IRU TIR ELECTRONIC PREDECLARATION / NOTIFICATION (IRU TIR EPD) and REAL TIME SAFETIR (RTS) DATA EXCHANGE WITH CUSTOMS

Web Services Interface Specification

Version 2.0

October -2015

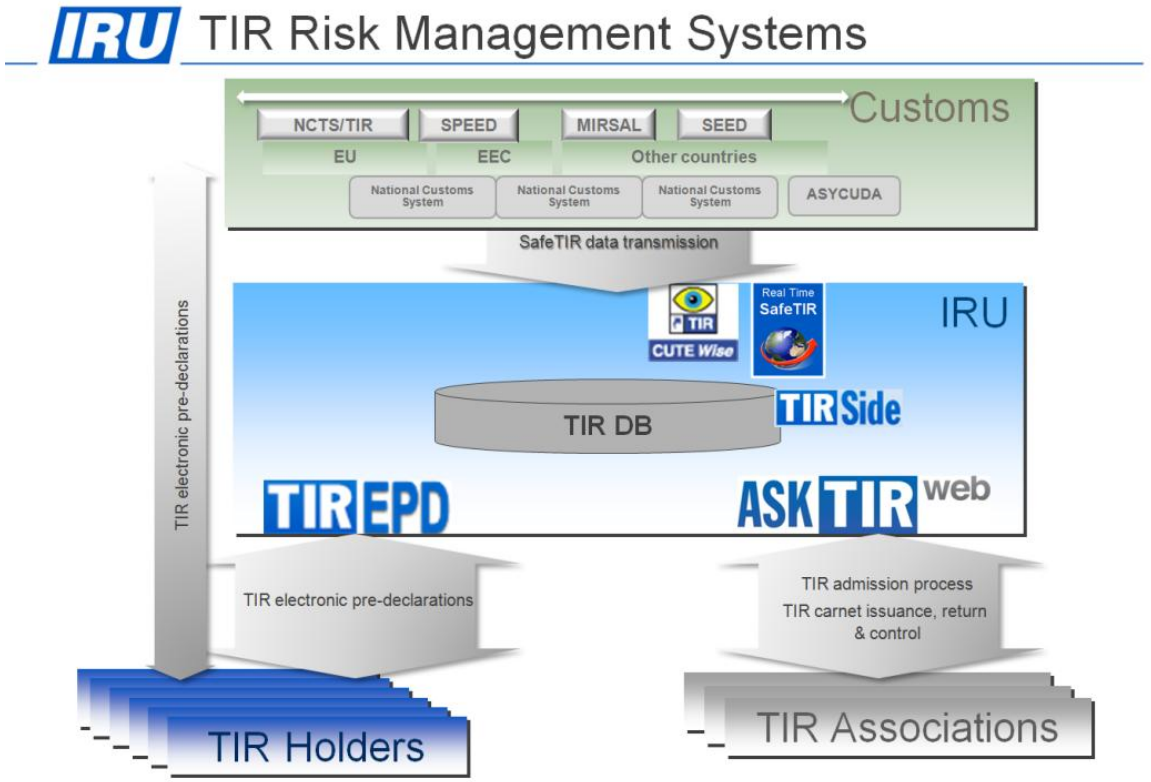
Table of Contents

Version History

Version	Description	Date
1 1.0	2 First document	3 July 2014
4 2.0	5 Reconciliation procedure	6 October 2015

7 Introduction and purpose of the document

This document specifies the web services used for facilitates the integration of the TIR Procedure within Customs Authorities' existing computerized systems, providing real time capabilities for both inland and border posts. RTS and TIR-EPD improves safety and security, facilitates trade, preserves Custom's existing information technology investments, and works within existing custom's TIR procedures.



This document is intended for IT experts of Customs Administrations of TIR Contracting Parties and it defines the interfaces of the following web services: **WSTCHQ**, **WSST**, **WSTIREPDB2G**, **WSTIREPDG2B, EGIS** и **WSRQ**.

8 Abbreviations

B2G	Business to Government
DB	Database
DES	Symmetric algorithm for encryption
ED	Electronic document or message

EPD	Electronic Pre-declaration / Notification
G2B	Government to Business
IRU	International Road Transport Union
IV	Initialization Vector
RSA	Asymmetric algorithm for encryption
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
Holder portal	Application hosted by the IRU for EPD Notifications
TripleDES	Variant of DES
RTS	Real Time SafeTIR
W3C	World Wide Web Consortium
WSDL	Web Service Description Language
WSTIREPDB2G	Web Service provided by the Customs Authorities for Holder messages (sent via the IRU)
WSTIREPDG2B	Web Service provided by the IRU for Customs' messages sent to Holders.
XML	Extensible Markup Language
XSD	XML Schema Definition
WSCt	Web Service Client interface used by the Customs Authorities for TIR Carnet Holder Query
WSDL	Web Service Description Language
WSST	Web Service provided by the IRU for SafeTIR Transmission and exit notifications
WSTCHQ	Web Service provided by the IRU for guarantee information
EGIS	Web Service provided by the IRU for guarantee information (including additional cargo data if available)
WSCs	Web Service Client interface used by the Customs Authorities for SafeTIR Transmission.
WSCe	Web Service Client interface used by the Customs Authorities for query on Guarantee Information
WSCr	Web Service Client interface used by the Customs Authorities to obtain the list of reconciliation requests from the IRU
WSRQ	Web Service provided by the IRU for Reconciliation Request query
WSRE	Method of WSST for Customs to transmit the replies to requests for reconciliation

9 Overview

The mode of interaction between systems is through Web Services using SOAP and XML. All exchanges are synchronous and are between the central facilities of the parties.

For the information exchange, the applications at Customs Authorities will act as "clients" and the applications at IRU will act as "servers" except when the Customs transmit EPD initiated messages.
Capabilities required from the Customs Authority:

- Web Service Client for SafeTIR Transmission and for for Reconciliation Response(WSCs);
- Web Service Client for TIR Carnet Holder Query (WSCt);
- Web Service Client for Guarantee Information (WSCe)
- Web Service for EPD initiated messages;
- Web Service Client to obtain the list of reconciliation requests from the IRU (WSCr);

- Security Engine:
    - Service of generation and exchange of Public key in X509 Certificates;
    - Service of generation of session keys for encryption of documents;
    - Service for creating and verifying hashes using SHA-1
    - Service of encryption/decryption.

Note:

The WS Web Service Client for SafeTIR Transmission (WSCs), Web Service Client for TIR Carnet Holder Query (WSCt) and Web Service Client for Guarantee Information (WSCe) described in this document are in fact the customs applications handling the TIR transit operations. The IRU can provide on demand code samples for services to help customs to implement these Web Services clients in their main applications.

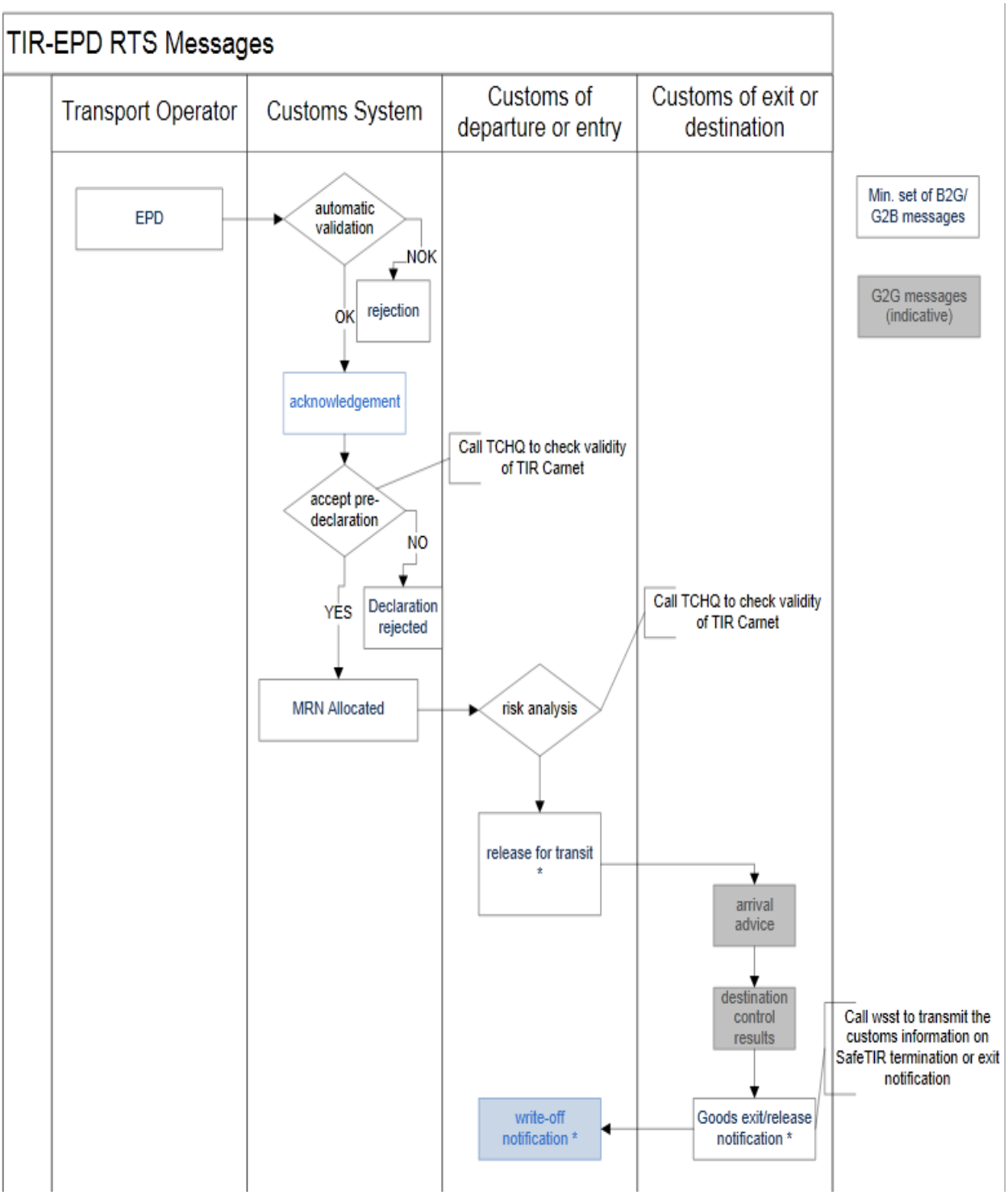
Capabilities required from the IRU:

- Web Service for SafeTIR Transmission (WSST);
- Web Service for TIR Carnet Holder Query (WSTCHQ);
- Web Service for Guarantee Information (EGIS)
- Web Service Client for EPD initiated messages;
- Web Service for for Reconciliation Request query (WSRQ)
- Security Engine:
    - Service of generation and exchange of Public key in X509 Certificates;
    - Service of generation of session keys for encryption of documents;
    - Service for creating and verifying hashes using SHA-1
    - Service of encryption/decryption.

Service for maintenance of subscribers and IRU certificates.

Note: the IT systems within the Customs Authorities required to handle this information exchange are beyond the scope of this document and not described here.

10 Sequence diagram – Messages exchange



The above activity diagram shows the various messages which can be exchanged between business and customs, once a guarantee has been issued to the holder by the issuing association

11 Electronic Pre-declaration / Notification Application Summary

The Customs Authorities are the consumers of the pre-declarations / notifications.

TIR guarantee Holders are the suppliers of the pre-declarations / notifications.

IRU's TIR Issuing associations are responsible for granting the access of TIR guarantee Holders to the TIR-EPD application.

IRU hosts the TIR-EPD application, which allows holders authorized by their association to submit electronic pre-declarations / notifications to the Customs Authorities. The role of the IRU is purely to act as a secure channel for the pre-declaration / notification data exchanged between holders and customs.

11.1 **Holder-Customs Data Exchange Scope**

The exchange allows the holder to submit Declaration messages:

- EPD notification

Customs may respond with:

- Receipt confirmation (optional)
- Receipt automatic technical rejection (optional)
- Customs reference
- Release for transit (departure and entry)
- EPD rejection
- Discharge
- Exit notification
- Termination
- EPD cancellation decision
- No release for transit

11.2 **Customs – IRU**

The exchange allows Customs to transmit SafeTIR termination messages, exit notifications and replies to reconciliation requests.

Customs must keep IRU informed about changes in their reference data and electronic addresses.

11.3 **IRU - Customs**

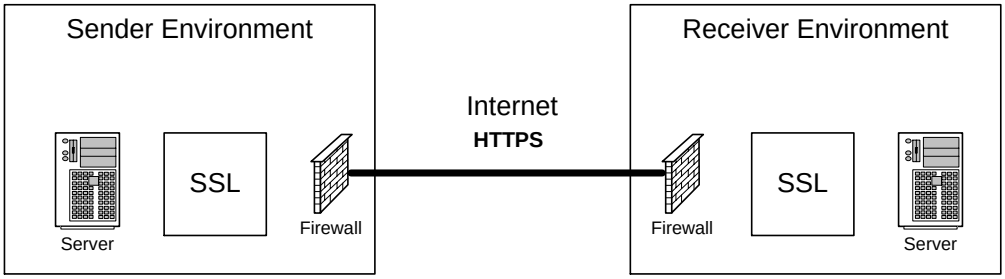
The exchange allows IRU to provide guarantee information, cargo data information and the list of reconciliation requests upon customs query.
IRU must inform Customs about any changes to their reference data and electronic addresses.

12 Security

12.1 **Transmission Security**

Both parties agree to use the Internet for the information exchange. However, the information in the exchanges is sensitive. It must be protected from unauthorized access to ensure the confidentiality and integrity of the information. Authentication of the senders and receivers of the information must also be ensured. However, except for SSL, no third party certification authority will be used.

Basic security will be provided using SSL Version 3.0 and HTTPS protocol.



With the exception of simple acknowledgement messages, additional security will be provided by encrypting the messages.

12.2 **Message Security**

Every XML message exchanged will have the following structure to store security related information.

- The XML will have one <Envelope> element and one <Body> element.
- The <Body> element will contain the business related data

12.2.1 Sender Message Security

The Sender of the XML message will follow the following steps:

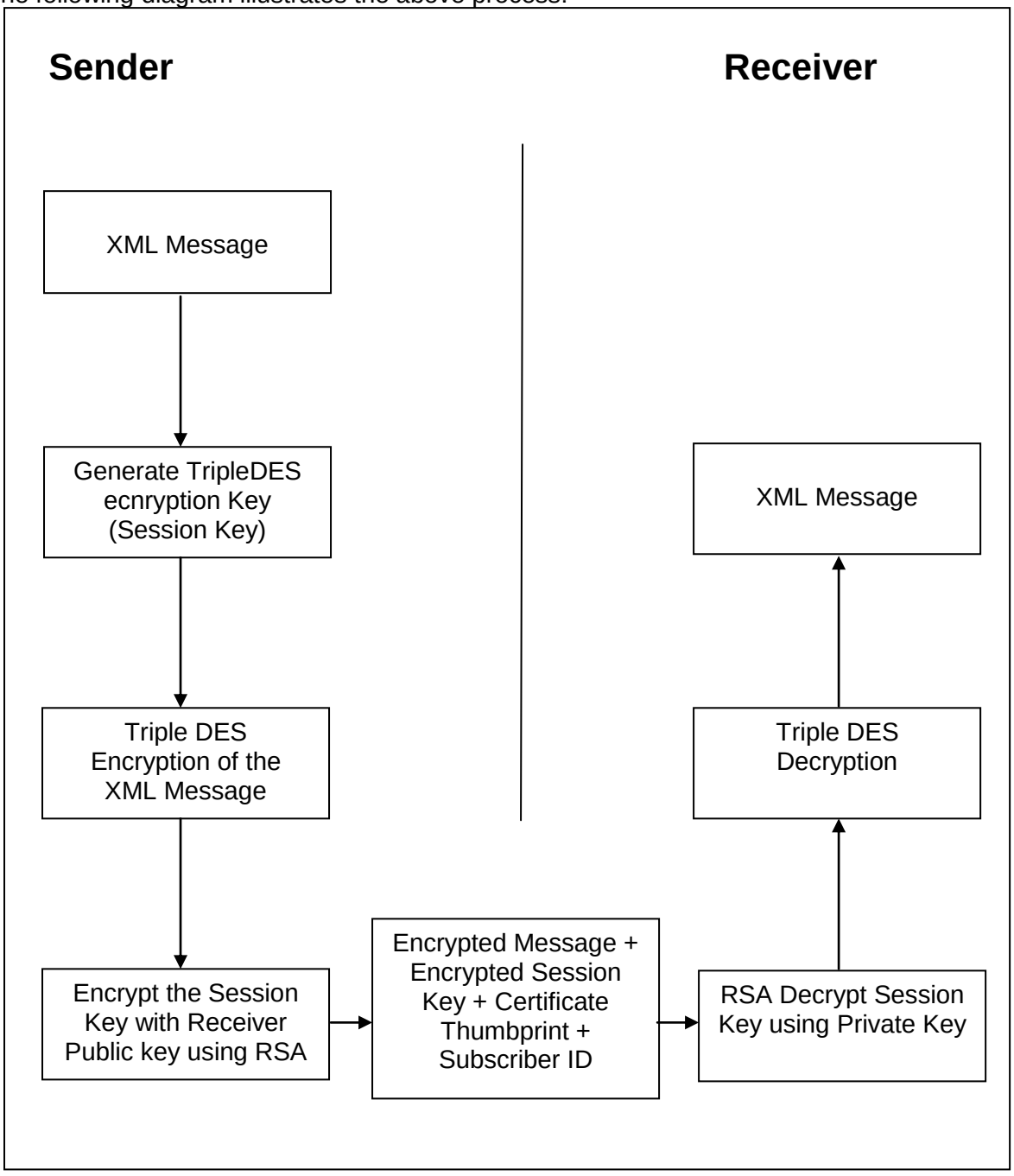
- The XML message contents included in the <Body> </Body> element (excluding the <Body></Body> tags), will be converted to an Unicode string.
- A Session Key will be generated: the XML message will be encrypted using TripleDES algorithm and the Session Key and the mutually agreed IV value.
- The Session Key will be encrypted using the recipients Public Key and RSA asymmetric key encryption algorithm.
- The SubscriberID, encrypted message, the “thumbprint” of the X509 certificate used to encrypt the Session Key and the encrypted Session Key will be exchanged using the Web Service.

12.2.2 Receiver Message Security

The receiver of the message will follow the following steps:

- The “thumbprint” will be used to identify the certificate and the private key.
- The Encrypted Session key will be decrypted using the recipients private Key to get the Session Key
- The Session Key will be used to decrypt the message using TripleDES algorithm to get the XML message

The following diagram illustrates the above process:



Key Exchange:

Customs Authorities and IRU will each generate one or more key pairs. The public keys are exchanged with each other using X.509 certificates. The X.509 certificate “thumbprint” will be used to identify the certificate (hence the keys) used in encryption.

Customs Authorities and IRU will each exchange and maintain a specific telephone number and e-mail address for the communication of needs relative to key exchange. These telephone numbers and e-mail addresses will be staffed continuously during the normal working hours of each party.

Changes to keys will happen under the following circumstances:

Predefined frequency: renewal every 2 years

Ad-hoc: When there is a need to replace or cancel keys (in case of compromise or other reasons).

In both above cases, the initiator of the request will alert/notify the other party by means of both telephone and e-mail.

13 Standards

13.1 **Web-Services and XML**

"Extensible Markup Language (XML) 1.0 (Second Edition)": http://www.w3.org/TR/REC-xml;

"Namespaces in XML": http://www.w3.org/TR/REC-xml-names;

"XML Schema Part 1: Structures" & "XML Schema Part 2: Datatypes", http://www.w3.org/TR/xmlschema-1/ & http://www.w3.org/TR/xmlschema-2/;

“Web Services Description Language (WSDL)” 1.1, W3C Note 15 March 2001: http://www.w3.org/TR/wsdl;
Web Services Security, SOAP Message Security 1.0, OASIS Standard, 01.03.2004.

13.2 **SOAP messages**

"Extensible Markup Language (XML) 1.0 (Second Edition)”: http://www.w3.org/TR/REC-xml;

"Namespaces in XML”: http://www.w3.org/TR/REC-xml-names;

“Simple Object Access Protocol (SOAP) 1.1”: http://www.w3.org/TR/2000/NOTE-SOAP-20000508/, see also http://www.w3.org/TR/SOAP/#soap11.

“Basic Profile Version 1.0a”: http://www.ws-i.org/Profiles/Basic/2003-08/BasicProfile-1.0a.html/

13.2.1 Encryption

TripleDES encryption in CBC mode will be used to encrypt the Session Key.

RSA Cryptography Specifications: http://www.ietf.org/rfc/rfc2437.txt

Encryption will be according to the formula:
***EncryptedMessage = TripleDESEncrypt(Base64Encode(XML))*.**

TransmittedMessage = **RSAEncrypt(SessionKey) + EncryptedMessage + SubscriberID + X.509 Certificate "thumbprint"**

Purpose	Algorithm	Strength / Detail
Message Encryption	TripleDES CBC	IV value = 0x03 0x01 0x04 0x01 0x05 0x09 0x02 0x06 (8 bytes)
Session Key Encryption	RSA	RSA Encryption using PKCS#1 v1.5 padding 1024 bit key

14 Message Version Control (Business rules)

14.1 **Message Formats**

Electronic Messages must adhere to strict rules regarding their format. These formats are governed by XML standards and business rules. This section describes the Business Rules that will be used. XML standards are defined in Section 7, Standards.

Two (2) message format identifiers will be used for messages from Customs Authority to IRU. One, called INFORMATION_EXCHANGE_VERSION is included in the SOAP message as an unencrypted parameter. This parameter will be used by the IRU to select the validations required for any encrypted data sent with the message, specifically, which xsd will be used. The second, called SENDER_DOCUMENT_VERSION, will be included as part of the encrypted data. This element is included optionally to support Customs Authority internal needs. The IRU will not now, nor in the future, use this information.

14.2 **Description of INFORMATION_EXCHANGE_VERSION**

Electronic Messages will identify the INFORMATION_EXCHANGE_VERSION according to the following template:

«n1.n2.n3»

where

n1 – number of version of INFORMATION_EXCHANGE_VERSION,
n2 – number of sub-version of INFORMATION_EXCHANGE_VERSION,
n3 – number of revision (n1,n2,n3 – numbers) of INFORMATION_EXCHANGE_VERSION.

In the case of substantial changes in the XML schema describing electronic documents, the value of the element Version shall be changed by changing the number of the version (sub-version) «n1.n2». Those changes shall be considered substantial that imply interference in the software that processes current ED, for example, change of tag names or structure of existing ED.

In the case of changes to the schema that impose new restrictions on the data to be transmitted, the nomenclature of the ED shall be extended, i.e. no substantial changes shall be introduced in the existing ED, and in the case of changes to the values to be transmitted (for example, return codes), only the number of the revision «n3» shall be changed.

The ED described in the present version of the document shall contain the value of field INFORMATION_EXCHANGE_VERSION equal to “1.0.0” for WSTIREPDB2G; WSTIREPDG2B and WSRQ. The value of field INFORMATION_EXCHANGE_VERSION equal to “2.0.0” for reply via WSST(Version 2).

INFORMATION_EXCHANGE_VERSION numbering will be applied consistently to all documents exchanged via a particular Web Service:.

- The INFORMATION_EXCHANGE_VERSION for both of these electronic documents will be applied at the message (SOAP) level as a parameter. The INFORMATION_EXCHANGE_VERSION of the exchange will be identified in the SOAP message sent by the client.
- The server will reply according to this INFORMATION_EXCHANGE_VERSION or alert the client that the INFORMATION_EXCHANGE_VERSION is not supported.

14.3 **Change of formats**

The change of formats shall be carried out with a simultaneous correction of that document and XML schema with the description of the ED. A change of formats may be carried out in two ways:

- With a simultaneous transition to new formats without the support of old formats.
- The option that implies simultaneous support - for a limited period of time - of both old and new formats may be used on the basis of mutual agreement.
-

14.4 **Description of SENDER_DOCUMENT_VERSION**

SENDER_DOCUMENT_VERSION is provided as an optional element to meet the internal requirements of the Customs Authority. IRU will accept any values (including no value) transmitted within the limits of the agreed xsd.

The ED described in the present version of the document shall contain the value of the field SENDER_DOCUMENT_VERSION equal to “1.0.0”.

15 Participants, Business Processes and Messages

This section describes the participants, business processes, and the list of messages that may be used.

15.1 **Information Exchange Participants**

The list of Interchange Participants is shown in Table 1.

Table 1. List of Interchange Participants

№	Participant	Responsible unit (service)
1	Customs Authority	To be determined
2	International Road Transport Union (IRU)	TIR System IT Department

The participant that provides each component of the system shall be responsible for its operational support and maintenance.

15.2 **Business Processes**

The list of business processes is shown in Table 2.

Table 2. List of business processes.

№	Business process	Characteristics of business process
1	Provide TIR Electronic Pre-declaration / Notification (and related messages) to customs when requested to do so by TIR guarantee Holders. **TIR EPD B2G** **(**TIREPDB2GServiceClass)	TIR guarantee Holders create TIR Electronic Pre-declaration / Notification (and related messages) using the TIR EPD application hosted by the IRU and specify to which customs authority they would like these messages sent.
2	Reply to TIR Electronic Pre-declaration / Notification (and related messages). **TIR EPD G2B** (TIREPDG2BService)	Customs reply to TIR Electronic Predeclaration / Notification (and related messages).
3	Transmission of the exit notification and SafTIR terminations **Realtime SafeTIR** (SafeTirUpload and SafeTIRHolderQueryServiceClass)	Customs will transmit to the IRU the exit notification or SafeTIR termination concerning the guarantee and should control the information on the issue guarantee
4	Query from Customs Authority to IRU regarding status of the TIR carnet (WSTCHQ)	Customs Authority will obtain information regarding TIR Carnets presented at one of their customs offices.
5	Query from Customs Authority to IRU regarding status of the TIR guarantee and Cargo information **(EGIS)**	Customs Authority will obtain information regarding TIR guarantee presented at one of their customs offices and its related cargo information (if available).
6	Query Customs Authorities to IRU under the «Reconciliation» procedure (WSRQ)	Customs Authorities will obtain IRU requests for reconciliation regarding TIR Carnets terminated at one of the customs offices of the its country.

7	Transmission of information from Customs Authorities to IRU under the «Reconciliation» procedure. (WSRE)	Customs Authorities will supply termination data to IRU on TIR Carnets specifically requested for reconciliation by the IRU.

15.3 **Business Operation**

15.3.1 TIR EPD B2G

IRU will act as client. Customs Authority will act as server.

Table 3. List of business operations of the TIREPDB2GServiceClass business process.

№	Business operation	Characteristic of business operation	Conditions and periodicity
1	Submit EPD notification	TIR guarantee holder requests that TIR EPD send a TIR Electronic Pre-declaration / Notification	As required.

15.3.2 TIR EPD G2B

IRU will act as server. Customs Authority will act as client.

Table 4. List of business operations of the TIREPDG2BService business process.

№	Business operation	Characteristic of business operation	Conditions and periodicity
2	Reply to “EPD notification ” (table 3, item 1)	Customs Authority replies.	Customs must reply as mentioned on the document Introduction Message exchange Version 1.4 of the IRU-attached to this document

15.3.3 Realtime SafeTIR

IRU will act as server. Customs Authority will act as client.

Table 5. List of business operations of the SafeTirUpload and SafeTIRHolderQueryServiceClass business process.

№	Business operation	Characteristic of business operation	Conditions and periodicity
3	SafeTIR data transmission ws:**SafeTIRUpload**	Transmission of the Customs records from the Customs Authorities to the IRU	Immediately when the TIR operation is terminated and the related data is entered into Customs IT system (Customs offices of exit/destination).

№	Business operation	Characteristic of business operation	Conditions and periodicity
4	TIR Carnet Holder query ws: SafeTIRHolderQueryServiceClass	Verification of the status of the TIR guarantee by Customs Authorities	Immediately when the TIR guarantee is presented to the Customs Office for Customs Clearance procedure and the TIR guarantee information is entered / becomes available in the Customs IT system.
5	queried data ws:EGISQuery (operation EGISQuery of EGISClassSoap)	Retrieve the status of the TIR guarantee with the latest known Cargo Data by Customs Authorities	Customs can use the service to retrieve the information provided by the Customs office of departure + guarantee status

15.3.4 Reconciliation Query process

IRU will play the role of server. Customs Authorities will be a client of the service.

Table 6 -- List of business operations of the ReconciliationQueryServiceClass business process.

№	Business operation	Characteristic of business operation	Conditions and periodicity
6	ReconciliationRequest ws:**WSRQ**	Download of the Reconciliation Requests, prepared by the IRU.	At least 5 times a day.

15.3.5 Reconciliation Reply process

IRU will play the role of server. Customs Authorities will be a client of the service.

Table 7 -- List of business operations of the ReconciliationQueryServiceClass business process.

№	Business operation	Characteristic of business operation	Conditions and periodicity
7	ReconciliationReply ws:**WSST**	«Reconciliation» procedure: Transmission of the replies to reconciliation requests.	As required to meet the agreed delay for replying to a request. The delay is measured as the difference between the time a reply is received and the time a request has been made available for download.

15.4 **Electronic messages**

Within the framework of business processes, the information interchange between the Parties shall be carried out by means of electronic messages containing the relevant electronic documents. The list of the electronic messages used and their purpose are shown in Table 6.

Table 6. List of electronic messages.

№	Message	Ws:Message name	Purpose
SUBMISSION			
1	EPD notification	ws:TIRPreDeclaration	A message notifying a Customs Authority that a transit under cover of TIR guarantee will be presented.
SUBMISSION DATA VALIDATION			
1.1A	Receipt confirmation (optional)	ws:TIRPreDeclarationReceived	A message from a Customs Authority that a TIRPreDeclaration has been received. Means fully decoded, passed functional validation checks.
1.1B	Receipt automatic technical rejection (optional)	ws:TIRPreDeclarationError	A message from a Customs Authority that a TIRPreDeclaration has caused a technical error.
SUBMISSION TRANSIT VALIDATION			
1.2A	Customs reference	ws:TIRPreDeclarationAccepted	A message from a Customs Authority that a TIRPreDeclaration has been accepted and the transit registered under a movement reference number.
1.2B	EPD rejection	ws:TIRPreDeclarationRejected	A message from a Customs Authority that a TIRPreDeclaration has been rejected.
CANCELLATION			
1.3	Cancellation decision	ws:TIRPreDeclarationCancellationReply	A message from a Customs Authority that a Customs reference has been cancelled.
SUBMISSION TRANSIT DISPOSITION			
1.4A	Release for transit	ws:TIRPreDeclarationReleased	A message from a Customs Authority that a TIRPreDeclaration has been processed and the movement under cover of TIR guarantee has been released.
1.4B	No release for transit	ws:TIRPreDeclarationTransitRefused	A message from a Customs Authority that a TIRPreDeclaration has been processed and the movement under cover of TIR guarantee has been refused for transit

№	Message	Ws:Message name	Purpose
EXIT			
1.5A	Exit notification		A message from a Customs Authority that the operation covered by the guarantee has been terminated at the Customs Office of Exit of the Customs territory
1.5B	ws:**SafeTIRUploadAck**		Result of receipt of the SafeTIRUploadAck
TERMINATION			
1.6A	Termination		Information on TIR guarantees terminations for transmission to IRU
1.6B	ws:**SafeTIRUploadAck**		Result of receipt of the SafeTIRUploadAck
DISCHARGE			
1.7	Discharge	ws: TIRPreDeclarationDischargeNotification	A message from a Customs Authority that the operation covered by the guarantee has been discharged
GUARANTEE INFORMATION			
1.8A	TIRHolderQuery		Query for information on guarantees
1.8B	TIRHolderResponse		Information on guarantees
Cargo Data information			
1.9A	EGISQuery		Query for guarantee information + Cargo data
1.9B	EGISResponse		Provide guarantee information + Cargo data
RECONCILIATION PROCEDURE			
2	RQ Query		Query from Customs Authorities to provide list of reconciliation requests prepared by the IRU
2.1	WSRE Reply		Information on TIR guarantees termination status to IRU for which reconciliation request was made.

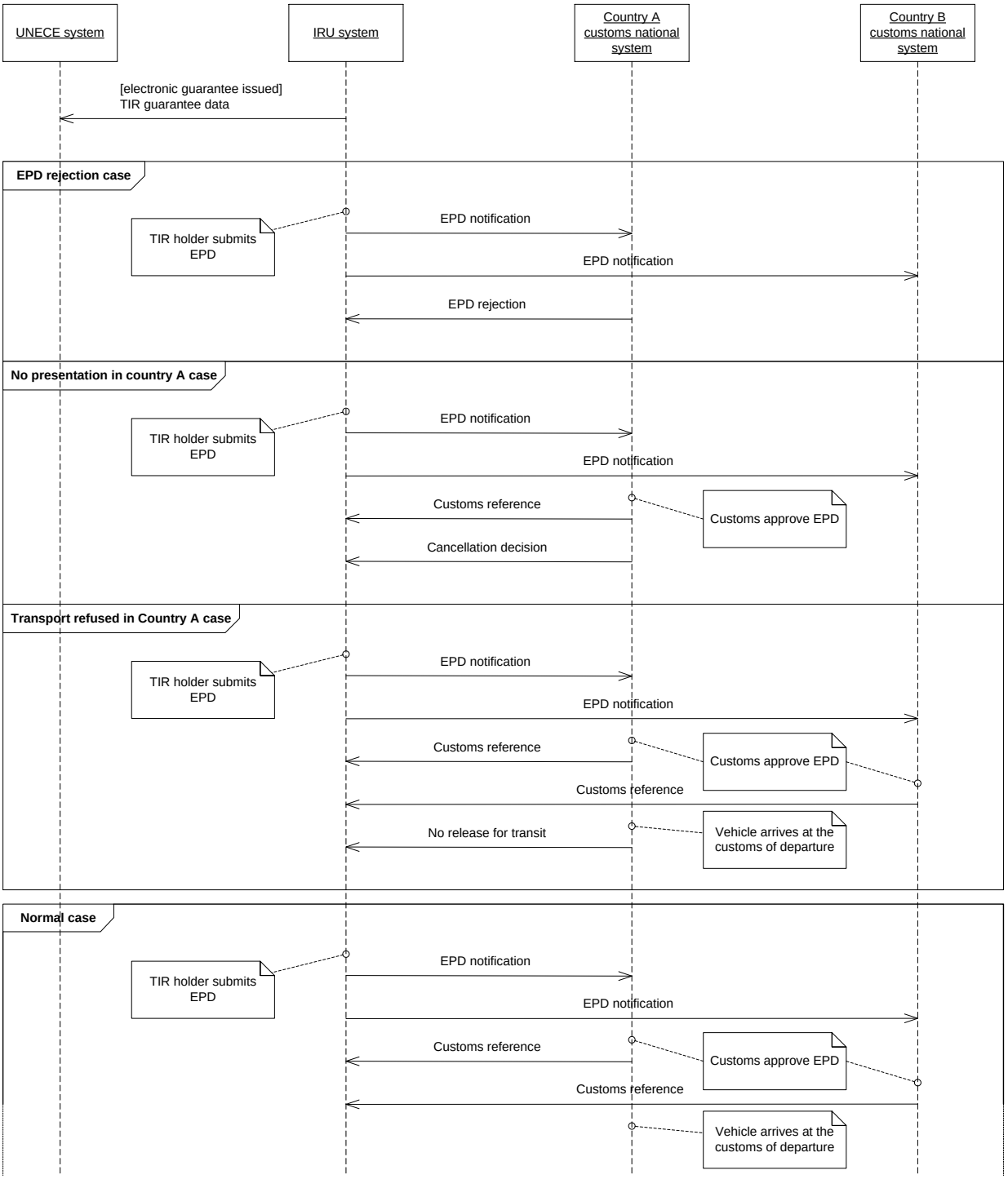
ws: Prefix of namespace. Title of document is unique in this name space

All message exchanges use the following model:

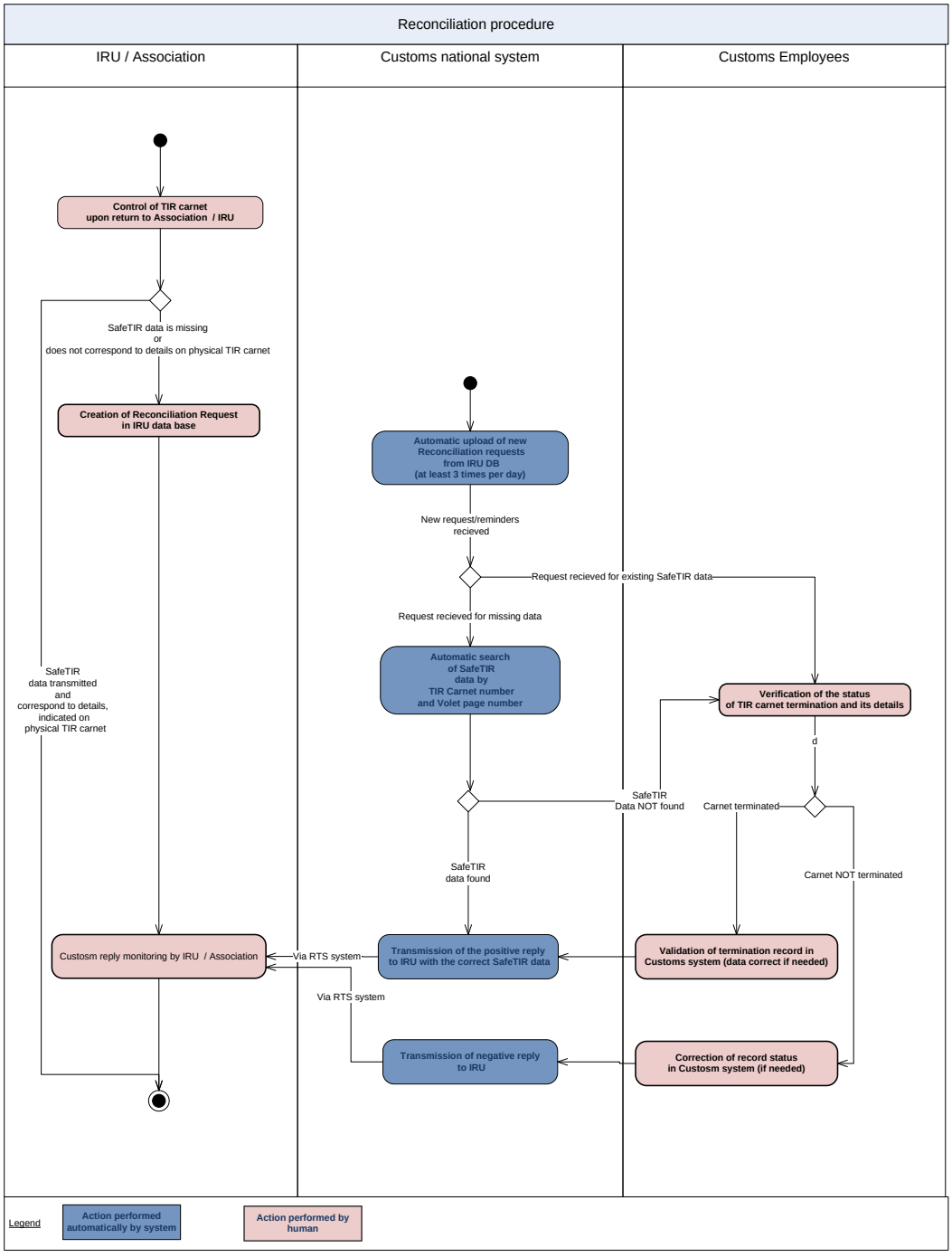
The Web Client initiates and sends the message.
The Web Server acknowledges the message.
Positive acknowledgement, or
Negative acknolwledgement, or
SOAP faultout

The diagrams in the next several sections show the client initiated messages. Each of these client initiated messages has a corresponding acknowledgement message returned by the web service in real time. These service acknowledgements are not included in the diagrams to keep them simpler. Service generated acknowledgements have the same name as the original client message plus “Ack”.

15.4.1 Electronic Pre-declaration / Notification Submission



15.4.2 Proposed schema of reconciliation procedure



15.5 **Message Body Elements**

15.5.1 Client Messages

All client messages contain the following elements:

ELEMENT	TYPE	DESCRIPTION
SubscriberID	String	Identifies the sender of the message.
CertificateID	String	The thumbprint of the certificate used to encrypt the session key.
ESessionKey	base64binary	The encrypted session key.
SubscriberMessageID	String	The ID the subscriber uses to uniquely identify the message.
InformationExchangeVersion	String	The version number of the message.
MessageName	String	The type of message. (9.4 ws:message name)
TimeSent	DateTime	Time stamp when message was sent.
MessageContents	base64binary	The encrypted XML data.

15.5.2 Server Acknowledgement Messages

All server acknowledgement messages contain the following elements:

ELEMENT	TYPE	DESCRIPTION
HostID	String	Identifies the sender of the message.
SubscriberMessageID	String	The ID the subscriber used to uniquely identify the message.
ReturnCode	Int	Identifies success or failure.
ReturnCodeReason	Int	Optional element used to report specific message type error conditions for TIR EPD server (G2B) acknowledgements. ReturnCode always = 1399 when this is used. Value is ‘0’ when not used.

15.5.2.1 Return Codes: All acknowledgements on B2G and G2B WS

All server acknowledgement messages adhere to the following codes.

Value	Meaning
2	Success
1200	Any unclassified error
1210	Missing or invalid SubscriberID
1211	Missing or invalid CertificateID
1213	Missing or invalid ESessionKey
1214	Missing or invalid SubscriberMessageID
1215	Missing or invalid InformationExchangeVersion
1216	Missing or invalid MessageName
1217	Missing or invalid TimeSent
1218	Missing or invalid MessageContents
1301	Encryption/Decryption failure.
1302	Schema validation failure. The MessageContents failed to validate against the xsd.

16 TIR EPD B2G Web Service

The WSTIREPDB2G is hosted by the Customs Authority.

16.1 **WSDL**

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://www.iru.org" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" targetNamespace="http://www.iru.org"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://www.iru.org">
      <s:element name="TIREPDB2G">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="su" type="tns:TIREPDB2GUploadParams" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="TIREPDB2GUploadParams">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="SubscriberID" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="CertificateID" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="ESessionKey" type="s:base64Binary" />
          <s:element minOccurs="0" maxOccurs="1" name="SubscriberMessageID" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="InformationExchangeVersion" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="MessageName" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="TimeSent" type="s:dateTime" />
          <s:element minOccurs="0" maxOccurs="1" name="MessageContent" type="s:base64Binary" />
        </s:sequence>
      </s:complexType>
      <s:element name="TIREPDB2GResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="TIREPDB2GResult" type="tns:TIREPDB2GUploadAck" />
```

```
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:complexType name="TIREPDB2GUploadAck">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="HostID" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="SubscriberMessageID" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="ReturnCode" type="s:int" />
      </s:sequence>
    </s:complexType>
  </s:schema>
</wsdl:types>
<wsdl:message name="TIREPDB2GSoapIn">
  <wsdl:part name="parameters" element="tns:TIREPDB2G" />
</wsdl:message>
<wsdl:message name="TIREPDB2GSoapOut">
  <wsdl:part name="parameters" element="tns:TIREPDB2GResponse" />
</wsdl:message>
<wsdl:portType name="TIREPDB2GServiceClassSoap">
  <wsdl:operation name="TIREPDB2G">
    <wsdl:input message="tns:TIREPDB2GSoapIn" />
    <wsdl:output message="tns:TIREPDB2GSoapOut" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="TIREPDB2GServiceClassSoap" type="tns:TIREPDB2GServiceClassSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="TIREPDB2G">
    <soap:operation soapAction="http://www.iru.org/TIREPDB2G" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="TIREPDB2GServiceClassSoap1" type="tns:TIREPDB2GServiceClassSoap">
```

```
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="TIREPDB2G">
      <soap12:operation soapAction="http://www.iru.org/TIREPDB2G" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="TIREPDB2GServiceClass">
    <wsdl:port name="TIREPDB2GServiceClassSoap" binding="tns:TIREPDB2GServiceClassSoap">
      <soap:address location="http://localhost/RTS2/TIREPDB2G/TIREPDB2GService.asmx" />
    </wsdl:port>
    <wsdl:port name="TIREPDB2GServiceClassSoap1" binding="tns:TIREPDB2GServiceClassSoap1">
      <soap12:address location="http://localhost/RTS2/TIREPDB2G/TIREPDB2GService.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

17 TIR EPD G2B Web Service

The WSTIREPDG2B is hosted by the IRU.

17.1 **WSDL**

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://www.iru.org" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" targetNamespace="http://www.iru.org"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://www.iru.org">
      <s:element name="TIREPDG2B">
        <s:complexType>
```

```
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="su" type="tns:TIREPDG2BUploadParams" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:complexType name="TIREPDG2BUploadParams">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="SubscriberID" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="CertificateID" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="ESessionKey" type="s:base64Binary" />
        <s:element minOccurs="0" maxOccurs="1" name="SubscriberMessageID" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="InformationExchangeVersion" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="MessageName" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="TimeSent" type="s:dateTime" />
        <s:element minOccurs="0" maxOccurs="1" name="MessageContent" type="s:base64Binary" />
      </s:sequence>
    </s:complexType>
    <s:element name="TIREPDG2BResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="TIREPDG2BResult" type="tns:TIREPDG2BUploadAck" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:complexType name="TIREPDG2BUploadAck">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="HostID" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="SubscriberMessageID" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="ReturnCode" type="s:int" />
      </s:sequence>
    </s:complexType>
  </s:schema>
</wsdl:types>
<wsdl:message name="TIREPDG2BSoapIn">
  <wsdl:part name="parameters" element="tns:TIREPDG2B" />
</wsdl:message>
<wsdl:message name="TIREPDG2BSoapOut">
```

```
  <wsdl:part name="parameters" element="tns:TIREPDG2BResponse" />
</wsdl:message>
<wsdl:portType name="TIREPDG2BServiceClassSoap">
  <wsdl:operation name="TIREPDG2B">
    <wsdl:input message="tns:TIREPDG2BSoapIn" />
    <wsdl:output message="tns:TIREPDG2BSoapOut" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="TIREPDG2BServiceClassSoap" type="tns:TIREPDG2BServiceClassSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="TIREPDG2B">
    <soap:operation soapAction="http://www.iru.org/TIREPDG2B" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="TIREPDG2BServiceClassSoap1" type="tns:TIREPDG2BServiceClassSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="TIREPDG2B">
    <soap12:operation soapAction="http://www.iru.org/TIREPDG2B" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="TIREPDG2BService">
  <wsdl:port name="TIREPDG2BServiceClassSoap" binding="tns:TIREPDG2BServiceClassSoap">
    <soap:address location="http://localhost/RTS2/TIREPDG2B/TIREPDG2BService.asmx" />
  </wsdl:port>
  <wsdl:port name="TIREPDG2BServiceClassSoap1" binding="tns:TIREPDG2BServiceClassSoap1">
```

```
      <soap12:address location="http://localhost/RTS2/TIREPDG2B/TIREPDG2BService.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

18 WSST WSDL

The safetiruploadwsdl contains two methods:

- WSST – used for transmission of the SafeTIR termination and exit notification records to the IRU
- WSRE – used to reply to reconciliation requests

Definitive description of the Web Service Interface:

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://www.iru.org" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" targetNamespace="http://www.iru.org"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://www.iru.org">
      <s:element name="WSST">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="su" type="tns:SafeTIRUploadParams" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="SafeTIRUploadParams">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="SubscriberID" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="MessageTag" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="ESessionKey" type="s:base64Binary" />
          <s:element minOccurs="0" maxOccurs="1" name="safeTIRUploadData" type="s:base64Binary" />
          <s:element minOccurs="0" maxOccurs="1" name="CopyToID" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="Sender_MessageID" type="s:string" />
        </s:sequence>
      </s:complexType>
```

```
<s:element name="WSSTResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="WSSTResult" type="tns:SafeTIRUploadAck" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType name="SafeTIRUploadAck">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="Version" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="Sender" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="ResponseTime" type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1" name="ReturnCode" type="s:int" />
    <s:element minOccurs="0" maxOccurs="1" name="Sender_MessageID" type="s:string" />
  </s:sequence>
</s:complexType>
<s:element name="WSRE">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="su" type="tns:SafeTIRReconParams" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType name="SafeTIRReconParams">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="SubscriberID" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="MessageTag" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="ESessionKey" type="s:base64Binary" />
    <s:element minOccurs="0" maxOccurs="1" name="SafeTIRReconData" type="s:base64Binary" />
    <s:element minOccurs="0" maxOccurs="1" name="Sender_MessageID" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="Information_Exchange_Version" type="s:string" />
  </s:sequence>
</s:complexType>
<s:element name="WSREResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="WSREResult" type="tns:SafeTIRUploadAck" />
```

```
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:schema>
</wsdl:types>
<wsdl:message name="WSSTSoapIn">
  <wsdl:part name="parameters" element="tns:WSST" />
</wsdl:message>
<wsdl:message name="WSSTSoapOut">
  <wsdl:part name="parameters" element="tns:WSSTResponse" />
</wsdl:message>
<wsdl:message name="WSRESoapIn">
  <wsdl:part name="parameters" element="tns:WSRE" />
</wsdl:message>
<wsdl:message name="WSRESoapOut">
  <wsdl:part name="parameters" element="tns:WSREResponse" />
</wsdl:message>
<wsdl:portType name="SafeTirUploadSoap">
  <wsdl:operation name="WSST">
    <wsdl:input message="tns:WSSTSoapIn" />
    <wsdl:output message="tns:WSSTSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="WSRE">
    <wsdl:input message="tns:WSRESoapIn" />
    <wsdl:output message="tns:WSRESoapOut" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="SafeTirUploadSoap" type="tns:SafeTirUploadSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="WSST">
    <soap:operation soapAction="http://www.iru.org/WSST" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
```

```
    </wsdl:operation>
    <wsdl:operation name="WSRE">
      <soap:operation soapAction="http://www.iru.org/WSRE" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="SafeTirUploadSoap12" type="tns:SafeTirUploadSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="WSST">
      <soap12:operation soapAction="http://www.iru.org/WSST" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="WSRE">
      <soap12:operation soapAction="http://www.iru.org/WSRE" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="SafeTirUpload">
    <wsdl:port name="SafeTirUploadSoap" binding="tns:SafeTirUploadSoap">
      <soap:address location="http://localhost/RTS2/WSST_WS/safetirupload.asmx" />
    </wsdl:port>
    <wsdl:port name="SafeTirUploadSoap12" binding="tns:SafeTirUploadSoap12">
```

```
      <soap12:address location="http://localhost/RTS2/WSST_WS/safetirupload.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

18.1 **SafeTIR Upload Data Definitions**

The SafeTIR Upload has the following elements:

MessageTag
XML data type: String
Allow NULLs: Not allowed
Notes: “thumbprint” of the certificate used to encrypt the data.
Cardinality: [1]

SubscriberID
XML data type: String
Allow NULLs: Not allowed
Notes: Identifies the Sender. Maximum length 255 characters.
Cardinality: [1]

ESessionKey
XML data type: base64Binary
Allow NULLs: Not allowed
Notes: The encrypted session key.
Cardinality: [1]

SafeTIRUploadData
XML data type: base64Binary
Allow NULLs: Not allowed
Notes: The encrypted (SafeTIR data + hash).
Cardinality: [1]

Sender_MessageID
XML data type: String
Allow NULLs: Not allowed
Notes: Required element allowing the IRU to report message processing failures back to the sender. Maximum length 255 characters.
Cardinality: [1]

18.2 **SafeTIR Upload Ack Data Definitions**

SafeTIR Upload Acknowledgement has the following elements

Sender
XML data type: String
Allow NULLs: Not allowed
Notes: "IRU" Maximum length 255 characters.
Cardinality: [1]

ResponseTime
XML data type: datetime
Allow NULLs: Not allowed
Notes: UTC time the acknowledgment was sent
Cardinality: [1]

ReturnCode
XML data type: int
Allow NULLs: Not allowed

Cardinality: [1]
Values:

Value	Applied To	Message
2		Success
Values representing error:		
1200		Any unclassified error
1210	Message Tag	No Message Tag Value
1212	SubscriberID	Subscriber ID Missing or Bad formatted or not registered
1213	ESessionKey	Session key missing
1214	SafeTIRUploadData, TIRCarnetHolderQueryParams	Encrypted data missing
1222	Sender_MessageID	Value is more than maximum length specified or Missing

Sender_MessageID
XML data type: String
Allow NULLs: Not allowed
Notes: Required element. The ID assigned by the authority that originally sent the message. Maximum length 255 characters.
Cardinality: [1]

18.3 **SafeTIRUploadData Element**

The SafeTIRUploadData element is hashed and encrypted. Once it has been decrypted it should conform to the definitions below.

19 WSTCHQ WSDL

Description of the Web Service Interface:

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://www.iru.org" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" targetNamespace="http://www.iru.org"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://www.iru.org">
      <s:element name="WSTCHQ">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="su" type="tns:TIRHolderQuery" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="TIRHolderQuery">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="SubscriberID" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="MessageTag" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="ESessionKey" type="s:base64Binary" />
          <s:element minOccurs="0" maxOccurs="1" name="TIRCarnetHolderQueryParams" type="s:base64Binary" />
          <s:element minOccurs="0" maxOccurs="1" name="Query_ID" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:element name="WSTCHQResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="WSTCHQResult" type="tns:TIRHolderResponse" />
          </s:sequence>
        </s:complexType>
      </s:element>
```

```
    <s:complexType name="TIRHolderResponse">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="MessageTag" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="ESessionKey" type="s:base64Binary" />
        <s:element minOccurs="0" maxOccurs="1" name="TIRCarnetHolderResponseParams" type="s:base64Binary" />
        <s:element minOccurs="1" maxOccurs="1" name="ReturnCode" type="s:int" />
        <s:element minOccurs="0" maxOccurs="1" name="Query_ID" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:schema>
</wsdl:types>
<wsdl:message name="WSTCHQSoapIn">
  <wsdl:part name="parameters" element="tns:WSTCHQ" />
</wsdl:message>
<wsdl:message name="WSTCHQSoapOut">
  <wsdl:part name="parameters" element="tns:WSTCHQResponse" />
</wsdl:message>
<wsdl:portType name="SafeTIRHolderQueryServiceClassSoap">
  <wsdl:operation name="WSTCHQ">
    <wsdl:input message="tns:WSTCHQSoapIn" />
    <wsdl:output message="tns:WSTCHQSoapOut" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="SafeTIRHolderQueryServiceClassSoap" type="tns:SafeTIRHolderQueryServiceClassSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="WSTCHQ">
    <soap:operation soapAction="http://www.iru.org/WSTCHQ" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="SafeTIRHolderQueryServiceClassSoap12" type="tns:SafeTIRHolderQueryServiceClassSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
```

```
    <wsdl:operation name="WSTCHQ">
      <soap12:operation soapAction="http://www.iru.org/WSTCHQ" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="SafeTIRHolderQueryServiceClass">
    <wsdl:port name="SafeTIRHolderQueryServiceClassSoap" binding="tns:SafeTIRHolderQueryServiceClassSoap">
      <soap:address location="http://localhost/RTS2/TCHQ_WS/tirCarnetQuery.asmx" />
    </wsdl:port>
    <wsdl:port name="SafeTIRHolderQueryServiceClassSoap12" binding="tns:SafeTIRHolderQueryServiceClassSoap12">
      <soap12:address location="http://localhost/RTS2/TCHQ_WS/tirCarnetQuery.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

19.1 **TIR Holder Query Data Definitions**

SubscriberID
XML data type: string
Allow NULLs: Not allowed
Notes: Identifies the Sender. Maximum length 255 characters.
Cardinality: [1]

MessageTag
XML data type: String
Allow NULLs: Not allowed
Notes: “thumbprint” of the certificate used to encrypt the data.
Cardinality: [1]

ESessionKey
XML data type: base64Binary
Allow NULLs: Not allowed

Notes: The encrypted session key.
Cardinality: [1]

TIRCarnetHolderQueryParams
XML data type: base64Binary
Allow NULLs: Not allowed
Notes: The hashed and encrypted query data
Cardinality: [1]

Query_ID
XML data type: String
Allow NULLs: Allowed
Notes: Optional value the sender may use to track this message. Maximum length 20 characters.
Cardinality: [0..1]

20 Cargo data WSDL

20.1 **Description of the Web Service Interface:**

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://rts.iru.org/EGIS"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" targetNamespace="http://rts.iru.org/EGIS"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://rts.iru.org/EGIS">
      <s:element name="EGISQuery">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="su" type="tns:EGISQueryType" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="EGISQueryType">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="SubscriberID" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="MessageTag" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="ESessionKey" type="s:base64Binary" />
          <s:element minOccurs="0" maxOccurs="1" name="EGISQueryParams" type="s:base64Binary"
/>
          <s:element minOccurs="0" maxOccurs="1" name="Query_ID" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:element name="EGISResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="EGISResult"
type="tns:EGISResponseType" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="EGISResponseType">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="MessageTag" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="ESessionKey" type="s:base64Binary" />
          <s:element minOccurs="0" maxOccurs="1" name="EGISResponseParams"
type="s:base64Binary" />
          <s:element minOccurs="1" maxOccurs="1" name="ReturnCode" type="s:int" />
          <s:element minOccurs="0" maxOccurs="1" name="Query_ID" type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="EGISQuerySoapIn">
    <wsdl:part name="parameters" element="tns:EGISQuery" />
```

```
  </wsdl:message>
  <wsdl:message name="EGISQuerySoapOut">
    <wsdl:part name="parameters" element="tns:EGISResponse" />
  </wsdl:message>
  <wsdl:portType name="EGISClassSoap">
    <wsdl:operation name="EGISQuery">
      <wsdl:input message="tns:EGISQuerySoapIn" />
      <wsdl:output message="tns:EGISQuerySoapOut" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="EGISClassSoap" type="tns:EGISClassSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="EGISQuery">
      <soap:operation soapAction="http://rts.iru.org/EGIS/EGISQuery" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="EGISClassSoap12" type="tns:EGISClassSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="EGISQuery">
      <soap12:operation soapAction="http://rts.iru.org/EGIS/EGISQuery" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="EGISClass">
    <wsdl:port name="EGISClassSoap" binding="tns:EGISClassSoap">
      <soap:address
location="http://localhost/RTS2/EGIS_WS/electronicGuaranteeInformationService.asmx" />
    </wsdl:port>
    <wsdl:port name="EGISClassSoap12" binding="tns:EGISClassSoap12">
      <soap12:address
location="http://localhost/RTS2/EGIS_WS/electronicGuaranteeInformationService.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

20.2 **EGIS query Data Definitions**

SubscriberID

XML data type: string
Allow NULLs: Not allowed
Notes: Identifies the Sender. Maximum length 255 characters.
Cardinality: [1]

MessageTag

XML data type: String
Allow NULLs: Not allowed
Notes: “thumbprint” of the certificate used to encrypt the data.
Cardinality: [1]

ESessionKey
XML data type: base64Binary
Allow NULLs: Not allowed
Notes: The encrypted session key.
Cardinality: [1]

EGISQueryParams
XML data type: base64Binary
Allow NULLs: Not allowed
Notes: The hashed and encrypted query data
Cardinality: [1]

Query_ID
XML data type: String
Allow NULLs: Allowed
Notes: Optional value the sender may use to track this message. Maximum length 20 characters.
Cardinality: [0..1]

21 WSRQ Details[1]

21.1 WSRQ WSDL: description of the Web Service Interface

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://www.iru.org"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
targetNamespace="http://www.iru.org" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://www.iru.org">
      <s:element name="WSRQ">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="su"
type="tns:ReconciliationQuery" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ReconciliationQuery">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="MessageTag" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="SubscriberID" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="ESessionKey" type="s:base64Binary"
/>
          <s:element minOccurs="0" maxOccurs="1" name="ReconciliationQueryData"
type="s:base64Binary" />
```

```
          <s:element minOccurs="0" maxOccurs="1" name="Sender_MessageID" type="s:string"
/>
          <s:element minOccurs="0" maxOccurs="1" name="Information_Exchange_Version"
type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:element name="WSRQResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="WSRQResult"
type="tns:ReconciliationResponse" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ReconciliationResponse">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="Sender" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="MessageTag" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="ESessionKey" type="s:base64Binary"
/>
          <s:element minOccurs="0" maxOccurs="1" name="ReconciliationRequestData"
type="s:base64Binary" />
          <s:element minOccurs="0" maxOccurs="1" name="Sender_MessageID" type="s:string"
/>
          <s:element minOccurs="1" maxOccurs="1" name="ReturnCode" type="s:int" />
        </s:sequence>
      </s:complexType>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="WSRQSoapIn">
    <wsdl:part name="parameters" element="tns:WSRQ" />
  </wsdl:message>
  <wsdl:message name="WSRQSoapOut">
    <wsdl:part name="parameters" element="tns:WSRQResponse" />
  </wsdl:message>
  <wsdl:portType name="ReconciliationQueryServiceClassSoap">
    <wsdl:operation name="WSRQ">
      <wsdl:input message="tns:WSRQSoapIn" />
      <wsdl:output message="tns:WSRQSoapOut" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="ReconciliationQueryServiceClassSoap"
type="tns:ReconciliationQueryServiceClassSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="WSRQ">
      <soap:operation soapAction="http://www.iru.org/WSRQ" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="ReconciliationQueryServiceClassSoap12"
type="tns:ReconciliationQueryServiceClassSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="WSRQ">
      <soap12:operation soapAction="http://www.iru.org/WSRQ" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
```

```
  <wsdl:service name="ReconciliationQueryServiceClass">
    <wsdl:port name="ReconciliationQueryServiceClassSoap"
binding="tns:ReconciliationQueryServiceClassSoap">
      <soap:address location="http://localhost/rts2/wsrq_ws/wsrq.asmx" />
    </wsdl:port>
    <wsdl:port name="ReconciliationQueryServiceClassSoap12"
binding="tns:ReconciliationQueryServiceClassSoap12">
      <soap12:address location="http://localhost/rts2/wsrq_ws/wsrq.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

21.2 **Reconciliation Query Data Definitions**

These are the parameters passed by the Customs Authorities to the IRU as part of the SOAP message.

The Reconciliation Query has the following elements:

MessageTag
XML data type: String
Allow NULLs: Not allowed
Notes: “thumbprint” of the certificate used to encrypt the data.
Cardinality: [1]

SubscriberID
XML data type: String
Allow NULLs: Not allowed
Notes: Identifies the Sender. Maximum length 255 characters.
Cardinality: [1]

ESessionKey
XML data type: base64Binary
Allow NULLs: Not allowed
Notes: The encrypted session key.
Cardinality: [1]

ReconciliationQueryData
XML data type: base64Binary
Allow NULLs: Not allowed
Notes: The encrypted (Query data + hash).
Cardinality: [1]

Sender_MessageID
XML data type: String
Allow NULLs: Not allowed
Notes: Required element allowing the IRU to report message processing failures back to the sender. Maximum length 255 characters.
Cardinality: [1]

Information_Exchange_Version
XML data type: String
Allow NULLs: Not allowed
Notes: Required element representing the version of the information exchange. Currently “1.0.0”
Cardinality: [1]

21.2.1 Reconciliation Request Data Definitions

Reconciliation Request has the following elements

Sender
XML data type: String
Allow NULLs: Not allowed
Notes: "IRU" Maximum length 255 characters.
Cardinality: [1]

MessageTag
XML data type: String
Allow NULLs: Not allowed
Notes: “thumbprint” of the certificate used to encrypt the data.
Cardinality: [1]

ESessionKey
XML data type: base64Binary
Allow NULLs: Not allowed
Notes: The encrypted session key.
Cardinality: [1]

ReconciliationRequestData
XML data type: base64Binary
Allow NULLs: Not allowed
Notes: The encrypted (Request data + hash).
Cardinality: [1]

Sender_MessageID
XML data type: String
Allow NULLs: Not allowed
Notes: Required element allowing the IRU to report message processing failures back to the sender. Maximum length 255 characters.
Cardinality: [1]

ReturnCode
XML data type: int
Allow NULLs: Not allowed
Cardinality: [1]
Values:

Value	Applied To	Message
2		Success
Values representing error:		
1200		Any unclassified error
1210	Message Tag	No Message Tag Value
1212	SubscriberID	Subscriber ID Missing or Bad formatted or not registered
1213	ESessionKey	Session key missing
1214	ReconciliationQueryData	Encrypted data missing
1222	Sender_MessageID	Value is more than maximum length specified or Missing
1223	Information_Exchange_Version	Version is not valid

1250	Database query timeout	Service unavailable. Please try later.

21.3 **WSST (Version 2)**

21.3.1 WSST WSDL

Please see section 12 WSST WSDL

21.3.2 ReconciliationRequestReply Data Definitions

Reconciliation Reply has the following elements

SubscriberID
XML data type: String
Allow NULLs: Not allowed
Notes: Identifies the Sender. Maximum length 255 characters.
Cardinality: [1]

MessageTag
XML data type: String
Allow NULLs: Not allowed
Notes: “thumbprint” of the certificate used to encrypt the data.
Cardinality: [1]

ESessionKey
XML data type: base64Binary
Allow NULLs: Not allowed
Notes: The encrypted session key.
Cardinality: [1]

SafeTIRReconData
XML data type: base64Binary
Allow NULLs: Not allowed
Notes: The encrypted (Request data + hash).
Cardinality: [1]

Sender_MessageID
XML data type: String
Allow NULLs: Not allowed
Notes: Required element allowing the IRU to report message processing failures back to the sender. Maximum length 255 characters.
Cardinality: [1]

Information_Exchange_Version
XML data type: String
Allow NULLs: Not allowed
Notes: Required element representing the version of the business document. Planned “2.0.0”
Cardinality: [1]

21.3.3 ReconciliationRequestReplyAck

SafeTIR Upload Acknowledgement has the following elements

Sender
XML data type: String
Allow NULLs: Not allowed
Notes: "IRU" Maximum length 255 characters.
Cardinality: [1]

ResponseTime
XML data type: datetime
Allow NULLs: Not allowed
Notes: UTC time the acknowledgment was sent
Cardinality: [1]

ReturnCode
XML data type: int
Allow NULLs: Not allowed
Cardinality: [1]
Values:

Value	Applied To	Message
2		Success
Values representing error:		
1200		Any unclassified error
1210	Message Tag	No Message Tag Value
1212	SubscriberID	Subscriber ID Missing or Bad formatted or not registered
1213	ESessionKey	Session key missing
1214	ReconciliationRequestReplyData	Encrypted data missing
1222	Sender_MessageID	Value is more than maximum length specified or Missing
1223	Information_Exchange_Version	Version is not valid

Sender_MessageID
XML data type: String
Allow NULLs: Not allowed
Notes: Required element. The ID assigned by the authority that originally sent the message. Maximum length 255 characters.
Cardinality: [1]

22 Reference Tables (Code Lists)

TIR EPD uses international standard codes where available.

22.1 **Country Codes**

ISO 3116-1 alpha-2 maintained by the International Standards Organization.
http://www.iso.org/

22.2 **Language Codes**

Language codes: ISO 639-1 maintained by the International Standards Organization.
http://www.iso.org/

22.3 **Commodity Codes**

Harmonized Commodity Description and Coding System (HS 6 digit) maintained by the World Customs Organization.
http://www.wcoomd.org/

23 Attachements list

Following attachments are available upon requests:

- EPD_XML-Mappings-EN.xlsx - document describes the content of the TIR-EPD messages and the list of codes used on the messages
- EPD_Schemas-*.zip - XSD schemas of the TIR-EPD RTS messages
- Introduction to the Electronic Messages Exchange

*****

< === End of Document === >