



European Union

This project is funded by the European Union



**Technical Assistance to the Customs Administration of Serbia
to Support the Modernisation of the Customs Systems**

Functional specification ECC GW- for traders

<ECC GW specificationv1.3_eng>

PROJECT REF: EuropeAid/131367/C/SER/RS

SAGA
new frontier group

KOMIX
SYSTEMS INTEGRATOR

Aquasoft



European Union

Technical Assistance to the Customs Administration of Serbia to Support the Modernisation of the Customs Systems



Document Title : Functional specification ECC GW-For traders
 Document ref. : ECC GW v1.3
 Status : For Publishing
 Date : 07-07-2014
 Author : Consortium
 Distribution list :

Approvals

Approved by: Ljiljana Đokić Date: 21-07-2014.
 Project Manager
 Consortium

Approved by: Olga Protić Date: __-__-2014.
 Project manager
 CAS

REVISION HISTORY

| Ed. | Rev. | Date | Description | Action* | Pages |
|-----|------|------------|----------------|---------|---|
| 1 | 2 | 09-01-2014 | For publishing | I/R | All |
| 1 | 3 | 21-07-2014 | For publishing | R | Chapters: 3.1, 6.1.2, 6.1.3, 6.1.4, 6.2, 6.2.3, 6.2.5, 6.4 |
| | | | | | |

(*) Action: I=Insert R=Replace



Contents

| | |
|--|-----------|
| 1. INTRODUCTION..... | 3 |
| DESCRIPTION OF ELECTRONIC CUSTOMS CLEARANCE GATEWAY (ECC GW) | 3 |
| 2. ECC GW PROCESSES | 5 |
| 2.1 MESSAGE SENDING..... | 5 |
| 2.2 MESSAGE POLLING..... | 6 |
| 2.3 MESSAGE DELIVERING..... | 7 |
| 2.4 MESSAGE DELIVERY CONFIRMATION..... | 8 |
| 3. MESSAGE STRUCTURE..... | 9 |
| 3.1 ECC ENVELOPE..... | 9 |
| 3.2 ECC RESPONSE..... | 13 |
| 3.3 ACKNOWLEDGEMENT..... | 13 |
| 3.4 LIST OF MESSAGE'S GUIDS..... | 14 |
| 4. ECC GW INTERFACE..... | 15 |
| 4.1 METHODS..... | 15 |
| 4.1.1 Send..... | 15 |
| 4.1.2 Poll..... | 16 |
| 4.1.3 Deliver..... | 17 |
| 4.1.4 Confirm..... | 17 |
| 5. SECURITY..... | 19 |
| 5.1 SECURITY REQUIREMENTS | 19 |
| 5.2 SECURITY PRINCIPLES | 19 |
| 5.3 DIGITAL SIGNATURES | 20 |
| 5.4 SECURE ELECTRONIC SIGNATURE..... | 20 |
| 5.5 REQUIRED PROPERTIES FOR INPUT DOCUMENTS..... | 21 |
| 5.6 REQUIRED PROPERTIES FOR OUTGOING MESSAGES..... | 22 |
| 6. APPENDIX..... | 23 |
| 6.1 ENVELOPE..... | 23 |
| 6.1.1 ECC Envelope Xml schema..... | 23 |
| 6.1.2 Business message for Delivery and Confirm..... | 25 |
| 6.1.3 Example of XML Envelope before signing..... | 25 |
| 6.1.4 Example of XML Envelope after signing..... | 26 |
| 6.1.5 Example of XML Envelope for Delivery..... | 29 |
| 6.1.6 Example of XML Envelope for Confirm..... | 29 |
| 6.2 RESPONSE MESSAGE..... | 30 |
| 6.2.1 ECC response Xml schema..... | 30 |
| 6.2.2 Example of ACK Message..... | 31 |
| 6.2.3 Example of NAK Message..... | 32 |
| 6.2.4 Example of MessageIdentifiers..... | 32 |
| 6.2.5 Example of signed outgoing message..... | 32 |
| 6.3 WSDL..... | 34 |
| 6.4 LIST OF ERRORS CODES..... | 39 |

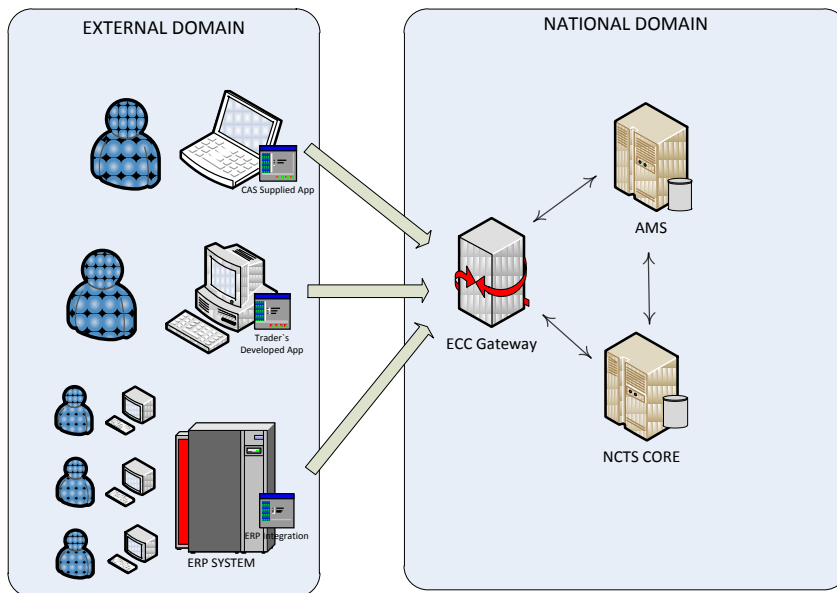


1. INTRODUCTION

Description of Electronic Customs Clearance Gateway (ECC GW)

The ECC Gateway is a system that provides the communication between the Serbian Customs Administration (CAS), and the external subjects, mainly economic operators (traders, declarants). It allows external users to conduct their business with CAS in real time.

From the security reasons, National and External domain will be separated and communication between them will be possible exclusively by usage of Electronic Customs Clearance Gateway (ECC GW).



ECC Gateway will interface economic operators systems on one side and internal NCTS system on the other side. Internet will be used for communications in the External Domain.

Basic function of ECC Gateway is to exchange relevant documents between customs offices and traders (operators). Data will be transferred in the form of messages (message based communication) between IS of CAS presented by Electronic Customs Clearance Gateway (ECC GW) and IS of economic operator.



European Union

Technical Assistance to the Customs Administration of Serbia to Support the Modernisation of the Customs Systems



The ECC envelope is the main communication message on the ECC Gateway. It is used for encapsulation of information about the message in transport. The information is needed for correct message routing to the server application, for security verification, and for the tracking of the message in the system.

Data will be transferred in the XML format. They will be encapsulated in XML message – communication envelope. This envelope will describe all communication parameters necessary for processing at the ECC GW side. All messages should be electronically signed. The communication channel should be secured by HTTPs transport.

There will be implemented appropriate communication protocol based on standard web services interface (SOAP, WSDL) allowing sending of messages to CAS and regular polling for presence of new messages directed to economic operator.

Electronic signatures will be processed by ECC Gateway in both directions. ECC GW will validate electronic signatures of all incoming messages which will be timestamped and archived after that.

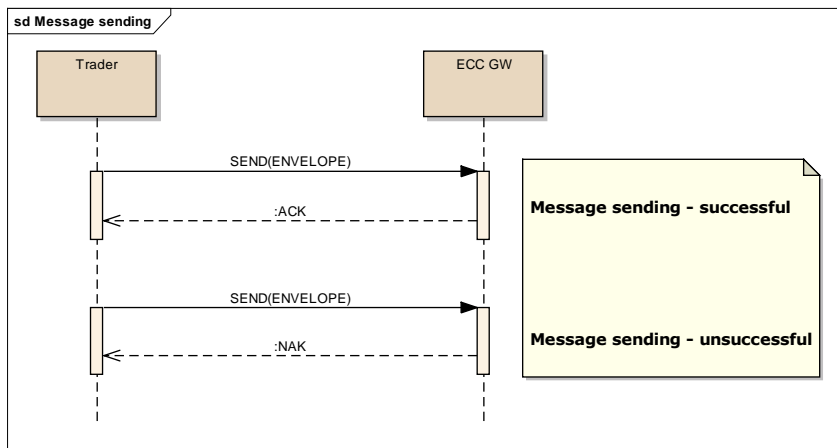
On the other side, ECC Gateway will sign all outgoing messages using system certificate under conditions given by Serbian national legislation.

All applications that will be developed according to this specification should pass through the process of registration and approval (technical certification) by the CAS authority.

2. ECC GW processes

2.1 Message sending

Process of message sending involves several steps. Process begins when the user sends envelope. Received message (ECC Envelope) is validated against its xml schema after which the message signature is pre-verified (verification of digital signature, certificate validity, certificate chain and checking of CRLs).The processes continues with external user authorization and finishes with message queuing, when the message is placed into the input queue of target application for later business processing.

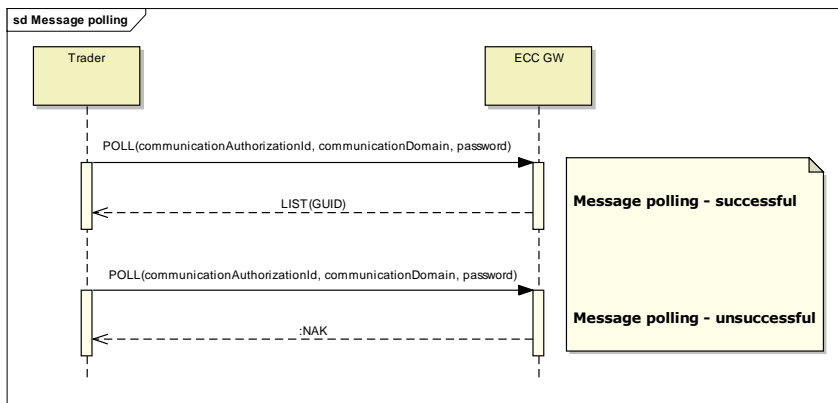


The process can be realized through two scenarios:

1. Successful scenario - User sends envelope through method *Send*. ECC GW returns acknowledge message
2. Unsuccessful scenario - User sends message envelope through method *Send*. ECC GW returns negative acknowledge message.

2.2 Message polling

Through the process of message polling, user receives a list of available messages from the system. User asks for the list of belonging messages for specified CommunicationAuthorizationID, communicationDomain and password. ECC GW returns the list of messages for a specific user. The list of messages may be empty. Unlike other messages which are signed, authorization for Message polling is performed using password for particular CommunicationAuthorizationID and CommunicationDomain.



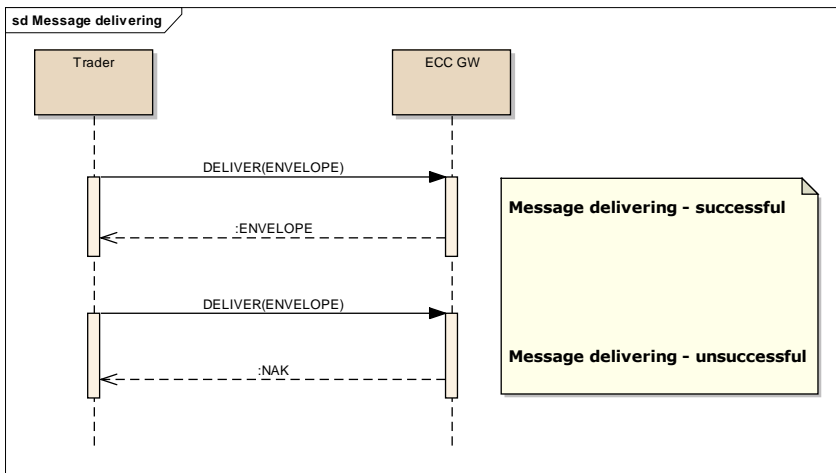
The process can be realized through two scenarios:

1. Successful scenario - User sends communicationAuthorizationID, communicationDomain and password through the method *Poll*. ECC GW returns the list of all unique message identifiers (guids) dedicated to him.
2. Unsuccessful scenario - User sends communicationAuthorizationID, communicationDomain and password through the method *Poll*. ECC GW returns negative acknowledge message.

2.3 Message delivering

In the previous process (Message polling), user gets a list of all unique message identifiers (guids) dedicated to him.

Based on this list, the user can receive the whole message, by sending an envelope with appropriate guid. After message validation, ECC GW will return the business message inserted in the appropriate envelope.



Message validation includes:

- Message (ECC Envelope) validation against the appropriate message specification (xml schema)
- Verification of digital signature
- Verification of certificate validity
- User authorization

Each envelope will be signed by ECC GW server signature to allow user to determine the authenticity of the message.

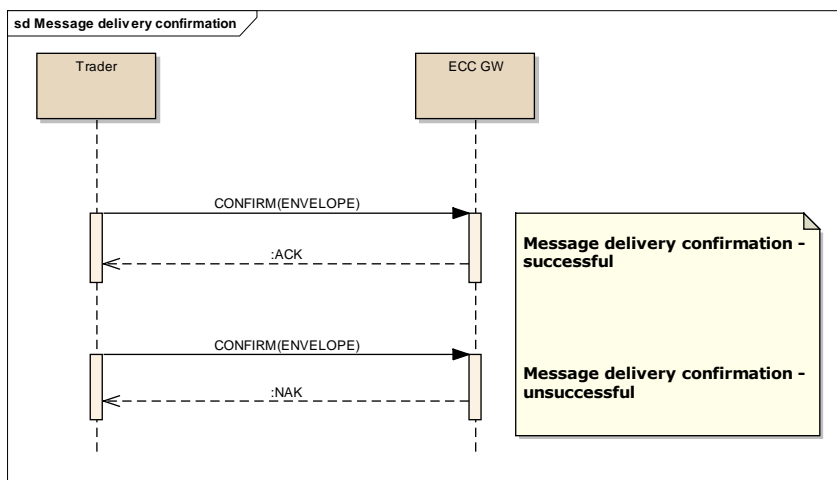
The process can be realized through two scenarios:

1. Successful scenario - User sends an envelope with appropriate guid through method *Deliver*. ECC GW returns envelope.
2. Unsuccessful scenario - User sends an envelope with appropriate guid through method *Deliver*. ECC GW returns negative acknowledgement.

2.4 Message delivery confirmation

After message delivering, in this process, user sends confirmation of message delivery. On the ECC GW side deleting of that message from the system is performed.

User sends envelope with appropriate *guid*, to confirm the message delivery after which the message is deleted from the system.



The process can be realized through two scenarios:

1. Successful scenario - User confirms the message delivery by sending the envelope with appropriate *guid* through method *Confirm*. ECC GW returns acknowledge message.
2. Unsuccessful scenario - User confirms the message delivery by sending the envelope with appropriate *guid* through method *Confirm*. ECC GW returns negative acknowledge message.



3. Message structure

3.1 ECC Envelope

All data to be transported by the system have to be encapsulated into the envelope. Following table describes the elements of the ECC envelope.

| Element name | XML type | Data type | Obligation towards ECC Gateway | Obligation from ECC Gateway | Description |
|---------------|----------|--------------------|--------------------------------|-----------------------------|--|
| ECC | Element | | Mandatory | Mandatory | Carries information about the envelope |
| Header | Element | | Mandatory[1] | Mandatory[1] | Carries information about the envelope. properties |
| OperationType | Element | string enumeration | Mandatory | Optional | Type of operation done by communicating party. |
| UniqueID | Element | GUID | Mandatory | Mandatory | Unique identifier determines just one instance of ECC envelope |
| Version | Element | string[10] | Mandatory | Mandatory | Denotes ECC envelope version |
| Domain | Element | string[20] | Mandatory | Mandatory | The name of the communication domain in which the data exchange has to take place |
| Message | Element | | Mandatory[1] | Mandatory[1] | Carries information about the message |
| MessageType | Element | string[30] | Mandatory | Mandatory | The name of the transported message, typically the root element of the enveloped XML |
| Participants | Element | | Mandatory[1] | Mandatory[1] | List of participants of the data exchange (e.g. Trader, Customs,...) |



| | | | | | | | |
|--|--|------------------------------|-----------|-------------|---------------|---------------|---|
| | | Participant | Element | | Mandatory | Mandatory | The participant of the data exchange |
| | | CommunicationAuthorizationID | Element | string[40] | Mandatory | Mandatory | The unique identifier of the communicating party |
| | | OrganizationID | Element | string[15] | Optional | Optional | The unique identifier of the represented party (trader) - VAT |
| | | ScenarioID | Element | GUID | Mandatory | Mandatory | The ID of the scenario within the procedure used by the communicating party |
| | | ReferenceNumber | Element | String[40] | Optional | Optional | Reference number of business message. |
| | | AppID | Element | string[50] | Optional | Optional | The ID of the Application generating respective business message |
| | | AppVersion | Element | string[20] | Optional | Optional | Version of the Application |
| | | ExtendedInfo | Element | | Optional[1] | Optional[1] | The space where to store further complementary information |
| | | Attribute | Element | | Optional[1-n] | Optional[1-n] | Complementary Attribute |
| | | Name | Attribute | string[20] | Mandatory | Mandatory | The name of the complementary attribute |
| | | Value | Attribute | string[256] | Mandatory | Mandatory | The value of the complementary attribute |
| | | Data | Element | | Mandatory[1] | Mandatory[1] | Contains XML data of business message |
| | | Signature | Element | | Mandatory | Mandatory | Placement of information about the electronic signature according to the Recommendation W3C - XML signature. This is the signature of the whole |

Comment [MOD1]: ScenarioID is mandatory field 1 for ADM messages

Comment [MOD2]: Added new element – ReferenceNumber



| | | | | | |
|--|--|--|--|--|-------------------------|
| | | | | | communication envelope. |
|--|--|--|--|--|-------------------------|

Authorization of signed message at the ECC GW is based on verifying, that for declared CommunicationAuthorizationID and Domain in the ECC Envelope the certificate used for a signature is registered in the AMS system.

In case of message sending (SEND method), message delivering (DELIVER method) or message delivery confirmation (CONFIRM method) input parameter is ECC envelope. In case of message delivering response (DELIVER method), response message contains ECC Envelope.

Structure of ECC Envelope is given in table above and populating of critical (mandatory) ECC Envelope elements will be explained in more details.

Each instance of the envelope must contain a unique GUID¹ identifier in the element "UniqueID". In case of more messages with the same "UniqueID" sent to ECC GW, only first message will be accepted. In case of message delivery response, ECC GW generates "UniqueID". Element "Version" indicates ECC Envelope version used by trader. Current version is "1.0". Element "Domain" is the name of the communication domain in which the data exchange has to take place. Possible values are i.e. "NCTS", "GMS" and may be extended later. "MessageType" element is the identification of the transported business message, typically the name of the root element of the enveloped XML. In case of message delivering or message delivery confirmation, element "MessageType" must have value of respective communication message (i.e. "ADM001").

Group "Participants" holds data about communication parties. In the first message of new business scenario it will contain original sender (trader) only. The first CAS response sending business message will add second party – CAS itself. In all subsequent communications related to given business scenario both parties must be contained.

Element "CommunicationAuthorizationID" is unique identifier of the communicating party – it is the reference number of respective Communication Authorization for given customs regime issued by CAS and registered inside of its AMS system (Authorization Management System), respectively "CAS".

"ScenarioID" is GUID generated by each party (trader and later CAS) in time of sending of the first business message in the new business scenario (i.e. submit of a new transit declaration). It serves as unique technical correlation identifier for trader and also for CAS at the communication level grouping together all business messages exchanged for given business scenario (i.e. one MRN at Customs Office of Departure). Each communicating party will assign its own "ScenarioID". They must remain invariable during the whole lifecycle of the communication. Both parties - ECC Gateway and trader SW must ensure that it will be returned unchanged in the response within the same business scenario.

ScenarioID for service messages (ADMxxx – DELIVER, CONFIRM) can be the new ScenarioID because from the list of GUIDs for delivery from previous POLL cannot be deduced corresponding message and scenario.

Comment [MOD3]: Posto je scenarioID postao obavezan, opis kako se koristi u ADM porukama

"ReferenceNumber" is reference number of business message from "Data" part of the envelope. Its value depends on type of message:

- For NCTS messages (ie ND015, ND013...) – LRN or MRN
- For GMS messages (ie ND026) – GRN

Comment [MOD4]: Description of new element Ref.Number

¹ For non-Microsoft implementations the UUID may be used instead of GUID, see <http://en.wikipedia.org/wiki/Uuid>



European Union

Technical Assistance to the Customs Administration of Serbia to Support the Modernisation of the Customs Systems



- For query messages (ie ND111A, ND034A...) and voucher's maintenance (ie ND224) is not necessarily to fill reference number

„AppID“and „AppVersion“elements will every time contain identifier and version of application emitting business message (and corresponding ECC envelope). They are very usefull for CAS HelpDesk experts for identification of known errors and generation of error statistics for every application.

Element „ExtendedInfo“ is currently not used, and is reserved for possible future extensions of the ECC envelope.

Element „Data“ contains exactly one business message for the specified communication domain. In case of either message delivering or message delivery confirmation, structure of business message is specified in chapter 6.1.2.



3.2 ECC response

ECC response is the ECC GW response to the trader request. ECCResponse consists of two elements as it is shown in the following table.

| Element name | XML type | Data type | Obligation towards ECC Gateway | Obligation from ECC Gateway | Description |
|--------------|----------|--------------------|--------------------------------|-----------------------------|---|
| ResponseType | Element | String enumeration | Illegal | Mandatory | Type of response |
| ResponseData | Element | anyType | Illegal | Mandatory | Contains xml data of appropriate ResponseType |

In case of Message sending (chapter 2.1.) the type of response is ACKNOWLEDGEMENT (ACK or NAK).

In case of Message polling (chapter 2.2.), type of response can be either MESSAGEIDENTIFIERS or ACKNOWLEDGEMENT (NAK).

In case of Message delivering (chapter 2.3.), type of response can be either ECC Envelope or ACKNOWLEDGEMENT (NAK).

Finally, in case of Message delivery confirmation (chapter 2.4.), type of response is ACKNOWLEDGEMENT (ACK or NAK).

3.3 Acknowledgement

| Element name | XML type | Data type | Obligation towards ECC Gateway | Obligation from ECC Gateway | Description |
|-----------------|----------|-----------|--------------------------------|-----------------------------|--|
| Acknowledgement | Element | | Illegal | Mandatory | |
| Result | Element | string | Illegal | Mandatory | NAK or ACK |
| Reference | Element | string | Illegal | Optional | Contains GUID of the original envelope to which the error refers if it is possible to display it |
| DateTime | Element | string | | Mandatory | |
| ErrCode | Element | int | Illegal | Optional | Numeric error code |



| | | | | | |
|----------------|---------|-------------|---------|----------|--|
| ErrType | Element | string[20] | Illegal | Optional | Denotes the error type or error source |
| ErrDescription | Element | string[256] | Illegal | Optional | Error description |
| ErrData | Element | string | Illegal | Optional | Error data (if necessary) |

3.4 List of message's guids

| Element name | XML type | Data type | Obligation towards ECC Gateway | Obligation from ECC Gateway | Description |
|--------------------|----------|-----------|--------------------------------|-----------------------------|----------------------------------|
| MessageIdentifiers | Element | | Illegal | Mandatory | Carries list of guids |
| MessageIdentifier | Element | string | Illegal | Optional | Contains GUID of polled messages |



4. ECC GW INTERFACE

4.1 Methods

In the appendix, in section 6.3, wsdl file is shown. WSDL will also be available on url where ECC GW web service will be deployed.

Following figure presents four methods of ECC GW.

| | Name | Description |
|--|---------|--|
| | Confirm | Confirms that the message has been delivered from the system core to the calling application |
| | Deliver | Delivers the message from the system core to the calling application |
| | Poll | Polls for a list of message identifiers |
| | Send | Sends the message to system core |

4.1.1 Send

Calling this method executes the Message sending process described in part 2.1 of this document

Syntax

```
public string Send(
    string envelope
)
```

Parameters

envelope

Type: String

String representation of a well formed XML containing an envelope² with business message³

Return value

Type: String

String representation of a well-formed XML containing response with acknowledgement message⁴

² Envelope XML schema is given in part 6.1.1. Example is given in part 6.1.4

³ Business message should be inserted in Data element of the Envelope. Business message structures will be specified in external documentation of respective communication domain



4.1.2 Poll

Calling this method executes the Message polling process described in part 2.2 of this document

Syntax

```
public string Poll(  
    string communicationAuthorizationId,  
    string communicationDomain,  
    string password  
)
```

Parameters

communicationAuthorizationId

Type: String
User's communication authorization identifier

communicationDomain

Type: String
User's communication domain

password

Type: String
User's password for polling of messages for given communicationAuthorization and Domain

Return value

Type: String
String representation of a well-formed XML containing response with message identifiers message⁵ (in case of no error) or acknowledgement message⁶ (in case of error with result NAK)

⁴ Element **Acknowledgement** in Response XML schema given in part 6.2.1. Examples are given in parts 6.2.2 and 6.2.3

⁵ Element **MessageIdentifiers** in Response XML schema given in part 6.2.1. Example is given in part 6.2.4

⁶ Element **Acknowledgement** in Response XML schema given in part 6.2.1. Example is given in part 6.2.3



4.1.3 Deliver

Calling this method executes the Message delivering process described in part 2.3 of this document

Syntax

```
public string Deliver(  
    string envelope  
)
```

Parameters

envelope

Type: String

String representation of a well-formed XML containing an envelope⁷ with message identifier⁸

Return Value

Type: String

String representation of a well-formed XML containing an response with envelope⁹ with business message¹⁰ (in case of no error) or acknowledgement message⁶ (in case of error with result NAK)

4.1.4 Confirm

Calling this method executes the Message delivery confirmation process described in part 2.4 of this document

Syntax

```
public string Confirm(  
    string envelope  
)
```

⁷ Envelope XML schema is given in part 6.1.1. Example is given in part 6.1.4

⁸ XML schema is given in part 6.1.2. Examples are given in part 6.1.5 and 6.1.6

⁹ Element ECC in Response XML schema given in part 6.2.1. Example is given in part 6.2.5

¹⁰ Business message should be inserted in Data element of the ECC element. Business message structures will be specified in external documentation of respective communication domain



European Union

Technical Assistance to the Customs Administration of Serbia to Support the Modernisation of the Customs Systems



Parameters

envelope

Type: String

String representation of a well-formed XML containing an envelope¹¹ with message identifier¹²

Return Value

Type: String

String representation of a well-formed XML containing an response with acknowledgement message¹³

¹¹ Envelope XML schema is given in part 6.1.1. Example is given in part 6.1.4

¹² XML schema is given in part 6.1.2. Examples are given in part 6.1.5 and 6.1.6

¹³ Element **Acknowledgement** in Response XML schema given in part 6.2.1. Examples are given in parts 6.2.2 and 6.2.3



European Union



5. SECURITY

5.1 Security Requirements

The systems in government domain belong to the "mission critical" system group. That is why, they should satisfy very high criteria and at the same time offer the maximum comfort in usage. Electronic Customs projects are one of them, and therefore, basic premises are given by law regulative, sub law acts and normatives that determine the electronic documents - message systems and business processes that have to be done on these documents. The crucial processes in CAS system are connected to electronic interchange of customs relevant documents.

According to Serbian laws about Electronic Data Communications, some key criteria are defined:

1. Electronic documents must be exchanged by agreed communication channels as formalized datasets.
2. Electronic documents are grouped according to the criterion of belonging to the same set required to realize business process or part of a business process.
3. Electronic documents must be unaltered at the receiver.
4. Electronic documents must be processed in real time (real time is determined solely from the business processes and thus the basic unit of measure is not common - response time in the system).
5. Electronic documents must be such to don't allow systematic sniffing of communication channels and such that it's not possible systematic data acquisition.
6. Electronic document on the receiver's address must be undeniable (non reputable).
7. Each sent message - an electronic document must have confirmation receipt. The receipt is also message - electronic document that meets all the criteria of a message but does not have a confirmation receipt. NAK and ACK messages are not electronic documents, but system level message structures. They will not be electronically signed.
8. In the case of expertise by the relevant institutions, with the help of a distributed set of messages and responses, there must be possible reconstruction of the sequence of events and the business processes which are expertised.

5.2 Security principles

ECC GW solution should be fully secure in the means of computer system security. Here is a review of standard security requirements and description of how it is covered by proposed system.

1. Identification

The system must know identity of a user who makes some actions in CAS environment.

2. Authentication

Certified identity of a user is the key stone of security. It must be proven, that user really is the person who he/she claims to be. Users are authenticated by certificates issued to them and registered by CAS except for message polling which is protected by password.

3. User rights



European Union



Authorization of trustworthy authenticated user ensures that user can communicate with the system. Authorization is done as part of trader registration by CAS, whether it is done by certificate or by password.

4. Integrity

Integrity means that the data sent from one site is the same as the data received at other site. Integrity is fully covered by usage of electronic signature.

5. Confidentiality

This means that nobody can read confidential data. Confidentiality is realized through encryption / decryption of sensitive data at the communication channel level (HTTPS).

6. Security audit

ECC GW is generating internal audit logs.

5.3 Digital signatures

Because of high requirements, Advanced Digital Signature will be used for user authentication and authorization, for securing data integrity and for non repudation of given transactions.

Digital certificates will be used to electronically sign business messages in both directions (from ECC GW and from trader's accredited person). At trader side, every person must have valid qualified electronic certificate issued from Authorized Qualifying Certification Authority. Certificate must be on smart card or USB PKI token (stated by law regulative). At ECC GW side, messages will be signed by ECC GW server certificate.

5.4 Secure Electronic Signature

It is fully governed by the recommendation W3C XML Signature (<http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/>).

Several facts need to be taken into account:

- 1) The XML canonization has to be performed and put into the signature structure

Canonicalization method

<http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments>

- 2) The signature is always applied on the element SignatureInfo which contains the minimized (SHA-256 hash) document,
- 3) The special characters like space, tab, CR/LF are significant for the signature, the canonization does not remove them and the changes in them cause error in verification of the signature. Re-format of XML may occur for example between the signature and the verification of the document.

In the ECC Gateway the hash algorithm is performed on the ECC Envelope level encapsulating business data also, not on the business data itself. Also the public part of the certificate must be included in the signature.



All security considerations below correspond with the following laws and regulations:

❖ The Law on Electronic Signature

With associated Bylaws:

- The Regulation on the evidence of certification bodies,
- The Rule on certification bodies issuing qualified electronic certificates in the Republic of Serbia,
- The Regulation on technical and technological procedures for creating qualified electronic signature and the criteria to be fulfilled by means of creating qualified electronic signature
- The Regulation on conditions for issuing qualified electronic certificates.

- ❖ The Law on Electronic Commerce
- ❖ The Law on Electronic Documents
- ❖ The Rule book on the Timestamp Mark
- ❖ The Regulation of Electronic Office Work
- ❖ The Law on Archiving (draft version).

The following are the major parties involved in a business transaction supported by electronic signatures as defined in the document:

- The Signer;
- The Verifier;
- Trusted Service Providers (TSP).

The following TSPs are used to support the functions defined in the present document:

- Certification Authorities;
- Time-Stamping Authorities.

5.5 Required properties for input documents

From the view of recommendation W3C, ECC Gateway supports XMLDSIG standard in the form of "enveloped signature". This means that the signature is created in the context of the whole ECC envelope which was already compiled in the time of signature. The element "Signature" is placed as a child of the ECC Envelope.

Messages from traders to CAS don't require timestamp but will be always timestamped at CAS side at the time of receiving.

Messages from CAS to traders will be timestamped at the time of processing them in ECC GW. This may not be the time of accepting them by traders (because of communication condition with trader).

An XMLDSIG signature has the following basic structure:

```

<ds:Signature ID?>
  <ds:SignedInfo/>
  <ds:SignatureValue/>
  <ds:KeyInfo/>           (must contain at least signing certificate)
  <ds:Object/>           (optional, multiple values allowed)
</ds:Signature>

```

ECC GW requires XAdES – BES with at least signing certificate as signed signature property as format of electronic signatures in input documents:

XAdES – BES:

Signature contains Object element with *QualifyingProperties* and some *SignedProperties* elements (e.g. *SigningTime*).

```
<Signature xmlns="....." Id="SignatureId_yyy">
  <SignedInfo>
    ...
    <Reference URI="#idSignedProperties_zzz"
      Type="http://uri.etsi.org/01903#SignedProperties">
      ...
      <Object>
        <QualifyingProperties Target="#SignatureId_yyy"
          xmlns:xd="http://uri.etsi.org/01903/v1.3.2#">
          <SignedProperties Id="idSignedProperties_zzz">
            <SignedSignatureProperty>
              <SigningTime>2/4/2009 2:44:01 PM</SigningTime>
            </SignedSignatureProperty>
          </SignedProperties>
        </QualifyingProperties>
      </Object>
    </SignedInfo>
  </Signature>
```

5.6 Required properties for outgoing messages

Required properties for outgoing messages must contain the following properties:

Mandatory:

Signed Signature Properties:

- The signature certificate used to create the signature
- The signature policy identifier

Unsigned Signature Properties:

- Signature Time Stamp

Optional:

Signed Signature Properties:

- Signature Production Place
- Signer Role

Signed Data Properties

- Data Object Format
- Commitment Type Indication

Example of signed outgoing message is shown in appendix, section 6.2.4.



6. APPENDIX

This part of document presents xml schemas and xml document examples. At the end, wsdl file is shown.

6.1 Envelope

6.1.1 ECC Envelope Xml schema

ECCEnvelope.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema" >
  <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-
schema.xsd"/>
  <xs:element name="ECC">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="OperationType" minOccurs="0">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:enumeration value="SEND"/>
                    <xs:enumeration value="DELIVER"/>
                    <xs:enumeration value="CONFIRM"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>

              <xs:element name="UniqueID" minOccurs="1" maxOccurs="1" >
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:pattern value="[a-f0-9]{8}(?:-[a-f0-9]{4}){3}-[a-f0-9]{12}"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="Version" minOccurs="1" maxOccurs="1" >
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:pattern value="[0-9]+[.]?[0-9]*/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="Domain" minOccurs="1" maxOccurs="1" >
                <xs:simpleType>
                  <xs:restriction base="xs:string">
```




```

<xs:pattern value="\w+"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="Message" minOccurs="1" maxOccurs="1" >
  <xs:complexType>
    <xs:sequence>
      <xs:element name="MessageType" type="xs:string" minOccurs="1"
maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Participants" minOccurs="1" maxOccurs="1">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Participant" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="CommunicationAuthorizationID" minOccurs="1"
maxOccurs="1" >
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:pattern value="\w+"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="OrganizationID" minOccurs="0" maxOccurs="1" >
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:pattern value="\w+"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="ScenarioID" minOccurs="1" maxOccurs="1" >
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:pattern value="[a-f0-9]{8}(?:-[a-f0-9]{4}){3}-[a-f0-9]{12}"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="ReferenceNumber" type="xs:string" minOccurs="0"
maxOccurs="1" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="ExtendedInfo" minOccurs="0">

```

Comment [MOD5]: ScenarioID mandatory – added into scheme

Comment [MOD6]: ReferenceNumber – added into scheme



```

<xs:complexType>
  <xs:sequence>
    <xs:element name="Attribute">
      <xs:complexType>
        <xs:attribute name="Name" type="xs:string" use="required" />
        <xs:attribute name="Value" type="xs:string" use="required" />
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Data" type="xs:anyType"/>
<xs:element xmlns:q1="http://www.w3.org/2000/09/xmldsig#" ref="q1:Signature"
minOccurs="0" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

6.1.2 Business message for Delivery and Confirm

In case of delivery and confirmation, user sends ECC envelope with few specificities. Element "MessageType" must have value "ADM001". Data element refers to "UniqueID" of requested message for delivery or confirmation. Xsd schema for specification of "UniqueID" is "MessageIdentifier" schema, and it is shown below.

Comment [MOD7]: Explanation how to create ADM message

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="MessageIdentifier" >
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:pattern value="[a-f0-9]{8}{?:-[a-f0-9]{4}}{3}-[a-f0-9]{12}"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:schema>

```

6.1.3 Example of XML Envelope before signing

Comment [MOD8]: We changed examples to real ones from test.

```

<ECC>
  <Header>
    <OperationType>SEND</OperationType>

```

Comment [MOD9]: Example from test: valid trader message with valid response from Core



```

<UniqueID>65b1510f-d735-4952-8a6d-0f7d6bfe1124</UniqueID>
<Version>1.0</Version>
<Domain>GMS</Domain>
<Message>
  <MessageType>ND026A</MessageType>
</Message>
<Participants>
  <Participant>
    <CommunicationAuthorizationID>13CZ510000EC00028</CommunicationAuthoriz
ationID>
    <OrganizationID>100394832</OrganizationID>
    <ScenarioID>65b1510f-d735-4952-8a6d-0f7d6bfe1124</ScenarioID>
    <ReferenceNumber>14RS123456N100110</ReferenceNumber>
    <AppID>TraderApp</AppID>
    <AppVersion>v1.1</AppVersion>
  </Participant>
</Participants>
</Header>
<Data>
  <ND026A>
    <Header>
      <RequestID>Request1</RequestID>
    </Header>
    <CustomsOfficeGuarantee>
      <ReferenceNumber>RS123456</ReferenceNumber>
    </CustomsOfficeGuarantee>
    <Principal>
      <TIN>CZDIPLOMATI</TIN>
    </Principal>
    <Guarantee>
      <Type>1</Type>
      <Regime>NT</Regime>
      <GuaranteeReference>
        <GRN>14RS123456N100110</GRN>
        <AccessCodePrimary>KXCU</AccessCodePrimary>
        <AccessCodes>
          <AccessCode>1234</AccessCode>
        </AccessCodes>
        <AccessCodes>
          <AccessCode>4321</AccessCode>
        </AccessCodes>
      </GuaranteeReference>
    </Guarantee>
  </ND026A>
</Data>
</ECC>

```

6.1.4 Example of XML Envelope after signing

```

<ECC>
  <Header>
    <OperationType>SEND</OperationType>
    <UniqueID>65b1510f-d735-4952-8a6d-0f7d6bfe1124</UniqueID>
    <Version>1.0</Version>
    <Domain>GMS</Domain>

```

Comment [MOD10]: Example from test: valid trader message with valid response from Core



European Union

Technical Assistance to the Customs Administration of Serbia to Support the Modernisation of the Customs Systems



```

<Message>
  <MessageType>ND026A</MessageType>
</Message>
<Participants>
  <Participant>
    <CommunicationAuthorizationID>13CZ510000EC00028</CommunicationAuthoriz
ationID>
    <OrganizationID>100394832</OrganizationID>
    <ScenarioID>65b1510f-d735-4952-8a6d-0f7d6bfe1124</ScenarioID>
    <AppID>TraderApp</AppID>
    <AppVersion>v1.1</AppVersion>
  </Participant>
</Participants>
</Header>
<Data>
  <ND026A>
    <Header>
      <RequestID>Request1</RequestID>
    </Header>
    <CustomsOfficeGuarantee>
      <ReferenceNumber>RS123456</ReferenceNumber>
    </CustomsOfficeGuarantee>
    <Principal>
      <TIN>CZDIPLOMATI</TIN>
    </Principal>
    <Guarantee>
      <Type>1</Type>
      <Regime>NT</Regime>
      <GuaranteeReference>
        <GRN>14RS123456N100110</GRN>
        <AccessCodePrimary>KXCU</AccessCodePrimary>
        <AccessCodes>
          <AccessCode>1234</AccessCode>
        </AccessCodes>
        <AccessCodes>
          <AccessCode>4321</AccessCode>
        </AccessCodes>
      </GuaranteeReference>
    </Guarantee>
  </ND026A>
</Data>
<Signature Id="Creator" xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315#WithComments" />
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
    <Reference URI="">
      <Transforms>
        <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature" />
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <DigestValue>2IyaOXdTgh+nNAtaxCswYbrFVRs=</DigestValue>
    </Reference>
    <Reference Type="http://uri.etsi.org/01903/v1.3.2#SignedProperties" URI="#Signe
dProperties">
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />

```




```

</SigningCertificate>
</SignedSignatureProperties>
</SignedProperties>
</QualifyingProperties>
</Object>
</Signature>
</ECC>

```

6.1.5 Example of XML Envelope for Delivery

```

<ECC>
<Header>
<OperationType>DELIVER</OperationType>
<UniqueID>0e57c868-9146-4f63-9c2c-806077a73942</UniqueID>
<Version>1.0</Version>
<Domain>NCTS</Domain>
<Message>
<MessageType>ADM001</MessageType>
</Message>
<Participants>
<Participant>
<CommunicationAuthorizationID>1245689785</CommunicationAuthorizationID>
<OrganizationID>100394832</OrganizationID>
<ScenarioID>15eda370-0668-45b7-b22b-125b309918c0</ScenarioID>
<AppID>TraderApp</AppID>
<AppVersion>v1.1</AppVersion>
</Participant>
</Participants>
</Header>
<Data>
<MessageIdentifier>7eb17fec-753a-4b8b-a3c7-edaa51d59003</MessageIdentifier>
</Data>
</ECC>

```

Comment [MOD11]: Included ScenarioID into ADM message

6.1.6 Example of XML Envelope for Confirm

```

<ECC>
<Header>
<OperationType>CONFIRM</OperationType>
<UniqueID>199d4811-3776-45f1-9807-cc47a919f499</UniqueID>
<Version>1.0</Version>
<Domain>NCTS</Domain>
<Message>
<MessageType>ADM001</MessageType>
</Message>
<Participants>
<Participant>
<CommunicationAuthorizationID>1245689785</CommunicationAuthorizationID>
<OrganizationID>100394832</OrganizationID>
<ScenarioID>ecd7009f-e26c-4424-bb3e-4e12f66acb04</ScenarioID>
<AppID>TraderApp</AppID>
<AppVersion>v1.1</AppVersion>
</Participant>
</Participants>

```



European Union



```

</Header>
<Data>
  <MessageIdentifier>
    7eb17fec-753a-4b8b-a3c7-edaa51d59003
  </MessageIdentifier>
</Data>
</ECC>

```

6.2 Response message

This is the response message from ECC GW. ECCResponse consists of ResponseType and ResponseData.

Response type could be: Acknowledgement, ECC Envelope or Message identifiers and depending on Response type, schemas of Response data differs. Schemas are shown below, except for the ECC which is shown in chapter 6.1.1.

If Response Type is ECC Envelope then OperationType element in the **Header** is not used.

Comment [MOD12]: Explanation of OperationType element

6.2.1 ECC response Xml schema

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema" >
  <xs:element name="ECCResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ResponseType">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="ACK"/>
              <xs:enumeration value="ECC"/>
              <xs:enumeration value="LIST"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="ResponseData" type="xs:anyType">
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>

```

Acknowledgement XML schema

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema" >
  <xs:element name="Acknowledgement" >
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Result" minOccurs="1" maxOccurs="1" >
          <xs:simpleType>
            <xs:restriction base="xs:string" >
              <xs:enumeration value="ACK"/>
              <xs:enumeration value="NAK"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```



```

</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="Reference" minOccurs="0" maxOccurs="1" >
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-f0-9]{8}(?:-[a-f0-9]{4}){3}-[a-f0-9]{12}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="DateTime" type="xs:string" minOccurs="1" maxOccurs="1" />
<xs:element name="errCode" minOccurs="0" maxOccurs="1" nillable="1"/>
<xs:element name="ErrType" type="xs:string" minOccurs="0" maxOccurs="1" />
<xs:element name="ErrDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="ErrData" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

MessageIdentifiers XML schema

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema" >
  <xs:element name="MessageIdentifiers">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="MessageIdentifier">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:pattern value="[a-f0-9]{8}(?:-[a-f0-9]{4}){3}-[a-f0-9]{12}"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

6.2.2 Example of ACK Message

```

<?xml version="1.0" encoding="utf-8"?>
<ECCResponse>
  <ResponseType>ACKNOWLEDGEMENT</ResponseType>
  <ResponseData>
    <Acknowledgement>
      <Result>ACK</Result>
      <Reference>82b1510f-d735-4952-8a6d-0f7d6bfe79</Reference>
      <DateTime>11/5/2013 10:51:28 AM</DateTime>
    </Acknowledgement>
  </ResponseData>
</ECCResponse>

```




6.2.3 Example of NAK Message

```
<ECCResponse>
  <ResponseType>Acknowledgement</ResponseType>
  <ResponseData>
    <Acknowledgement>
      <Reference>9991510f-d735-4952-8a6d-0f7d6bfe7924</Reference>
      <Result>NAK</Result>
      <DateTime>7/4/2014 3:23:19 PM</DateTime>
      <errCode>ERR201</errCode>
      <ErrorType>Security preverification failed</ErrorType>
      <ErrorDescription>Signature is not valid. Signed XML document not valid, cannot
verify signature.</ErrorDescription>
      <ErrorData>Signature is not valid. Signed XML document not valid, cannot verify
signature.</ErrorData>
    </Acknowledgement>
  </ResponseData>
</ECCResponse>
```

Comment [MOD13]: Corrected NAK response

6.2.4 Example of MessageIdentifiers

```
<?xml version="1.0" encoding="utf-8"?>
<ECCResponse>
  <ResponseType>MESSAGEIDENTIFIERS</ResponseType>
  <ResponseData>
    <MessageIdentifiers>
      <MessageIdentifier>199d4811-3776-45f1-9807-cc47a919f499</MessageIdentifier>
      <MessageIdentifier>bd71adfb-e0b4-4fc0-b745-e5ea24be15e3</MessageIdentifier>
      <MessageIdentifier>265f6b06-304d-45bf-9d36-9d53354ac536</MessageIdentifier>
    </MessageIdentifiers>
  </ResponseData>
</ECCResponse>
```

6.2.5 Example of signed outgoing message

```
<ECCResponse>
  <ResponseType>ECC</ResponseType>
  <ResponseData>
    <ECC>
      <Header>
        <UniqueID>0762da0c-bd84-4a94-8ce7-0d9eaf8ab44a</UniqueID>
        <Version>1.0</Version>
        <Domain>GMS</Domain>
        <Message>
          <MessageType>ND223A</MessageType>
        </Message>
        <Participants>
          <Participant>
            <CommunicationAuthorizationID>13CZ510000EC00028</CommunicationAut
horizationID>
            <ScenarioID>65b1510f-d735-4952-8a6d-0f7d6bfe1124</ScenarioID>
          </Participant>
          <Participant>
            <CommunicationAuthorizationID>CAS</CommunicationAuthorizationID>

```

Comment [MOD14]: Response from Core



```

    <OrganizationID>101685102</OrganizationID>
    <ScenarioID>8824e5b7-82ec-446f-8826-bca74fe233d9</ScenarioID>
    <AppID>ECCGateway</AppID>
    <AppVersion>1.0</AppVersion>
  </Participant>
</Participants>
</Header>
<Data>
  <ND223A>
    <Header>
      <RequestID>Request1</RequestID>
    </Header>
    <CustomsOfficeGuarantee>
      <ReferenceNumber>RS123456</ReferenceNumber>
    </CustomsOfficeGuarantee>
    <Principal>
      <TIN>CZDIPLOMATI</TIN>
    </Principal>
    <Guarantee>
      <Type>1</Type>
      <Regime>NT</Regime>
      <GuaranteeReference>
        <GRN>14RS123456N100110</GRN>
        <AccessCodes>
          <AccessCode>1234</AccessCode>
        </AccessCodes>
        <AccessCodes>
          <AccessCode>4321</AccessCode>
        </AccessCodes>
      </GuaranteeReference>
    </Guarantee>
  </ND223A>
</Data>
  <Signature Id="idSignature" xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-
c14n-20010315#WithComments" />
      <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1" />
      <Reference URI="">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature" />
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
        <DigestValue>8yB2MCihl2fpbTR8/Y94dhp5/Iw=</DigestValue>
      </Reference>
      <Reference URI="#SignedProperties" Type="http://uri.etsi.org/01903/v1.3.2#
SignedProperties">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
        <DigestValue>C+ZmcothEmVW/3QiR3FR6oovjPU=</DigestValue>
      </Reference>
    </SignedInfo>
  </Signature>

```



```

<SignatureValue>XXWFe5xCi4r7WSeIFG5eCKVVULURJGbOrIhgwV4MzEZTDS7kwg
vIJq99061DyiQJK//Z8tQ6PcdFTzq7YlwC2rZHRlIWXGJNyg6z1a7SpZe00hkZbzFdUX/ptVAyehBf
Q4oG4XNb6VRR81/mUHh7yhAmiqKI8r5H26n31IH/YK0=</SignatureValue>
<KeyInfo>
  <X509Data>
    <X509Certificate>MIIB3zCCAUYgAwIBAgIQnGjCrvINYJICKQYaG3fmLzAJBgUr
DgMCHQUAMA8xDALBgNVBAMTBFNhZ2EwHhcNMjMxMjMwMDAwWhcNMjMxMjMw
MDAwWjAPMQ0wCwYDVQQDEWRTYWdhMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCe
kLmIqGUSmGY4jgE4qlmi8gc0YdaJplziP49gSl8QnWfn744gZEoK3eFGSwKQ4jqB8cDruIN0NpjRzj
0IZLNuxSJ+BsrOyJfUCp2+LEk1CTs8C8/X0kk96i8YQSRHQr1V4MZrydz0Otp0M1xU9p2xviYmTQ
V+rUEa0uoMZjH8wIDAQABoQwQjBAbgNVHQEEOTA3gBASQLx+tK7QUZu5YhOTKzCYoREwD
zENMAAsGA1UEAxMEU2FnYyIQnGjCrvINYJICKQYaG3fmLzAJBgUrDgMCHQUAA4GBAGO83aoILp
N+qQX5FK33/tO80sgvOSEvb+xfzGbidgfOKIspILD+9Vclia82TDTM/RDmyX1FWrlq8XoM8Aujcg3
ZXXKBish0UEa5qE6ni2FBNzucQiqMIWphoNDscJyDvD+fMDPYycOMUIIn6w6xOb2OcbU0oms3hw
0xSOPHXwJsl</X509Certificate>
  </X509Data>
</KeyInfo>
<Object>
  <QualifyingProperties xmlns="">
    <SignedProperties Id="SignedProperties" Target="#idSignature">
      <SignedSignatureProperties>
        <SigningTime xmlns="http://www.saga.rs/#signatureProperties">7/4/
2014 11:12:09 AM</SigningTime>
        <SigningCertificate>
          <CertificateSerialNumber xmlns="http://www.saga.rs/#signaturePro
perties">9C68C2AEF20D60994229061A1B77E62F</CertificateSerialNumber>
          </SignedBy xmlns="http://www.saga.rs/#signatureProperties"CN=S
aga>
        </SigningCertificate>
      </SignedSignatureProperties>
    </SignedProperties>
  </QualifyingProperties>
</Object>
</Signature>
</ECC>
</ResponseData>
</ECCResponse>

```

6.3 WSDL

```

<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"
xmlns:wsa10="http://www.w3.org/2005/08/addressing"
xmlns:tns="http://saga.rs/ncts/services"
xmlns:misc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wxs="http://schemas.xmlsoap.org/ws/2004/09/mex"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:i0="http://tempuri.org/"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wswa="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"

```



European Union

Technical Assistance to the Customs Administration of Serbia to Support the Modernisation of the Customs Systems



```

xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="GatewayService"
targetNamespace="http://saga.rs/ncts/services">
  <wsdl:types>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
      elementFormDefault="qualified" targetNamespace="http://saga.rs/ncts/services">
      <xs:element name="Send">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="envelope" nillable="true" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="SendResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="SendResult" nillable="true" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Poll">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="communicationAuthorizationId" nillable="true"
              type="xs:string"/>
            <xs:element minOccurs="0" name="communicationDomain" nillable="true"
              type="xs:string"/>
            <xs:element minOccurs="0" name="password" nillable="true" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="PollResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="PollResult" nillable="true" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Deliver">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="envelope" nillable="true" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="DeliverResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="DeliverResult" nillable="true"
              type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Confirm">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="envelope" nillable="true" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </wsdl:types>

```



European Union

Technical Assistance to the Customs Administration of Serbia to Support the Modernisation of the Customs Systems



```

</xs:complexType>
</xs:element>
<xs:element name="ConfirmResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="ConfirmResult" nillable="true"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/"
attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/">
  <xs:element name="anyType" nillable="true" type="xs:anyType"/>
  <xs:element name="anyURI" nillable="true" type="xs:anyURI"/>
  <xs:element name="base64Binary" nillable="true" type="xs:base64Binary"/>
  <xs:element name="boolean" nillable="true" type="xs:boolean"/>
  <xs:element name="byte" nillable="true" type="xs:byte"/>
  <xs:element name="dateTime" nillable="true" type="xs:dateTime"/>
  <xs:element name="decimal" nillable="true" type="xs:decimal"/>
  <xs:element name="double" nillable="true" type="xs:double"/>
  <xs:element name="float" nillable="true" type="xs:float"/>
  <xs:element name="int" nillable="true" type="xs:int"/>
  <xs:element name="long" nillable="true" type="xs:long"/>
  <xs:element name="QName" nillable="true" type="xs:QName"/>
  <xs:element name="short" nillable="true" type="xs:short"/>
  <xs:element name="string" nillable="true" type="xs:string"/>
  <xs:element name="unsignedByte" nillable="true" type="xs:unsignedByte"/>
  <xs:element name="unsignedInt" nillable="true" type="xs:unsignedInt"/>
  <xs:element name="unsignedLong" nillable="true" type="xs:unsignedLong"/>
  <xs:element name="unsignedShort" nillable="true" type="xs:unsignedShort"/>
  <xs:element name="char" nillable="true" type="tns:char"/>
  <xs:simpleType name="char">
    <xs:restriction base="xs:int"/>
  </xs:simpleType>
  <xs:element name="duration" nillable="true" type="tns:duration"/>
  <xs:simpleType name="duration">
    <xs:restriction base="xs:duration">
      <xs:pattern value="\-?P(\d*D)?(T(\d*H)?(\d*M)?(\d*(\.\d*)?S)?)?"/>
      <xs:minInclusive value="-P10675199DT2H48M5.4775808S"/>
      <xs:maxInclusive value="P10675199DT2H48M5.4775807S"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="guid" nillable="true" type="tns:guid"/>
  <xs:simpleType name="guid">
    <xs:restriction base="xs:string">
      <xs:pattern value="[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:attribute name="FactoryType" type="xs:QName"/>
  <xs:attribute name="Id" type="xs:ID"/>
  <xs:attribute name="Ref" type="xs:IDREF"/>
</xs:schema>
</wsdl:types>

```



European Union

Technical Assistance to the Customs Administration of Serbia to Support the Modernisation of the Customs Systems



```

<wsdl:message name="IGatewayService_Send_InputMessage">
  <wsdl:part name="parameters" element="tns:Send"/>
</wsdl:message>
<wsdl:message name="IGatewayService_Send_OutputMessage">
  <wsdl:part name="parameters" element="tns:SendResponse"/>
</wsdl:message>
<wsdl:message name="IGatewayService_Poll_InputMessage">
  <wsdl:part name="parameters" element="tns:Poll"/>
</wsdl:message>
<wsdl:message name="IGatewayService_Poll_OutputMessage">
  <wsdl:part name="parameters" element="tns:PollResponse"/>
</wsdl:message>
<wsdl:message name="IGatewayService_Deliver_InputMessage">
  <wsdl:part name="parameters" element="tns:Deliver"/>
</wsdl:message>
<wsdl:message name="IGatewayService_Deliver_OutputMessage">
  <wsdl:part name="parameters" element="tns:DeliverResponse"/>
</wsdl:message>
<wsdl:message name="IGatewayService_Confirm_InputMessage">
  <wsdl:part name="parameters" element="tns:Confirm"/>
</wsdl:message>
<wsdl:message name="IGatewayService_Confirm_OutputMessage">
  <wsdl:part name="parameters" element="tns:ConfirmResponse"/>
</wsdl:message>
<wsdl:portType name="IGatewayService">
  <wsdl:operation name="Send">
    <wsdl:input wsaw:Action="http://saga.rs/ncts/services/IGatewayService/Send"
message="tns:IGatewayService_Send_InputMessage"/>
    <wsdl:output
wsaw:Action="http://saga.rs/ncts/services/IGatewayService/SendResponse"
message="tns:IGatewayService_Send_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Poll">
      <wsdl:input wsaw:Action="http://saga.rs/ncts/services/IGatewayService/Poll"
message="tns:IGatewayService_Poll_InputMessage"/>
      <wsdl:output wsaw:Action="http://saga.rs/ncts/services/IGatewayService/PollResponse"
message="tns:IGatewayService_Poll_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Deliver">
      <wsdl:input wsaw:Action="http://saga.rs/ncts/services/IGatewayService/Deliver"
message="tns:IGatewayService_Deliver_InputMessage"/>
      <wsdl:output
wsaw:Action="http://saga.rs/ncts/services/IGatewayService/DeliverResponse"
message="tns:IGatewayService_Deliver_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Confirm">
      <wsdl:input wsaw:Action="http://saga.rs/ncts/services/IGatewayService/Confirm"
message="tns:IGatewayService_Confirm_InputMessage"/>
      <wsdl:output
wsaw:Action="http://saga.rs/ncts/services/IGatewayService/ConfirmResponse"
message="tns:IGatewayService_Confirm_OutputMessage"/>
    </wsdl:operation>
  </wsdl:portType>
<wsdl:binding name="BasicHttpBinding_IGatewayService" type="tns:IGatewayService">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="Send">
    <soap:operation soapAction="http://saga.rs/ncts/services/IGatewayService/Send"

```



European Union



```
style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="Poll">
  <soap:operation soapAction="http://saga.rs/ncts/services/IGatewayService/Poll"
style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="Deliver">
  <soap:operation soapAction="http://saga.rs/ncts/services/IGatewayService/Deliver"
style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="Confirm">
  <soap:operation soapAction="http://saga.rs/ncts/services/IGatewayService/Confirm"
style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="GatewayService">
  <wsdl:port name="BasicHttpBinding_IGatewayService"
binding="tns:BasicHttpBinding_IGatewayService">
  <soap:address location="http:// saga.rs/ncts/services/GatewayService.svc"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```



6.4 List of errors codes

Following table contains list of error codes used in NAK message.

| Error code | Error type | Error description |
|------------|---------------------------------|--|
| ERR101 | Invalid envelope | UniqueID is not valid or missing |
| ERR102 | Invalid envelope | Version is not valid or missing |
| ERR103 | Invalid envelope | Domain is not valid or missing |
| ERR104 | Invalid envelope | MessageType is not valid or missing |
| ERR105 | Invalid envelope | CommunicationAuthorizationID is not valid or missing |
| ERR106 | Invalid envelope | OrganizationID is not valid |
| ERR107 | Invalid envelope | ScenarioID is not valid or missing |
| ERR108 | Invalid envelope | AppID is not valid |
| ERR109 | Invalid envelope | AppVersion is not valid |
| ERR110 | Invalid envelope | OperationType is not valid |
| ERR111 | Invalid envelope | General validation error |
| ERR112 | Invalid envelope | UniqueID is duplicated |
| ERR114 | Invalid envelope | Scenario CAS unknown |
| ERR115 | Invalid envelope | Scenario already exists |
| ERR201 | Security preverification failed | Signature is not valid |
| ERR202 | Security preverification failed | Certificate is not valid |
| ERR203 | Security preverification failed | Certificate chain is not valid |
| ERR204 | Security preverification failed | Certificate is revoked |
| ERR205 | Security preverification failed | General security error |
| ERR301 | Authorization failed | Authorization parameters are not defined |
| ERR302 | Authorization failed | User is not authorized for requested action |
| ERR303 | Authorization failed | General authorization error |
| ERR401 | Message queuing failed | General queuing error |
| ERR402 | Message queuing failed | Message domain is not valid |

Comment [MOD15]: New error types



| | | |
|--------|----------------------------|--|
| | | |
| ERR501 | Message polling error | User is not authorized for requested action |
| ERR502 | Message polling error | General polling error |
| | | |
| ERR601 | Message delivery error | Message type for delivery must be 'ADM001' |
| ERR602 | Message delivery error | Message is not in outgoing queue |
| ERR603 | Message delivery error | General delivery error |
| ERR604 | Message delivery error | Business message is not in correct format |
| | | |
| ERR701 | Message confirmation error | Message type for confirmation must be 'ADM001' |
| ERR702 | Message confirmation error | Message is not in outgoing queue |
| ERR703 | Message confirmation error | General confirmation error |
| ERR704 | Message confirmation error | Business message is not in correct format |
| | | |
| ERR001 | Unexpected error | Unexpected error has occurred |
| ERR002 | Message state | Message is not in correct state |



European Union

Technical Assistance to the Customs Administration of Serbia to Support the Modernisation of the Customs Systems



The contents of this publication are the sole responsibility of Saga-Aquasoft-Komix Consortium and can in no way be taken to reflect the views of the European Union