufzettels. Entspricht
dem Zeitpunkt an dem die Nachricht vollständig auf dem Intermediär angekommen war.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:sequence>

```

```

        </xsd:annotation>
    </xsd:element>
    <xsd:element name="Forwarding" type="osci:TimestampType" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>
                Datum und Zeit der Weiterleitung der Nachricht.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="Reception" type="osci:TimestampType" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>
                Datum und Zeit der Abholung der Nachricht vom Intermediär.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="Subject" type="xsd:string" minOccurs="0"/>
</xsd:sequence>
<xsd:attribute name="RecentModification" type="xsd:dateTime"
use="required"/>
</xsd:complexType>
<!-- [OSCIlauf] {Req.1} und folgende ende -->

<!-- [OSCIlauf] {Req.3} start -->
<xsd:complexType name="SignedProCardExtBndlTemplate">
    <xsd:annotation>
        <xsd:documentation>
            Template für die neuen, erweiterten, signierten ProcessCardBundle.
            Die Signatur ist optional.
        </xsd:documentation>
    </xsd:annotation>
</xsd:complexType>

<xsd:sequence>
    <xsd:element ref="ds:Signature" minOccurs="0" maxOccurs="1"/>
    <xsd:choice>
        <!-- AcceptDelivery -->
        <xsd:element name="ProCardExtAcceptDel"
type="osciLlauf2026:ProCardExtBndl_AcceptDelType"
minOccurs="1" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>
                    ProcessCard für eine AcceptDelivery.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <!-- ProcessDelivery -->
        <xsd:element name="ProCardExtProcessDel"
type="osciLlauf2026:ProCardExtBndl_ProcessDelType"
minOccurs="1" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>
                    ProcessCard für eine ProcessDelivery.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <!-- MediateDelivery Reply -->
        <xsd:element name="ProCardExtReply"
type="osciLlauf2026:ProCardExtBndl_ReplyProCardType"
minOccurs="1" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>
                    ProcessCard für eine MediateDelivery-Reply.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <!-- MediateDelivery Request -->
        <xsd:element name="ProCardExtReq"
type="osciLlauf2026:ProCardExtBndl_ReqProCardType"
minOccurs="1" maxOccurs="1">

```

```

        <xsd:annotation>
          <xsd:documentation>
            ProcessCard für eine MediateDelivery-Request.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>

      <!-- ResponseToFetchDelivery -->
      <xsd:element name="ProCardExtRespFetchDel"
type="osciLauf2026:ProCardExtBndl_RespFetchDelType"
minOccurs="1" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>
            ProcessCard für eine ResponseToFetchDelivery.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <!-- ResponseToForwardDelivery -->
      <xsd:element name="ProCardExtRespForwDel"
type="osciLauf2026:ProCardExtBndl_RespForwDelType"
minOccurs="1" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>
            ProcessCard für eine ResponseToForwardDelivery.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <!-- ResponseToStoreDelivery -->
      <xsd:element name="ProCardExtRespStoreDel"
type="osciLauf2026:ProCardExtBndl_RespStoreDelType"
minOccurs="1" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>
            ProcessCard für eine ResponseToStoreDelivery.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>
<!-- [OSCIlauf] {Req.3} ende -->

<!-- [OSCIlauf] {Req.2} eIDAS-ValidationReports start -->
<xsd:complexType name="ValidationReportsType">
  <xsd:annotation>
    <xsd:documentation>
      Struktur für Validation Reports.
      Alte Inspections können in dieser neuen Struktur abgelegt werden.
    </xsd:documentation>
  </xsd:annotation>

  <!-- [OSCIlauf] {Req.2.1} flexible Einbindung von eIDAS/ ETSI start -->
  <xsd:sequence>
    <xsd:element name="Data" type="xsd:anyType" minOccurs="0"
maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>Enthält Daten die mit einem eigenen Schema
definiert und nach einer dazugehörigen Spezifikation der Erweiterung verwendet
werden müssen.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
  <!-- [OSCIlauf] {Req.2.1} flexible Einbindung von eIDAS/ ETSI ende -->

  <!-- [OSCIlauf] {Req.2.2} Verwendung eines Katalogs/ einer Codelist start -
->
  <xsd:attribute name="Code" type="osciExt:NonEmptyStringType"
use="required">
    <!-- Beispiel: https://www.xrepository.de/details/urn:xoev-
```

```

de:kosit:codelist:osci.validation-reports_0.1.0#version -->
  <xsd:annotation>
    <xsd:documentation>Es ist der Eintrag der Spalte "Code" aus der
aktuell gültigen Codelist mit der Kennung urn:xoev-
de:kosit:codelist:osci.validation-reports einzutragen.
    </xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
<!-- [OSCIlauf] {Req.2.2} Verwendung eines Katalogs/ einer Codelist ende --
>

</xsd:complexType>
<!-- [OSCIlauf] {Req.2} eIDAS-ValidationReports ende -->

<!-- Template des neuen, erweiterten Laufzettels -->
<xsd:complexType name="ProCardExtBndlTemplate" abstract="true">
  <xsd:annotation>
    <xsd:documentation>
      Template für ein erweitertes ProcessCardBundle.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:sequence>
    <!-- Werte aus altem Laufzettel -->
    <xsd:element name="MessageId" type="osci:MessageIdType">
      <xsd:annotation>
        <xsd:documentation>
          MessageId der Nachricht.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="ProCardExt" type="osciLauf2026:ProCardExtTemplate">
      <xsd:annotation>
        <xsd:documentation>
          ProcessCard der Nachricht. Enthält Infos zu Empfang,
Weiterleitung/ Abholung, etc. der Nachricht zum und vom Intermediär.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>

    <!-- [OSCIlauf] {Req.2} start -->
    <xsd:element name="ValidationReport"
type="osciLauf2026:ValidationReportsType" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>
          eIDAS-/ETSI-ValidationReport von Zertifikaten und
Signaturen. Ersetzt das alte Element InspectionReport.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <!-- [OSCIlauf] {Req.2} ende -->

    <!-- [OSCIlauf] {Req.4} {Req.4.1} {Req.4.2} start -->
    <xsd:element name="TransportEncryptionInspection"
type="osciLauf2026:TransportEncryptionInspectionType"
minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>
          Informationen ob die Transportnachricht verschlüsselt war,
welcher Algorithmus zum Einsatz kam etc.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="TransportSignatureInspection"
type="osciLauf2026:TransportSignatureInspectionType"
minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>
          Information, ob die t-transportierte Nachricht signiert war

```

etc.

```

        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  <!-- [OSCIlauf] {Req.4} {Req.4.1} {Req.4.2} ende -->

  <!-- [OSCIlauf] {Req 1.3} {Req. 1.4} Erweiterbarkeit
    EG24-02 am 19.04.2024: Verwendung des XTA Extension-Mechanismus
  -->
  <xsd:element name="Extensions" minOccurs="0"
type="osciExt:ExtensionsType">
    <xsd:annotation>
      <xsd:documentation>Eine Liste von optionalen Erweiterungen wie
sie auch in XTA 2 Version 5 verwendet wird. Der Einsatz von Erweiterungen muss vor
dem Einsatz zwischen Sender, Intermediär und Empfänger besprochen und geklärt
sein.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:sequence>

  <!-- [OSCIlauf] {Req.3.1} Laufzettel signieren -->
  <!-- Id für Signatur -->
  <xsd:attribute name="Id" type="xsd:ID" use="required">
    <xsd:annotation>
      <xsd:documentation>ID um ein ProCardExtBndlTemplate signieren zu
können.
      [OSCIlauf] {Req.3} CR 2018-03 Laufzettel eigenständig
signieren.
    </xsd:documentation>
  </xsd:attribute>

  <!-- CR 2018-05 Aus welchem Netz wurde die Nachricht abgesendet
    CR wurde im EG 2024/01 wg. Nicht-Umsetzbarkeit verworfen.
    Mögliches Attribut wurde für Dokumentationszwecke und ggf. spätere
    Umsetzung erhalten.
  -->
  <xsd:attribute name="Network" type="xsd:string"/>
  -->

</xsd:complexType>

  <!-- Template des neuen, erweiterten Laufzettels -->

<!--
  <xsd:complexType name="SignedProCardExtBndlType">
    <xsd:annotation>
      <xsd:documentation>
        Template für ein erweitertes, signiertes ProcessCardBundle.
        [OSCIlauf] {Req.3} CR 2018-03 Laufzettel eigenständig signieren.
      </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
      <xsd:element name="ProCardExtBndl"
type="osciLauf2026:ProCardExtBndlTemplate"/>
      <xsd:element name="Signature" type="ds:SignatureType" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  -->

  <!-- [OSCIlauf] Anmerkung
    In order.xsd sind abstrakte Templates für ProcessCard und ProcessCardBundle
    definiert.
    Die in [OSCIlauf] in Kapitel "Geänderte Dateien" aufgeführten Dateien
    machen aus den
    abstrakten Typen konkrete Typen und ändern dabei minOccurs und maxOccurs.
  -->

```

Um nun nicht für alle Extended-Typen all die Datei damit auch noch zu befüllen, werden diese Typen hier deklariert und in den entsprechenden Dateien nur noch referenziert.

Damit sollen alle Extended-Typen möglichst zentral in dieser Datei definiert werden.

```
-->

<!-- [OSCIlauf] {Req.1}
  AcceptDelivery.xsd start -->
<xsd:complexType name="ProCardExt_AcceptDelType">
  <xsd:annotation>
    <xsd:documentation>
      ProcessCard für eine AcceptDelivery.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:complexContent>
    <xsd:restriction base="osciLauf2026:ProCardExtTemplate">
      <xsd:sequence>
        <xsd:element name="Creation" type="osci:TimestampType"/>
        <xsd:element name="Forwarding" type="osci:TimestampType"
minOccurs="1"/>
        <xsd:element name="Subject" type="xsd:string" minOccurs="0"/>
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ProCardExtBndl_AcceptDelType">
  <xsd:annotation>
    <xsd:documentation>
      Erweiterte ProcessCardBundle für eine AcceptDelivery.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:complexContent>
    <xsd:restriction base="osciLauf2026:ProCardExtBndlTemplate">
      <xsd:sequence>
        <xsd:element name="MessageId" type="osci:MessageIdType"/>
        <xsd:element name="ProCardExt"
type="osciLauf2026:ProCardExt_AcceptDelType"/>
        <xsd:element name="ValidationReport"
type="osciLauf2026:ValidationReportsType" minOccurs="0"/>
        <xsd:element name="TransportEncryptionInspection"
type="osciLauf2026:TransportEncryptionInspectionType"
minOccurs="0"/>
        <xsd:element name="TransportSignatureInspection"
type="osciLauf2026:TransportSignatureInspectionType"
minOccurs="0"/>
        <xsd:element name="Extensions" minOccurs="0"
type="osciExt:ExtensionsType"/>
      </xsd:sequence>

      <xsd:attribute name="Id" type="xsd:ID" use="required"/>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="SignedProCardExtBndl_AcceptDelType">
  <xsd:annotation>
    <xsd:documentation>
      Signierte, erweiterte ProcessCardBundle für eine AcceptDelivery.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:sequence>
```

```

        <xsd:element name="ProCardExtBndl"
type="osciLauf2026:ProCardExtBndl_AcceptDelType"
            minOccurs="1" maxOccurs="1"/>
        <xsd:element name="signature" type="ds:SignatureType"
            minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<!-- [OSCIlauf] {Req.1}
    AcceptDelivery.xsd ende -->

<!-- [OSCIlauf] {Req.1}
    ProcessDelivery.xsd start -->
<xsd:complexType name="ProCardExt_ProcessDelType">
    <xsd:annotation>
        <xsd:documentation>
            ProcessCard für eine ProcessDelivery.
        </xsd:documentation>
    </xsd:annotation>

    <xsd:complexContent>
        <xsd:restriction base="osciLauf2026:ProCardExtTemplate">
            <xsd:sequence>
                <xsd:element name="Creation" type="osci:TimestampType"/>
                <xsd:element name="Forwarding" type="osci:TimestampType"
minOccurs="1"/>
                <xsd:element name="Subject" type="xsd:string" minOccurs="0"/>
            </xsd:sequence>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ProCardExtBndl_ProcessDelType">
    <xsd:annotation>
        <xsd:documentation>
            Erweiterte ProcessCardBundle für eine ProcessDelivery.
        </xsd:documentation>
    </xsd:annotation>

    <xsd:complexContent>
        <xsd:restriction base="osciLauf2026:ProCardExtBndlTemplate">
            <xsd:sequence>
                <xsd:element name="MessageId" type="osci:MessageIdType"/>
                <xsd:element name="ProCardExt"
type="osciLauf2026:ProCardExt_ProcessDelType"/>

                <xsd:element name="ValidationReport"
type="osciLauf2026:ValidationReportsType" minOccurs="0"/>

                <xsd:element name="TransportEncryptionInspection"
type="osciLauf2026:TransportEncryptionInspectionType"
                    minOccurs="0"/>
                <xsd:element name="TransportSignatureInspection"
type="osciLauf2026:TransportSignatureInspectionType"
                    minOccurs="0"/>

                <xsd:element name="Extensions" minOccurs="0"
type="osciExt:ExtensionsType"/>
            </xsd:sequence>

            <xsd:attribute name="Id" type="xsd:ID" use="required"/>

        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="SignedProCardExtBndl_ProcessDelType">
    <xsd:annotation>
        <xsd:documentation>

```

```

        Signierte erweiterte ProcessCardBundle für eine ProcessDelivery.
    </xsd:documentation>
</xsd:annotation>

    <xsd:sequence>
        <xsd:element name="ProCardExtBndl"
type="osciLauf2026:ProCardExtBndl_ProcessDelType" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="signature" type="ds:SignatureType" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<!-- [OSCIlauf] {Req.1} ProcessDelivery.xsd ende -->

<!-- [OSCIlauf] {Req.1}
ResponseToFetchDelivery.xsd start -->
<xsd:complexType name="ProCardExt_RespFetchDelType">
    <xsd:annotation>
        <xsd:documentation>
            ProcessCard für eine ResponseToFetchDelivery.
        </xsd:documentation>
    </xsd:annotation>

    <xsd:complexContent>
        <xsd:restriction base="osciLauf2026:ProCardExtTemplate">
            <xsd:sequence>
                <xsd:element name="Creation" type="osci:TimestampType"/>
                <xsd:element name="Forwarding" type="osci:TimestampType"/>
                <xsd:element name="Reception" type="osci:TimestampType"
minOccurs="0"/>
                <xsd:element name="Subject" type="xsd:string" minOccurs="0"/>
            </xsd:sequence>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ProCardExtBndl_RespFetchDelType">
    <xsd:annotation>
        <xsd:documentation>
            Erweiterte ProcessCardBundle für eine ResponseToFetchDelivery.
        </xsd:documentation>
    </xsd:annotation>

    <xsd:complexContent>
        <xsd:restriction base="osciLauf2026:ProCardExtBndlTemplate">
            <xsd:sequence>
                <xsd:element name="MessageId" type="osci:MessageIdType"/>
                <xsd:element name="ProCardExt"
type="osciLauf2026:ProCardExt_RespFetchDelType"/>

                <xsd:element name="ValidationReport"
type="osciLauf2026:ValidationReportsType" minOccurs="0"/>

                <xsd:element name="TransportEncryptionInspection"
type="osciLauf2026:TransportEncryptionInspectionType"
minOccurs="0"/>
                <xsd:element name="TransportSignatureInspection"
type="osciLauf2026:TransportSignatureInspectionType"
minOccurs="0"/>

                <xsd:element name="Extensions" minOccurs="0"
type="osciExt:ExtensionsType"/>
            </xsd:sequence>

            <xsd:attribute name="Id" type="xsd:ID" use="required"/>

        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

```

```

    <xsd:complexType name="SignedProCardExtBndl_RespFetchDelType">
      <xsd:annotation>
        <xsd:documentation>
          Signierte erweiterte ProcessCardBundle für eine
ResponseToFetchDelivery.
        </xsd:documentation>
      </xsd:annotation>

      <xsd:sequence>
        <xsd:element name="ProCardExtBndl"
type="osciLauf2026:ProCardExtBndl_RespFetchDelType" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="signature" type="ds:SignatureType" minOccurs="0"
maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  <!-- [OSCIlauf] {Req.1}
    ResponseToFetchDelivery.xsd ende -->

  <!-- [OSCIlauf] {Req.1}
    ResponseToForwardDelivery.xsd start -->
  <xsd:complexType name="ProCardExt_RespForwDelType">
    <xsd:complexContent>
      <xsd:restriction base="osciLauf2026:ProCardExtTemplate">
        <xsd:sequence>
          <xsd:element name="Creation" type="osci:TimestampType"/>
          <xsd:element name="Forwarding" type="osci:TimestampType"
minOccurs="0"/>
          <xsd:element name="Reception" type="osci:TimestampType"
minOccurs="0"/>
          <xsd:element name="Subject" type="xsd:string" minOccurs="0"/>
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="ProCardExtBndl_RespForwDelType">
    <xsd:annotation>
      <xsd:documentation>
        Erweiterte ProcessCardBundle für eine ResponseToForwardDelivery.
      </xsd:documentation>
    </xsd:annotation>

    <xsd:complexContent>
      <xsd:restriction base="osciLauf2026:ProCardExtBndlTemplate">
        <xsd:sequence>
          <xsd:element name="MessageId" type="osci:MessageIdType"/>
          <xsd:element name="ProCardExt"
type="osciLauf2026:ProCardExt_RespForwDelType"/>
          <xsd:element name="ValidationReport"
type="osciLauf2026:ValidationReportsType" minOccurs="0"/>
          <xsd:element name="TransportEncryptionInspection"
type="osciLauf2026:TransportEncryptionInspectionType"
minOccurs="0"/>
          <xsd:element name="TransportSignatureInspection"
type="osciLauf2026:TransportSignatureInspectionType"
minOccurs="0"/>
          <xsd:element name="Extensions" minOccurs="0"
type="osciExt:ExtensionsType"/>
        </xsd:sequence>

        <xsd:attribute name="Id" type="xsd:ID" use="required"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

```

```

<xsd:complexType name="SignedProCardExtBndl_RespForwDelType">
  <xsd:annotation>
    <xsd:documentation>
      Signierte erweiterte ProcessCardBundle für eine
ResponseToForwardDelivery.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:sequence>
    <xsd:element name="ProCardExtBndl_ResponseToForwardDelType"
type="osciLauf2026:ProCardExtBndl_RespForwDelType" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="signature" type="ds:SignatureType" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<!-- [OSCIlauf] {Req.1}
ResponseToForwardDelivery.xsd ende -->

<!-- [OSCIlauf] {Req.1}
ResponseToMediateDelivery.xsd start
Es gibt in Request und Reply
je eine Struktur für ProcessCard und ProcessCardBundle.
Diese müssen so erweitert werden, dass sie jeweils auch mit der Extended-
Version
umgehen können.
-->

<!-- Request -->
<xsd:complexType name="ProCardExt_ReqProCardType">
  <xsd:annotation>
    <xsd:documentation>
      ProcessCard für einen Request innerhalb einer MediateDelivery.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:complexContent>
    <xsd:restriction base="osciLauf2026:ProCardExtTemplate">
      <xsd:sequence>
        <xsd:element name="Creation" type="osci:TimestampType"/>
        <xsd:element name="Forwarding" type="osci:TimestampType"
minOccurs="1"/>
        <xsd:element name="Reception" type="osci:TimestampType"
minOccurs="1"/>
        <xsd:element name="Subject" type="xsd:string" minOccurs="0"/>
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ProCardExtBndl_ReqProCardType">
  <xsd:annotation>
    <xsd:documentation>
      Erweiterte ProcessCardBundle für einen Request innerhalb einer
MediateDelivery.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:complexContent>
    <xsd:restriction base="osciLauf2026:ProCardExtBndlTemplate">
      <xsd:sequence>
        <xsd:element name="MessageId" type="osci:MessageIdType"/>
        <xsd:element name="ProCardExt"
type="osciLauf2026:ProCardExt_ReqProCardType"/>
        <xsd:element name="ValidationReport"
type="osciLauf2026:ValidationReportsType" minOccurs="0"/>
        <xsd:element name="Extensions" minOccurs="0"
type="osciExt:ExtensionsType"/>
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

```

```

        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="SignedProCardExtBndl_ReqProCardType">
    <xsd:annotation>
      <xsd:documentation>
        Signierte erweiterte ProcessCardBundle für einen Request innerhalb
einer MediateDelivery.
      </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          Optional signiertes erweitertes/ neues ProcessCardBundle
(ProcessCardExtendedBundle) .
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="ProCardExtBndl"
type="osciLauf2026:ProCardExtBndl_ReqProCardType" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="signature" type="ds:SignatureType" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

  <!-- Reply -->
  <xsd:complexType name="ProCardExt_ReplyProCardType">
    <xsd:annotation>
      <xsd:documentation>
        ProcessCard für eine Reply innerhalb einer MediateDelivery.
      </xsd:documentation>
    </xsd:annotation>

    <xsd:complexContent>
      <xsd:restriction base="osciLauf2026:ProCardExtTemplate">
        <xsd:sequence>
          <xsd:element name="Creation" type="osci:TimestampType"/>
          <xsd:element name="Forwarding" type="osci:TimestampType"
minOccurs="1"/>
          <xsd:element name="Subject" type="xsd:string" minOccurs="0"/>
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="ProCardExtBndl_ReplyProCardType">
    <xsd:annotation>
      <xsd:documentation>
        Erweiterte ProcessCardBundle für eine Reply innerhalb einer
MediateDelivery.
      </xsd:documentation>
    </xsd:annotation>

    <xsd:complexContent>
      <xsd:restriction base="osciLauf2026:ProCardExtBndlTemplate">
        <xsd:sequence>
          <xsd:element name="MessageId" type="osci:MessageIdType"/>
          <xsd:element name="ProCardExt"
type="osciLauf2026:ProCardExt_ReplyProCardType"/>
          <xsd:element name="ValidationReport"
type="osciLauf2026:ValidationReportsType" minOccurs="0"/>
          <xsd:element name="TransportEncryptionInspection"
type="osciLauf2026:TransportEncryptionInspectionType"
minOccurs="0"/>
          <xsd:element name="TransportSignatureInspection"

```

```

type="osciLauf2026:TransportSignatureInspectionType"
    minOccurs="0"/>

    <xsd:element name="Extensions" minOccurs="0"
type="osciExt:ExtensionsType"/>
    </xsd:sequence>
    </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="SignedProCardExtBndl_ReplyProCardType">
    <xsd:annotation>
        <xsd:documentation>
            Signierte erweiterte ProcessCardBundle für eine Reply innerhalb
einer MediateDelivery.
        </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
        <xsd:element name="ProCardExtBndl"
type="osciLauf2026:ProCardExtBndl_ReplyProCardType" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="signature" type="ds:SignatureType" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<!-- [OSCIlauf] {Req.1}
    ResponseToMediateDelivery.xsd ende -->

<!-- ResponseToPartialStoreDelivery.xsd start
    keine Änderung/ Erweiterung notwendig, da auf ResponseToStoreDelivery
aufbauend.
-->
<!--
<xsd:element name="ProcessCardBundleExtended"
type="osciLauf2026:ProCardExtBndlTemplate"/>
-->
<!-- ResponseToPartialStoreDelivery.xsd ende -->

<!-- ResponseToPartialStoreDeliveryOldNS.xsd start
    keine Änderung/ Erweiterung notwendig, da auf ResponseToStoreDeliveryOldNS
aufbauend.
-->
<!-- ResponseToPartialStoreDeliveryOldNS.xsd ende -->

<!-- [OSCIlauf] {Req.1}
    ResponseToStoreDelivery.xsd start -->
<xsd:complexType name="ProCardExt_RespStoreDelType">
    <xsd:annotation>
        <xsd:documentation>
            ProcessCard für eine ResponseToStoreDelivery.
        </xsd:documentation>
    </xsd:annotation>

    <xsd:complexContent>
        <xsd:restriction base="osciLauf2026:ProCardExtTemplate">
            <xsd:sequence>
                <xsd:element name="Creation" type="osci:TimestampType"/>
                <xsd:element name="Subject" type="xsd:string" minOccurs="0"/>
            </xsd:sequence>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ProCardExtBndl_RespStoreDelType">
    <xsd:annotation>
        <xsd:documentation>

```

```

        Erweiterte ProcessCardBundle für eine ResponseToStoreDelivery.
    </xsd:documentation>
</xsd:annotation>

    <xsd:complexContent>
        <xsd:restriction base="osciLauf2026:ProCardExtBndlTemplate">
            <xsd:sequence>
                <xsd:element name="MessageId" type="osci:MessageIdType"/>
                <xsd:element name="ProCardExt"
type="osciLauf2026:ProCardExt_RespStoreDelType"/>

                <xsd:element name="ValidationReport"
type="osciLauf2026:ValidationReportsType" minOccurs="0"/>

                <xsd:element name="TransportEncryptionInspection"
type="osciLauf2026:TransportEncryptionInspectionType"
                    minOccurs="0"/>
                <xsd:element name="TransportSignatureInspection"
type="osciLauf2026:TransportSignatureInspectionType"
                    minOccurs="0"/>

                <xsd:element name="Extensions" minOccurs="0"
type="osciExt:ExtensionsType"/>
            </xsd:sequence>

            <xsd:attribute name="Id" type="xsd:ID" use="required"/>

        </xsd:restriction>
    </xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="SignedProCardExtBndl_RespStoreProCardType">
    <xsd:annotation>
        <xsd:documentation>
            Signierte erweiterte ProcessCardBundle für eine
ResponseToStoreDelivery.
        </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
        <xsd:element name="ProCardExtBndl"
type="osciLauf2026:ProCardExtBndl_RespStoreDelType" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="signature" type="ds:SignatureType" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<!-- [OSCIlauf] {Req.1}
ResponseToStoreDelivery.xsd ende -->

</xsd:schema>

```

7. 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
- [ETSI] „ETSI TS 119 102-2 V1.4.1 (2023-06)- Electronic Signatures and Infrastructures (ESI); Procedures for Creation and Validation of AdES Digital Signatures; Part 2: Signature Validation Report“, Juni 2023, Download: https://www.etsi.org/deliver/etsi_ts/119100_119199/11910202/01.04.01_60/ts_11910202v010401p.pdf
- [EUR-LEX] Verordnung (EU) Nr. 910/2014 des Europäischen Parlaments und des Rates vom 23. Juli 2014 über elektronische Identifizierung und Vertrauensdienste für elektronische Transaktionen im Binnenmarkt und zur Aufhebung der Richtlinie 1999/93/EG, August 2014, Download: <https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=CELEX:32014R0910>