E -VLOŽIŠČE

SISTEM ZA VARNO ELEKTRONSKO VROČANJE

Aplikacija ebms-sed
(ebMS 3.0 Secure e-delivery)

1.0

KONTROLA VERZIJ

ZADNJA VERZIJA:

Verzija	1.0
Datum	01.11.2015
Avtor	Jože Rihtaršič
Odgovornost	Jože Rihtaršič
Zaupnost	/
Datoteka	

ZGODOVINA:

Verzija	Datum	Avtor	Opis
1.0	01.11.2015	Jože Rihtaršič	Dokument kreiran.

REVIZIJE:

Revizija	Datum	Avtor	Opis

ZAŠČITA DOKUMENTA

Kazalo vsebine

Namen dokumenta

Dokument opisuje tehnične značilnosti aplikativnih rešitev ebms-sed aplikacije. Namenjen je tehničnemu osebju kot pomoč pri razvoju, namestitvi in vzdrževanju aplikacije ebms-sed.

Dokument vsebuje logični model in fizično arhitekturo sistema, opis podatkovnega modela in spletnih storitev aplikacije ebms-sed ter nekatere karakteristike, ki so vezane na arhitekturo sistema. V logičnem modelu so predstavljeni moduli ebms-sed aplikacije. V fizični arhitekturi je predstavljeno kako je aplikacija postavljena na ciljnem okolju.

Namen aplikacije ebms-sed

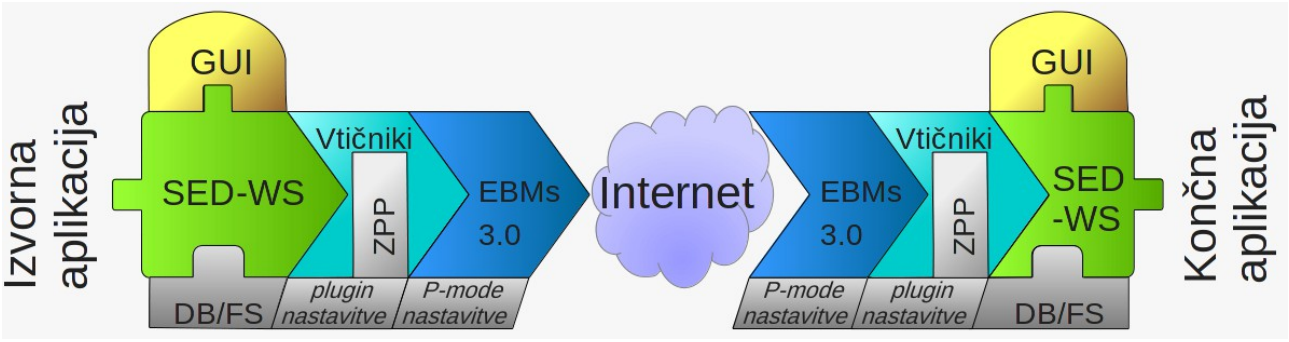
Elektronska izmenjava podatkov med podjetji in organizacijami ni novost v modernem poslovanju. V preteklosti je nastalo veliko standardov (tako v Sloveniji kot v tujini), ki urejajo elektronsko izmenjavo podatkov in dokumentov. Starejši standardi so večinoma namenski in se osredotočajo na obliko zapisa in nabor podatkov, ki jih vsebujejo sporočila. Ker so bile integracije informacijskih sistemov izdelane po meri in v omejenem številu, problem varnosti ni bil tako izrazit. Z rastjo števila partnerjev in namenov elektronske izmenjave se določanje vsebine prenosa (datoteke, infoseti podatkov potrebni za strojno obdelavo itd..) definira na višjem nivoju, standardi za elektronsko izmenjavo pa se osredotočajo na zagotavljanje avtentičnosti sporočil, zagotavljanje identifikacij naslovnika in prejemnika, zagotavljanje zanesljivosti in dokazljivosti prenosa itd.

OASIS ebMS 3.0 standard predstavlja korak naprej na področju uporabe spletnih storitev pri B2B izmenjavi podatkov in dokumentov. OASIS ebMS 3.0 uporablja obstoječe standarde spletnih storitev WS-* in jih združuje v enotno in celovito specifikacijo za varno in zanesljivo izmenjavo dokumentov. Standard ebMS 3.0 bazira na standardih SOAP 1.1, SOAP 1.2, SOAP s priponkami , WS - Security 1.0, 1.1, WS-Addressing WS - Reliability 1.1 in WS - ReliableMessaging 1.2. Standard tako vsebuje navodila za uporabo obstoječih spletnih standardov ter navodila za pakiranje sporočila in priponk, skupaj z opredelitvami sporočanja koreografij za izmenjave dokumentov .
Vrhovno sodišče RS je pri nadgradnji obstoječega protokola SVEV 1.0, ki se trenutno uporablja za elektronsko vročanje sodnih odpravkov, upoštevalo trende uporabe tehnologij čezmejnega elektronskega vročanja, ki so usmerjeni v uporabo OASIS standarda ebMS 3.0. Uporaba protokola je omogočila odpravo vseh „po-meri“ narejenih shem, ki se trenutno uporabljajo v specifikacijah „SVEV 1.0“ za pakiranje in varnost sporočil, ter jih nadomestila s standardi ebMS 3.0. Standardizirano pakiranje sporočil, zagotavljanje varnostni in zanesljivosti je omogočilo uporabo ogrodij, ki podpirajo ebMS 3.0 standarde oz. WS-* sklade. Aplikacija SED je primer implementacije protokola SVEV 2.0, ki bazira na ebMS 3.0 standardu. Namen aplikacije je strankam, izvajalcem e-storitev sodišča in ponudnikom sistema za varno e-vročanje olajšati integracijo z informacijskimi sistemi sodišča.

Logični model

Aplikacija ebms-sed je sestavljena iz modulov:

- **modul SED-WS:** spletni servisi (WSDL) namenjeni integraciji z zalednimi sistemi;
- **modul vtičniki:** modul omogoča dodajanje vtičnikov za izvajanje „poslovnih tokov“ v storitvah vročanja, ter možnost izvajanja dodatnih kontrol dohodnih sporočil;
- **modul ebMS 3.0:** implementacija ebMS 3.0 standarda;
- **modul GUI:** Grafični vmesnik za nadzor in spremljanje dohodne in izhodne pošte.



Slika 1: Logični model.

Modul SED-WS

Namen modula je izdelava preprostega spletnega vmesnika za pošiljanje izhodne in prevzemanje dohodne pošte v zaledne sisteme. Naloga modula je tudi izdelava evidence izhodnih in dohodnih pošiljk ter beleženje in hranjenje vseh dogodkov na izhodni in dohodni pošti. Dodatno je pri izhodni pošti naloga preverjanje skladnosti sporočil s „p-mode“ nastavitvami. Pri tem se preverja obstoj p-mode konfiguracije za naslovnika, poslovni kontekst „storitev, akcija“, vsebovanost metapodatkov in vsebin, ki jih sporočilo mora vsebovati.

Modul vtičniki

Namen modula je omogočiti poljubno poslovno logiko pri preverjanju dohodne in izhodne pošte ter omogočiti avtomatizirano izvajanje „poslovnih tokov“ v posameznih storitvah. Kot primer je priložen vtičnik, ki skrbi za elektronsko vročanje v storitvi ZPPDelivery. Kjer pri izhodni pošti zgenerira ključ za simetrično šifriranje ter šifrira vse vsebine. Ob sprejemu povratnice vrne ključ za dešifriranje pošiljatelju povratnice. Na drugi strani ob sprejemu pošiljke s storitvijo ZPPDelivery avtomatično izdela povratnico, jo podpiše s naslovnikovim certifikatom ter jo posreduje pošiljatelju pošiljke. Hkrati naslovniku posreduje dohodno pošiljko „Obvestilo o prispeli pošiljki“. Ob sprejemu ključa za dešifriranje, dešifrira dohodno pošiljko in jo preko SED-WS v dešifrirani obliki omogoči za prevzem v zaledne sisteme.

Pri izhodni pošti sta modula SED-WS in modul „vtičniki“ povezana preko JMS tehnologije. Tako lahko administrator določa število „worker-jev“, ki pošiljajo pošiljke v ciljne sisteme.

Modul ebMS 3.0

Namen modula je implementacija ebMS 3.0 standarda za varno in zanesljivo vročanje. Modul glede na p-mode konfiguracijo zapakira sporočilo ebms 3.0 obliko, jo podpiše/kriptira, opremi s parametri za zanesljiv prenos ter jo posreduje v naslovnikov sistem. Na naslovnikovi strani preveri veljavnost popisa, avtenticira pošiljatelja ter poišče ustrezno p-mode konfiguracijo. (Pravila izmenjave ti. „e-delivery agreement“ se nahaja v p-mode konfiguraciji, ki mora biti usklajena med pošiljateljem in prejemnikom). V primeru uspešnega sprejema se dohodno sporočilo preko modula „SED-WS“ omogoči na prevzem v zaledne sisteme. V primeru neveljavnega podpisa in neskladja s p-mode konfiguracijo pošiljatelju vrne ustrezno EBMS 3.0 napako.

Modul GUI – grafični vmesnik

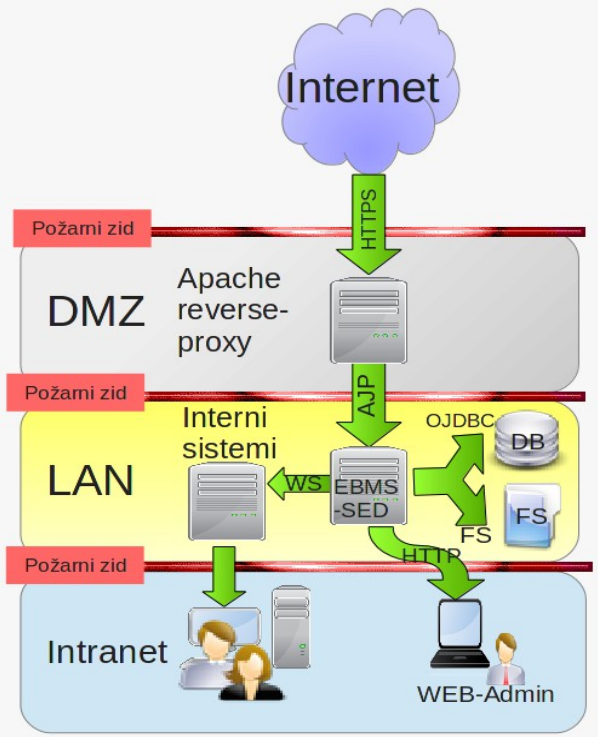
Grafični vmesnik je namenjen za administracijo ebms-sed aplikacije ter za pregled izhodne in dohodne pošte za posamezne predale. Administracija ebms-sed vsebuje izdelavo in urejanje posameznih p-mode nastavitev., dodajanje novih predalov ter nastavitev za spremljanje delovanja aplikacije ebms-sed. V primeru pregleda izhodno dohodne pošte za posamezne e-predale omogoča pregled in iskanje izhodne in dohodne pošte ter pregled vseh dogodkov pri vročanju oz. prejemu pošte.

Fizična arhitektura

Aplikacija ebms-sed je JEE aplikacija, ki za delovanje potrebuje aplikacijski strežnik. Metapodatki se preko JPA 2.0 in hibernate hranijo v relacijski bazi. Binarne vsebine se za čas trajanja prenosa hranijo na lokalnem datotečnem sistemu. Ko je izhodna pošiljka uspešno dostavljena v naslovnikov sistem se vsebine avtomatsko izbrišejo.
V primeru dohodne pošte se ob prevzemu pošiljke v zaledni sistem, se binarne vsebine avtomatsko izbrišejo. V relacijski bazi pa se hranijo vsi metapodatki in dogodki na posamezni pošiljki.

Aplikacija je bila razvita in testirana z uporabo tehnologij:

- aplikacijskega strežnik: wildfly 9.0.2.Final
- Podatkovna baza: postgresql
- java: JDK 1.8.x



Slika 2: Fizična arhitektura.

Projekt ebms-sed

Projekt ebms-sed je izdelan z maven tehnologijo, kar omogoča CI (Continuous integration) razvoj ter uporabo različnih IDE, ki podpirajo maven projekte. Aplikacija je sestavljena modularno in sledi logičnemu modelu predstavljenem na začetku tega dokumenta.

Sestava embs-sed projekta:

- **ebms-sed: o**snovni projekt določa vse plugin-e in verzije odvisnih knjižnic;
  - **ebms-sed-libs**: skupne knjižnice, ki jih uporabljajo posamezni moduli;
    - **ebms-msh-xsd:** modul generira java objekte iz ebMS 3.0 shem. SOAP_1.1.xsd (http://schemas.xmlsoap.org/soap/envelope/) ,SOAP_1.2.xsd (http://www.w3.org/2003/05/soap-envelope/), xml.xsd (http://www.w3.org/2001/03/xml.xsd) , ebms-header-3_0-200704.xsd (http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/core/ebms-header-3_0-200704.xsd) , ebbp-signals-2.0.xsd (http://docs.oasis-open.org/ebxml-bp/2.0.4/ebbp-signals-2.0.4.xsd) , xenc-schema.xsd (http://www.w3.org/TR/xmlenc-core/xenc-schema.xsd) , xmldsig-core-schema.xsd (http://www.w3.org/TR/xmldsig-core/xmldsig-core-schema.xsd) , xlink.xsd (http://www.w3.org/TR/xml-i18n-bp/xmlspec/xlink.xsd);

    - **ebms-sed-wsdl**: modul generira java klienta in WS-API iz spletnega opisa (wsdl) SED-WS modula.
    - **ebms-sed-commons**: skupna orodja in šifiranti.
- **ebms-sed-module**: implementacija ebms-sed spletnih storitev
- **ebms-sed-web**: implementacija web-gui vmesnika.
- **ebms-msh-module**: implementacija ebMS 3.0. modula

Podatkovni model

Podatkovni model je razdeljen na dva dela: podatkovne strukture za izhodno pošto imajo predpono „SED_OUTBOX“ ter podatkovne strukture za dohodno pošto imajo predpono „SED_INBOX“

Tabela SED_OUTBOX

Tabela vsebuje metapodatke o izhodni pošti.

Stolpec	tip	št. znakov	Opis
id	številka	/	Enolična oznaka izhodne pošiljke (generira bazna sekvenca).
msg_id ,	besedilo	64	ebms 3.0 oznaka sporočila.
sender_msg_id	besedilo	64	oznaka izhodne pošiljatelja.
service	besedilo	64	ebms 3.0 storitev.
action	besedilo	64	ebms 3.0 akcija.
conv_id	besedilo	64	ebms 3.0 storitev.
subject	besedilo	512	Opis sporočila.
receiver_ebox	besedilo	64	Elektronski SVEV naslov prejemnika.
receiver_name	besedilo	128	Naziv prejemnika.
sender_ebox	besedilo	64	Elektronski SVEV naslov pošiljatelja.
sender_name	besedilo	128	Naziv pošiljatelja.
status	besedilo	32	Status pošiljke.
date_status	datum	/	Datum spremembe statusa
date_submited	datum	/	Datum sprejema pošiljke v vročanje.
date_sent	datum	/	Datum uspešne oddaje pošiljke v naslovnikov MSH.
date_received	datum	/	Datum sprejema pošiljke v naslovnikov MSH. Navadno je date_sent enak date_received, vendar to ne velja, kadar se vroča preko posrednikov t. Im. ebms multihop.
date_delivered	datum	/	Datum prevzema pošiljke v naslovnikov zaledni sistem oz. datum vročitve.

Tabela SED_OUTBOX_EVENT

Tabela vsebuje metapodatke o dogodkih na izhodni pošiljki.

Stolpec	tip	št. znakov	Opis
id	številka	/	Enolična oznaka dogodka (generira bazna sekvenca).
mail_id	številka	/	Referenca na izhodno pošiljko SED_OUTBOX.ID
sender_msg_id	besedilo	64	Referenca na izhodno pošiljko SED_OUTBOX.sender_msg_id
status	besedilo	32	Status pošiljke, ki ga je povzročil dogodek
date	date	/	Čas dogodka
description	besedilo	512	Opis dogodka
user_id	besedilo	64	Id uporabnika, ki je povzročil dogodek. (Pošiljanje, prevzem,brisanje...)
application_id	besedilo	64	Id aplikacije, preko katere se je izvršil dogodek.

Tabela:SED_OUTBOX_PAYLOAD

Tabela vsebuje metapodatke o vsebini pošiljke (priponke).

Stolpec	tip	št. znakov	Opis
id	številka	/	Enolična oznaka vsebine/priponke (generira bazna sekvenca).
ebms_id	besedilo	128	Oznaka v ebms sporočilu.
mail_id	številka	/	Referenca na izhodno pošiljko SED_OUTBOX.ID
name	besedilo	128	Naziv priponke.
description	besedilo	512	Opis priponke.
type	besedilo	64	Tip priponke.
filename	besedilo	64	Naziv datoteke.
filepath	besedilo	1028	Pot do priponke na lokalnem strežniku.
mime_type	besedilo	128	Mimetip priponke.
encoding	besedilo	128	Kodiranje priponke (UTF-8, base64, binary).
md5	besedilo	32	MD5 zgostitvena vrednost priponke

Tabela SED_OUTBOX_PROPERTY

Tabela vsebuje dodatne lastnosti izhodne pošiljke, ki olajšajo sprejem pri naslovniku. Dodatne lastnosti so dogovor med pošiljateljem in prejemnikom in so določene v P-Mode konfiguraciji.

Stolpec	tip	št. znakov	Opis
id	številka	/	Enolična oznaka vsebine/priponke (generira bazna sekvenca).
mail_id	številka	/	Referenca na izhodno pošiljko SED_OUTBOX.ID

name	besedilo	128	Naziv lastnosti.
value	besedilo	512	Lastnost.

Tabela SED_INBOX

Tabela vsebuje metapodatke o dohodni pošiljki.

Stolpec	tip	št. znakov	Opis
id	številka	/	Enolična oznaka izhodne pošiljke (generira bazna sekvenca).
msg_id ,	besedilo	64	ebms 3.0 oznaka sporočila.
sender_msg_id	besedilo	64	oznaka izhodne pošiljatelja.
service	besedilo	64	ebms 3.0 storitev.
action	besedilo	64	ebms 3.0 akcija.
conv_id	besedilo	64	ebms 3.0 storitev.
subject	besedilo	512	Opis sporočila.
receiver_ebox	besedilo	64	Elektronski SVEV naslov prejemnika.
receiver_name	besedilo	128	Naziv prejemnika.
sender_ebox	besedilo	64	Elektronski SVEV naslov pošiljatelja.
sender_name	besedilo	128	Naziv pošiljatelja.
status	besedilo	32	Status pošiljke.
date_status	datum	/	Datum spremembe statusa
date_submited	datum	/	Datum sprejema pošiljke v vročanje.
date_sent	datum	/	Datum uspešne oddaje pošiljke v naslovnikov MSH.
date_received	datum	/	Datum sprejema pošiljke v naslovnikov MSH. Navadno je date_sent enak date_received, vendar to ne velja, kadar se vroča preko posrednikov t. Im. ebms multihop.
date_delivered	datum	/	Datum prevzema pošiljke v naslovnikov zaledni sistem oz. datum vročitve.

Tabela SED_INBOX_EVENT

Tabela vsebuje metapodatke o dogodkih na izhodni pošiljki.

Stolpec	tip	št. znakov	Opis
id	številka	/	Enolična oznaka dogodka (generira bazna sekvenca).
mail_id	številka	/	Referenca na izhodno pošiljko SED_INBOX.ID
status	besedilo	32	Status pošiljke, ki ga je povzročil dogodek
date	date	/	Čas dogodka
description	besedilo	512	Opis dogodka

user_id	besedilo	64	Id uporabnika, ki je povzročil dogodek. (Pošiljanje, prevzem,brisanje...)
application_id	besedilo	64	Id aplikacije, preko katere se je izvršil dogodek.

Tabela SED_INBOX_PAYLOAD

Tabela vsebuje metapodatke o vsebini pošiljke (priponke).

Stolpec	**tip**	**št. znakov**	**Opis**
id	številka	/	Enolična oznaka vsebine/priponke (generira bazna sekvenca).
ebms_id	besedilo	128	Oznaka v ebms sporočilu.
mail_id	številka	/	Referenca na izhodno pošiljko SED_INBOX.ID
name	besedilo	128	Naziv priponke.
description	besedilo	512	Opis priponke.
type	besedilo	64	Tip priponke.
filename	besedilo	64	Naziv datoteke.
filepath	besedilo	1028	Pot do priponke na lokalnem strežniku.
mime_type	besedilo	128	Mimetip priponke.
encoding	besedilo	128	Kodiranje priponke (UTF-8, base64, binary).
md5	besedilo	32	MD5 zgostitvena vrednost priponke

Tabela SED_INBOX_PROPERTY

Tabela vsebuje dodatne lastnosti izhodne pošiljke, ki olajšajo sprejem pri naslovniku. Dodatne lastnosti so dogovor med pošiljateljem in prejemnikom in so določene v P-Mode konfiguraciji.

Stolpec	**tip**	**št. znakov**	**Opis**
id	številka	/	Enolična oznaka vsebine/priponke (generira bazna sekvenca).
mail_id	številka	/	Referenca na izhodno pošiljko SED_INBOX.ID
name	besedilo	128	Naziv lastnosti.
value	besedilo	512	Lastnost.

Spletni vmesnik

Namen spletnega vmesnika je omogočiti izdelavo integracije ebms-sed z zalednimi sistemi. Metode omogočajo pošiljanje izhodne, prevzemanje dohodne pošte ter poizvedovanju po statusih posameznih pošiljk. Vsi klici spletnih morajo vsebovati enolično oznako aplikacije, ki kliče servis ter enolično oznako uporabnika, ki je povzročil klic servisa. Podrobnejši opisi posameznih parametrov so v WSDL-opisu spletnega servisa.

Odprava pošte

Metoda: submitMail.

Namen metode je oddaja izhodne pošiljke v vročanje. Vhodni parameter je infoset podatkov „OutMail“, ki vsebuje pošiljatelja, naslovnika, opis vsebine, priponke in dodatne parametre pošiljke.

Seznam izhodne pošte

Metoda: getOutMailList.

Namen metode je pregled izhodne pošte. Obvezen podatek je pošiljateljev predal. Dodatni parametri iskanja so: interval oddaje pošiljke v odpravo, naslovnik, status pošiljke itd.

Seznam dogodkov na izhodni pošiljki

Metoda: getOutMailEvents.

Namen metode je pridobivanje seznama dogodkov na izhodni pošiljki . Seznam dogodkov služi kot revizijska sled in se lahko uporablja za sledenje pošiljke.

Zadnji spremenjeni statusi

Metoda: getLastOutMailEvents.

Namen metode je posodabljanje statusov izhodne pošte v izvornih aplikacijah. Metoda vrača spremenjene statuse v določenem časovnem obdobju.

Seznam dohodne pošte

Metoda: getInMailList.

Namen metode je pridobivanje seznama dohodne pošte. Na podlagi vrnjenega seznama ciljna aplikacija lahko kontrolirano prevzema dohodno pošto. V infosetu vrnjenih podatkov ni binarnih vsebin dohodne pošte.

Pridobi dohodno pošiljko

Metoda: getInMail.

Namen metode je pridobitev dohodne pošiljke z binarnimi vsebinami.

Spremeni dohodno pošto

Metoda: modifyInMail.

Namen metode je posodabljanje statusov dohodne pošte. V primer uspešnega prevzema v ciljni sistem se pošiljka označi kot prevzeta.

Seznam dogodkov na dohodni pošiljki

Metoda: getInMailEvents.

Namen metode je pridobivanje seznama dogodkov na dohodni pošiljki. Seznam dogodkov služi kot revizijska sled.

ebMS p-mode nastavitve

Standard ebMS 3.0 predpisuje nabor parametrov t. im. P-Mode parametri (angl. *Processing Mode*), s katerimi lahko določamo nivo in parametre varnosti in zanesljivosti transporta ter način naslavljanja in preverjanja tipov priponk v sporočilu. Parametri P-Mode so razporejeni v šest vsebinsko povezanih kategorij (OASIS, 2007b):

- **splošni parametri**, kot so enolična oznaka konfiguracije, referenca na pogodbo za izmenjavo dokumentov, identifikator naslovnika/prejemnika sporočila ter vloge pri izmenjavi dokumentov;
- **protokol**: določa spodaj ležeči protokol izmenjave (HTTP, SMTP, FTP) ter naslov (URL ali email) prejemnikovega MSH;
- **poslovni kontekst**: določa namen, storitev, akcijo in obliko vsebine;
- **napake**: razdelek določa ravnanje in poročanje v primeru napak pri prenosu;
- **zanesljivost prenosa**: parametri določajo uporabo mehanizmov za zagotavljanje zanesljivosti prenosa;
- **varnost**: parametri določajo nivo varnosti, pravila in certifikate za enkripcijo in podpisovanje sporočil.

V nadaljevanju je XSD shema p-mode konfiguracije, ki je bila razvita za aplikacijo ebms-sed. Posazemni parametri so dokumentirani v shemi.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://svev.msh.org/pmode" targetNamespace="http://svev.msh.org/pmode"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation/>
  </xs:annotation>
  <xs:element name="PModes">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="PMode" type="tns:PMode" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="PMode">
    <xs:sequence>
      <xs:element name="Agreement" type="xs:string">
        <xs:annotation>
          <xs:documentation>The reference to the agreement governing this message exchange (maps
to
eb:AgreementRef in message header)</xs:documentation>
        </xs:annotation>
      </xs:element>
```

```
    <xs:element name="MEP" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The type of ebMS MEP associated with this P-Mode. The value must be a URI</xs:documentation>
        <xs:documentation>NOTE: The current version of Holodeck B2B only allows One-Way MEPs so there is just one value.</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="http://www.oasis-open.org/committees/ebxml-msg/one-way"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="MEPbinding" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The transport channel binding assigned to the MEP (push, pull)</xs:documentation>
        <xs:documentation>NOTE: The current version of Holodeck B2B only allows One-Way MEPs so there is just two values for Push and Pull binding.</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="http://www.oasis-open.org/committees/ebxml-msg/push"/>
          <xs:enumeration value="http://www.oasis-open.org/committees/ebxml-msg/pull"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Initiator" type="tns:Party" minOccurs="0">
      <xs:annotation>
        <xs:documentation>This element includes the information on the initiator of the MEP, i.e. the partner that sends the first message. Note that this is not always the sender of the user message because in the pull scenario the first message (the PullRequest) is sent by the partner that will receive the user message. See website and/or chapter 2 of the ebMS V3 Core Specification for more info.

When included the element must contain one or more party ids that identify the trading partner and the business role the partner is acting in.

NOTE: This element is optional in the P-Mode. When not specified the information must be specified when a user message is submitted to Holodeck B2B. Attention should be paid to the fact that the meta-data on submission is expressed in sender and receiver roles which may not be equal to initiator and responder.

</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Responder" type="tns:Party" minOccurs="0">
      <xs:annotation>
        <xs:documentation>PMode.Responder and its subelements are optional if PMode.Initiator is present.) Qualifies the party responding to the initiator party in this MEP. Any user message sent to the responder must have its eb:Messaging/eb:UserMessage/eb:PartyInfo/eb:To element contain the same PartyId elements
```

```
as the PartyId elements defined in this parameter</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Leg" type="tns:Leg" maxOccurs="2"/>
    <xs:element name="ReceptionAwareness" type="tns:ReceptionAwareness" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="id" use="required">
    <xs:annotation>
      <xs:documentation>The identifier for the P-Mode, e.g. the name of the business transaction: PurchaseOrderFromACME. This identifier is user-defined and optional, for the convenience of P-Mode management. It must uniquely identify the P-Mode among all P-Modes deployed on the same MSH, and may be absent if the P-Mode is identified by other means, e.g. embedded in a larger structure that is itself identified, or has parameter values distinct from other P-Modes used on the same MSH. If the ID is specified, the AgreementRef/@pmode attribute value is also expected to be set in associated messages.</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:minLength value="1"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="extends">
    <xs:annotation>
      <xs:documentation>reference to PMode@id which  is base for this pmode</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="isTemplate" type="xs:boolean" default="false"/>
 </xs:complexType>
 <xs:complexType name="Party">
  <xs:sequence>
    <xs:element name="PartyId" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="type">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:minLength value="1"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="Role" type="xs:string">
      <xs:annotation>
        <xs:documentation>Name of the role assumed by the party receiving the first message of this MEP. Either the message element eb:Messaging/eb:UserMessage/eb:PartyInfo/eb:From/eb:Role or the element
```

```
eb:Messaging/eb:UserMessage/eb:PartyInfo/eb:To/eb:Role of each message in
this MEP must have this value, depending on the direction of message transfer</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Authorization" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Describe authorization information for messages
sent by Responder. These parameters need to be matched by a wsse:UsernameToken element in a
message (in a security header only intended for authorization) for this message to be
processed successfully on receiver side – here by Initiator MSH.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:attribute name="username"/>
        <xs:attribute name="password"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
 </xs:complexType>
 <xs:complexType name="Leg">
  <xs:sequence>
    <xs:element name="Protocol" type="tns:Protocol" minOccurs="0"/>
    <xs:element name="BusinessInfo" type="tns:BusinessInfo" minOccurs="0"/>
    <xs:element name="ErrorHandling" minOccurs="0">
      <xs:annotation>
        <xs:documentation>This P-Mode group concerns errors generated by the reception of the
message (for
either a User message or a Signal message, unless indicated otherwise) sent over leg 1
of the MEP.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Report">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="ReceiverErrorsTo" type="xs:string" minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>This parameter indicates the address, or
comma-separated list of addresses, to which to send ebMS errors generated by the MSH that
receives the message in error; e.g. this may be the address of the MSH sending the message in
error</xs:documentation>
                  </xs:annotation>
                </xs:element>
                <xs:element name="AsResponse" type="xs:boolean" minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>This Boolean parameter indicates whether (if
"true") errors generated from receiving a message in error are sent over the back-channel of the
underlying protocol associated with the message in error, or not.</xs:documentation>
                  </xs:annotation>
                </xs:element>
                <xs:element name="ProcessErrorNotifyProducer" type="xs:boolean" minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>This Boolean parameter
```

```
indicates whether (if "true") the Consumer (application/party) of a User Message matching this P-Mode should be notified when an error occurs in the Receiving MSH, during processing of the received User message</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element name="DeliveryFailuresNotifyProducer" type="xs:boolean" minOccurs="0">
                <xs:annotation>
                  <xs:documentation>This Boolean parameter indicates whether (if "true") the Producer (application/party) of a User Message matching this P-Mode must always be notified when the delivery to Consumer failed, or whether (if "false"), in some cases, it is sufficient to notify the Consumer only (Report.ProcessErrorNotifyConsumer="true"). This assumes that Reliability.AtLeastOnce.Contract is "true". This also assumes that the Sending MSH implementation has the ability to determine or to be made aware of all cases of non-delivery that occur after the message has been received by the Receiving MSH.</xs:documentation>
                </xs:annotation>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Security" type="tns:Security" minOccurs="0"/>
 </xs:sequence>
 <xs:attribute name="type" default="fore-channel">
  <xs:simpleType>
   <xs:restriction base="xs:string">
    <xs:pattern value="fore-channel"/>
    <xs:pattern value="back-channel"/>
   </xs:restriction>
  </xs:simpleType>
 </xs:attribute>
</xs:complexType>
<xs:complexType name="Protocol">
 <xs:sequence>
  <xs:element name="Address" type="xs:anyURI" minOccurs="0">
   <xs:annotation>
    <xs:documentation>The value of this parameter represents the address (endpoint URL) of the Receiver MSH (or Receiver Party) to which Messages under this P-Mode leg are to be sent. Note that a URL generally determines the transport protocol (for example, if the endpoint is an email address, then the transport protocol must be SMTP; if the address scheme is "http", then the transport protocol must be HTTP).</xs:documentation>
   </xs:annotation>
  </xs:element>
  <xs:element name="SOAPVersion" minOccurs="0">
   <xs:annotation>
    <xs:documentation>this parameter indicates the SOAP version to be used (1.1 or 1.2). In some implementations, this parameter may be constrained by the implementation, and not set by users.</xs:documentation>
   </xs:annotation>
```

```
      <xs:simpleType>
        <xs:restriction base="xs:float">
          <xs:enumeration value="1.1"/>
          <xs:enumeration value="1.2"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
 </xs:complexType>
 <xs:complexType name="BusinessInfo">
  <xs:annotation>
   <xs:documentation>This set of parameters only applies to user messages.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
   <xs:element name="Service" type="xs:string" minOccurs="0">
    <xs:annotation>
     <xs:documentation>Name of the service to which the User message is intended to
be delivered. Its content should map to the element
eb:Messaging/eb:UserMessage/eb:CollaborationInfo/eb:Service.</xs:documentation>
    </xs:annotation>
   </xs:element>
   <xs:element name="Action" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
     <xs:documentation>Name of the action the User message is intended to invoke. Its
content should map to the element
eb:Messaging/eb:UserMessage/eb:CollaborationInfo/eb:Action</xs:documentation>
    </xs:annotation>
    <xs:complexType>
     <xs:simpleContent>
      <xs:extension base="xs:string">
       <xs:attribute name="after" type="xs:string"/>
       <xs:attribute name="direction" default="out">
        <xs:simpleType>
         <xs:restriction base="xs:token">
          <xs:pattern value="in"/>
          <xs:pattern value="out"/>
          <xs:pattern value="bidirection"/>
         </xs:restriction>
        </xs:simpleType>
       </xs:attribute>
      </xs:extension>
     </xs:simpleContent>
    </xs:complexType>
   </xs:element>
   <xs:element name="Properties" minOccurs="0">
    <xs:annotation>
     <xs:documentation>The value of this parameter is a list of properties. A
property is a data structure that consists of four values: the property name, which can be used as
an identifier of the property (e.g. a required property named "messagetype" can be noted as:
Properties[messagetype].required="true"); the property description; the property data type; and a
Boolean value, indicating whether the property is expected or optional, within the User message.
This parameter controls the contents of the element
```

```
eb:Messaging/eb:UserMessage/eb:MessageProperties.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Property" type="tns:Property" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="PayloadProfiles" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="PayloadProfile" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
              <xs:attribute name="name" type="xs:string">
                <xs:annotation>
                  <xs:documentation>(or Content-ID) that is the part identifier, and can be used
as an index in the notation PayloadProfile[]</xs:documentation>
                </xs:annotation>
              </xs:attribute>
              <xs:attribute name="MIME" type="xs:string">
                <xs:annotation>
                  <xs:documentation>data type (text/xml, application/pdf, etc.);</xs:documentation>
                </xs:annotation>
              </xs:attribute>
              <xs:attribute name="maxSize" type="xs:integer">
                <xs:annotation>
                  <xs:documentation>maximum size in
kilobytes</xs:documentation>
                </xs:annotation>
              </xs:attribute>
              <xs:attribute name="required" type="xs:boolean">
                <xs:annotation>
                  <xs:documentation>Boolean value indicating whether the part is expected or optional,
within the
User message</xs:documentation>
                </xs:annotation>
              </xs:attribute>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
        <xs:attribute name="maxSize">
          <xs:annotation>
            <xs:documentation>This parameter allows for specifying a
maximum size in kilobytes for the entire payload, i.e. for the total of all payload
parts.</xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute name="action"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="MPC" type="xs:anyURI" default="http://docs.oasis-open.org/ebxml-
```

```
msg/ebms/v3.0/ns/core/200704/defaultMPC" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The value of this parameter is the identifier of the MPC (Message
Partition Channel) to which the message is assigned. It maps to the attribute
eb:Messaging/eb:UserMessage/@mpc.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
 </xs:complexType>
 <xs:complexType name="ReceiptType">
  <xs:sequence>
    <xs:element name="ReplyPattern">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="RESPONSE"/>
          <xs:enumeration value="CALLBACK"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="To" type="xs:string" minOccurs="0"/>
  </xs:sequence>
 </xs:complexType>
 <xs:complexType name="Property">
  <xs:sequence>
    <xs:element name="name"/>
    <xs:element name="value"/>
    <xs:element name="action" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
 </xs:complexType>
 <xs:complexType name="Security">
  <xs:sequence>
    <xs:element name="X509" type="tns:X509" minOccurs="0"/>
    <xs:element name="UsernameToken" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The value of this parameter is the username to include in a WSS
Username Token.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:attribute name="username"/>
        <xs:attribute name="password">
          <xs:annotation>
            <xs:documentation>he value of this parameter is the password to
use inside a WSS Username Token.</xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute name="created">
          <xs:annotation>
            <xs:documentation>he Boolean value of this parameter indicates
whether the WSS UsernameToken element should have a Created timestamp
element.</xs:documentation>
          </xs:annotation>
        </xs:attribute>
```

```
        <xs:attribute name="digest">
          <xs:annotation>
            <xs:documentation>The Boolean value of this parameter indicates
whether a password digest should be included in the WSS UsernameToken
element</xs:documentation>
          </xs:annotation>
        </xs:attribute>
      </xs:complexType>
    </xs:element>
    <xs:element name="PModeAuthorize" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The Boolean value of this parameter indicates whether
messages on this MEP leg must be authorized for processing under this P-Mode. If the
parameter is "true" this implies that either
PMode.Responder.Authorization.{username/password}, if the message is sent by Responder, or
PMode.Initiator.Authorization if the message is sent by Initiator, must be used for this purpose,
as specified in Section 7.10. For example, when set to "true" for a PullRequest message sent by
the Initiator, the pulling will only be authorized over the MPC indicated by this Pull signal if (a)
the MPC is the same as specified in the P-Mode leg for the pulled message, and (b) the signal
contains the right credentials (e.g. username/password).</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="SendReceipt" minOccurs="0">
      <xs:complexType>
        <xs:attribute name="sendReceipt" type="xs:boolean" use="required">
          <xs:annotation>
            <xs:documentation>The Boolean value of this parameter indicates whether a
signed receipt (Receipt ebMS signal) containing a digest of the message must be sent
back.</xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute name="replyPattern" use="required">
          <xs:annotation>
            <xs:documentation>This parameter indicates whether the Receipt
signal is to be sent as a callback (value "callback"), or synchronously in the back-channel
response (value "response"). If not present, any pattern may be used</xs:documentation>
          </xs:annotation>
          <xs:simpleType>
            <xs:restriction base="xs:token">
              <xs:enumeration value="callback"/>
              <xs:enumeration value="response"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="WSSVersion" default="1.1">
    <xs:annotation>
      <xs:documentation>value of this parameter represents the version of WS-Security to be
used</xs:documentation>
    </xs:annotation>
```

```
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="1.1"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
  <xs:complexType name="Certificate">
    <xs:attribute name="alias" type="xs:string" use="required">
      <xs:annotation>
        <xs:documentation>Alias of certificate in keystore</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
  <xs:complexType name="X509">
    <xs:sequence>
      <xs:element name="Signature" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Certificate" type="tns:Certificate" minOccurs="0">
              <xs:annotation>
                <xs:documentation>The value of this parameter identifies the public
certificate to use when verifying signed data.</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="Sign" minOccurs="0">
              <xs:annotation>
                <xs:documentation>The value of this parameter is a list of the names of XML
elements (inside the SOAP envelope) that should be signed, as well as whether or not
attachments should also be signed. The list is represented in two sublists that extend this
parameter: Sign.Element[] and Sign.Attachment[]. An element within the Element[] list could
be specified either by its XML name or by its qualified name (its XML name and the namespace
to which it belongs). An element within the Attachment[] list is identified by the Content-
Id.</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:complexContent>
                  <xs:extension base="tns:References">
                    <xs:attribute name="signCertAlias"/>
                  </xs:extension>
                </xs:complexContent>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="hashFunction" type="xs:string"
default="http://www.w3.org/2001/04/xmlenc#sha512">
            <xs:annotation>
              <xs:documentation>The value of this parameter identifies the
algorithm that is used to compute the digest of the message being signed. The definitions for
these values are in the [XMLDSIG] specification.</xs:documentation>
            </xs:annotation>
          </xs:attribute>
```

```
        <xs:attribute name="algorithm" type="xs:string"
default="http://www.w3.org/2001/04/xmldsig-more#rsa-sha512">
          <xs:annotation>
            <xs:documentation>The value of this parameter identifies the
algorithm that is used to compute the value of the digital signature. The definitions for these
values are found in the [XMLDSIG] or [XMLENC] specifications</xs:documentation>
          </xs:annotation>
        </xs:attribute>
      </xs:complexType>
    </xs:element>
    <xs:element name="Encrypt" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The value of this parameter lists the names of
XML elements(inside the SOAP envelope) that should be encrypted, as well as whether or not
attachments should also be encrypted. The list is represented in two sublists that extend this
parameter: Encrypt.Element[] and Encrypt.Attachment[]. An element within these lists is
identified as in Security.X509.Sign lists.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Certificate" type="tns:Certificate" minOccurs="0">
            <xs:annotation>
              <xs:documentation>The value of this parameter identifies the
public certificate to use when encrypting data.</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="Encrypt" type="tns:References" minOccurs="0">
            <xs:annotation>
              <xs:documentation>The value of this parameter is a list of the names of XML
elements (inside the SOAP envelope) that should be signed, as well as whether or not
attachments should also be signed. The list is represented in two sublists that extend this
parameter: Sign.Element[] and Sign.Attachment[]. An element within the Element[] list could
be specified either by its XML name or by its qualified name (its XML name and the namespace
to which it belongs). An element within the Attachment[] list is identified by the Content-
Id.</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
        <xs:attribute name="algorithm" use="required">
          <xs:annotation>
            <xs:documentation>The value of this parameter identifies the
encryption algorithm to be used. The definitions for these values are found in the [XMLENC]
specification.</xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute name="minimumStrength" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="References">
  <xs:sequence>
```

```
    <xs:element name="Elements" minOccurs="0">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="XPath" maxOccurs="unbounded">
        <xs:annotation>
         <xs:documentation>Relative reference from Envelope:
          env:Header/eb3:Messaging
          namespaces:
          env  -> http://www.w3.org/2003/05/soap-envelope
          eb3->  http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/

         </xs:documentation>
        </xs:annotation>
        <xs:complexType>
         <xs:sequence>
          <xs:element name="Namespace" maxOccurs="unbounded">
           <xs:complexType>
            <xs:attribute name="prefix" use="required"/>
            <xs:attribute name="namespace" use="required"/>
           </xs:complexType>
          </xs:element>
         </xs:sequence>
         <xs:attribute name="xpath"/>
        </xs:complexType>
       </xs:element>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
   <xs:attribute name="signElements" type="xs:boolean" default="true"/>
   <xs:attribute name="signAttachments" type="xs:boolean" default="true"/>
  </xs:complexType>
  <xs:complexType name="ReceptionAwareness">
   <xs:sequence>
    <xs:element name="Retry" minOccurs="0">
     <xs:annotation>
      <xs:documentation>(contains a composite string
specifying: (a) maximum number of retries or some timeout, (b) frequency of retries
or some retry rule). The string contains a sequence of parameters of the form:
name=value, separated by either comas or ';'. Example:
"maxretries=10,period=3000", in case the retry period is 3000 ms.</xs:documentation>
     </xs:annotation>
     <xs:complexType>
      <xs:attribute name="maxRetries" type="xs:int" default="3"/>
      <xs:attribute name="multiplyPeriod" type="xs:int" default="3"/>
      <xs:attribute name="period" type="xs:int" default="5678"/>
     </xs:complexType>
    </xs:element>
    <xs:element name="DuplicateDetection" minOccurs="0">
     <xs:annotation>
      <xs:documentation>(contains an im-
plementation specific composite string. As an example this string may specify either
```

(a) maximum size of message log over which duplicate detection is supported, (b) maximum time window over which duplicate detection is supported). The string contains a sequence of parameters of the form: name=value, separated by either comas or ‘;’. Example: “maxsize=10Mb,checkwindow=7D”, in case the duplicate check window is guaranteed of 7 days minimum.</xs:documentation>
</xs:annotation>
<xs:complexType>
<xs:attribute name="windowPeriode" type="xs:duration" default="P1Y"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:schema>