

**User Manual
Of information system**



D1.4 CS OTE external interface

Part D1.4.3 CS OTE web services interface

This document and its contents are confidential. The document may not be reproduced in whole or in part or shown to third parties or used for purposes other than those for which it was provided without the prior written approval of the company OTE, a.s.

<i>Datum</i>	<i>Description</i>	<i>Revize</i>
19.10.2009	Amendments and clarification of the synchronous / asynchronous processing	1.3.
19.10.2009	Addition of call-back service for simplified communication via EDI (electricity) (see chapter 4.5.2)	1.3.
23.10.2009	Addition of call-back service for ETSO messages transferred from OTE to the participant's system (chapter 4.4.3)	1.4
20.11.2009	Addition of commonGasService web service	1.5
2.12.2009	Ammendments to the gas services: cdsEdigasService – newly NOMRES is possible on input cdsEdigasCallbacksService – newly APERAK is possible on output	1.6.
4.12.2009	StatusRequestService extending – as a output 2 documents can be transfereed together (i.g..ConfirmationReport and AnomalyReport)	1.7.
20.4.2012	New service - commonMarketService	1.9
4.6.2012	SOAP connection test description	1.10
27.10.2015	Revision	1.11
20.01.2017	Add new formats	1.12
04.02.2020	Moving ETSO structures from commonService to commonMarketService	1.13
18.8.2020	Addition of (HTTP) TLS level connection test	1.14

Table of content

1	Introduction.....	5
1.1	Web services	5
1.1.1	Authentication.....	5
1.1.2	WS protocol	6
1.2	S/MIME.....	15
2	Principles of communication scenarios via SOAP channel.....	17
2.1	Synchronous communication scenarios	17
2.2	Asynchronous communication scenarios	17
2.2.1	Procedure communication in mode server-server.....	18
2.2.2	Procedure communication in mode client-server	19
2.2.3	SOAP connection test	20
2.2.4	TLS level connection test (HTTP).....	20
3	Services - Electricity.....	21
3.1	Services for Participant – OTE communication.....	21
3.1.1	CDSService.....	21
3.1.2	MarketService.....	22
3.1.3	ReportService.....	23
3.1.4	CommonService.....	23
3.1.5	CommonMarketService	26
3.2	Services for communication OTE -> Participant	27
3.2.1	CDSCallbackService.....	27
3.2.2	MarketCallbackService.....	28
3.2.3	ReportCallbackService	29
3.2.4	CommonCallbackService	31
3.3	Services for ETSO standard support	31
3.3.1	CapacityService	31
3.3.2	ScheduleService.....	32
3.3.3	ScheduleCallbackService.....	32
3.3.4	StatusRequestService.....	33
3.4	Services for EDI standard support	34
3.4.1	EDIService.....	34
3.4.2	EDICallbackService.....	35
4	Services - Gas	36
4.1	Services for communication Participant ->OTE	36
4.1.1	CDSGasService.....	36
4.1.2	CDSEdigasService	37
4.1.3	CommonGasService	38
4.1.4	ReportGasService	40
4.2	Services for communication OTE->Participant (call-back services).....	41

4.2.1	CDSGasCallbackService	41
4.2.2	CDSEdigasCallbackService.....	42
4.2.3	ReportGasCallbackService	43
5	Services - OZE.....	45
5.1	Services for communication Participant -> OTE	45
5.1.1	RESService	45
5.2	Services for communication OTE->Participant (call-back services).....	46
5.2.1	RESCallbackService	46
6	Example of signed document.....	48

1 Introduction

For automated data exchange between external subjects and CS OTE, it is used:

1. **SOAP protocol v 1.1 communication , SOAP-Document type.**
Transfer is realized at the HTTPS connection level, which ensure communication encryption. This make the confidentiality requirement satisfied. This mode serves for XML structured data exchange only.
2. **SOAP v 1.1 communication, simplified mode.**
Similar to the previous point, data are embedded to one element. This mode serves for EDI format message sending.
3. **SMTP communication with S/MIME message application.**
For S/MIME, it is used only text MIME message, secured with el. signature and encryption. It can be used for XML and other documents formats, e.g. EDI.

The use of the above mentioned communication channels can be combined for one subscriber with respect to the following limitations:

- Output channel resolution (OTE-> subscriber) is performed for the RMP-msgcode combination (for OTE specifications) resp. RMP message type (for ETSO or EDIGAS messages).
- Input channel can be combined arbitrarily - ie data can be sent to the system by any channel (or enter via the web interface).
- At the same time, this setting allows to set up OTE communication with several communication servers on the subscriber side, but always respecting the RÚT-msgcode constraints (for messages according to OTE specification) resp. RMP message type (for ETSO or EDIGAS messages) for one server.

1.1 Web services

Web services called via HTTPS protocol, are primary interface of CS OTE. According to the scenarios, we can distinguish 2 types:

- 1) client - server - initialized always on the part of external participant. Client system must support web compliant with this specification.
- 2) server-server - initilized also from OTE system. There must be web services compliant with this specification on the side of external participant.

1.1.1 Authentication

CS OTE use authentication through client certificate X.509, which is possible to use by transfer over HTTP protocol with SSL expansion. The solution was choosen according to the requirement for sensitive data transfer. From the security point of view, it is a better solution than direct authentisation, using (system) username and password. For private key security, there are better methods than for the password.

So, in the B2B sphere, client certificate is used for unique identification, to whom corresponds system user in CS OTE.

1.1.2 WS protocol

Communication through web services is based on SOAP v 1.1. Particulars of the individual components are dependent on the type of Web service. Web services can be divided into two types:

- Services for the exchange of XML structured data
- Services for EDI data exchange

1.1.2.1 Services for the exchange of XML structured data

1.1.2.1.1 WS header (Element Header)

To increase the security and integrity of data transmitted by SOAP protocol Web services using standard WS-Security. Other standards from WS * category, for example. WS-RM, WS-Trust ... are not used.

1. WS – Security

defined by OASIS syndicate (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss). It consists of following parts:

- **UsernameToken**– used for authentication, **not implemented in CS OTE**, authentication at SSL/TLS level is used
- **Timestamp** – introducing time validity and request time creation in the course of call and SOAP response. In CS OTE web services is mandatory part.
- **Signature** – owing to XML el. signature adoption (XML Signature), it ensures transferred data integrity. It applies for additional integrity check –whol SOAP envelope, already at the level of standard WS implementation. In CS OTE web services is mandatory part
- **Encryption - not implemented in CS OTE**, assured with SSL, not implemented in CS OTE,

WS security - propopsal of use sumarisation

Possibility of use of WS-* standard and their parts is indicated in following table:

Standard	Part	CS OTE application
WS – Security 1.1	Timestamp	Applied
WS – Security 1.1	Signature	Used for integrity check at the standard implementation SOAP protocol level

Certificate, which is used for el. signature must be defined as BinarySecurityToken outside the element, containing el. signature (wlwmwnt Signature). In the SecurityTokenReference segment, it must be referred in the manner defined in the standard as Direct References. Other mode (ie. such as SubjectKeyIdentifier,...) is by CS OTE not supported.

Through el. signature, the integrity is assured in these parts:

- 1) Timestamp (namespace <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd>)
- 2) Body (namespace <http://schemas.xmlsoap.org/soap/envelope/>)

1.1.2.1.2 Message body (Element Body)

XML documents transmitted in the message body can be fitted with an electronic tag in XML Signature format. The ability to embed XML signature is defined by XSD template of transmitted document. It is recommended to always insert XML signatures.

For selected XML documents (CDSINVOICE, CDSDATA , CDSGASINVOICE) is XML Signature compulsory, involves the transmission of these messages among the participants through the CS OTE and end-to-end integrity of messages between participants must be secured.

1. XML signature

W3C recommendation <http://www.w3.org/TR/xmldsig-core/> Enveloped Signature Transformation <http://www.w3.org/2000/09/xmldsig#enveloped-signature> is applied. I.e. the whole XML document is procured with el. stamp, appropriate namespace defined <http://www.w3.org/2000/09/xmldsig#> is inserted before document closing root element (e.g. CDINVOICE, CDSDATA, MASTERDATA etc.). Public certificate of X.509 format must be inserted into the structure, which is specified in stated namespace (KeyInfo/X509Data/X509Certificate). For signing, it is necessary to use a key, with RSA algorithm and SHA-1 or SHA-2 (SHA256, SHA384 nebo SHA512) hash function.

An example of structure of signed document is shown in following example.

2. XML data structure

The structure of XML data is specified by a separate document formats XML, published on the public website of OTE. Complete WSDL documents of Web services for the exchange of XML documents are published alongside with individual templates on OTE web pages. In published WSDL documents

relevant XSD templates are placed directly, and received data are validated against these templates.

Returned answer depends on the type of communication scenario, which is supported by the service. The individual communication scenarios are described in the section Solving communication scenarios using SOAP channel.

Returned answer depends on the type of communication scenario, which is supported by the service. The individual communication scenarios are described in the section Solving communication scenarios using SOAP channel.

Query Example

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
1.1.2.2 <SOAP – zjednodušená varianta-ENV:Header>
  <wsse:Security SOAP-ENV:mustUnderstand="1">
    <wsu:Timestamp wsu:Id="Timestamp-445df5c5-e7cc-4fac-89bd-e2ba0a40161b">
      <wsu:Created>2013-10-20T12:04:01Z</wsu:Created>
      <wsu:Expires>2013-10-20T14:04:01Z</wsu:Expires>
    </wsu:Timestamp>
    <wsse:BinarySecurityToken EncodingType="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"
ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-
profile-1.0#X509v3" wsu:Id="Id-73eaed6c-6df0-4208-b3f4-
3e77ffa4f2d7">MIEXDCCA0SgAwIBAgIDH1NEMA0GCSqGSIb3DQEBCwUAMIG.
..</wsse:BinarySecurityToken>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
        <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"/>
        <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
        <Reference URI="#Timestamp-445df5c5-e7cc-4fac-89bd-e2ba0a40161b">
          <Transforms>
```



```
<Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
</Transforms>
<DigestMethod Algorithm="http://www.w3.org/2000/09/xmlsig#sha1" />
<DigestValue>J3wRVLnWfYXvBhqm/pDDTtp5iiw=</DigestValue>
</Reference>
<Reference URI="#Id-acf95827-208c-42eb-afad-87fd72dd116d">
<Transforms>
<Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
</Transforms>
<DigestMethod Algorithm="http://www.w3.org/2000/09/xmlsig#sha1" />
<DigestValue>/yAjHPVhvyX3DrfiD6/esAiha5s=</DigestValue>
</Reference>
</SignedInfo>
<SignatureValue>QLGzH5SeVsmIdVfS1P...</SignatureValue>
<KeyInfo>
<wsse:SecurityTokenReference>
<wsse:Reference URI="#Id-73eae6c-6df0-4208-b3f4-3e77ffa4f2d7"
Value Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-
profile-1.0#X509v3" />
</wsse:SecurityTokenReference>
</KeyInfo>
</Signature>
</wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body wsu:Id="Id-acf95827-208c-42eb-afad-87fd72dd116d">
<SendRequest xmlns="http://www.ote-cr.cz/schema/service/common/market">
<COMMONMARKETREQ xmlns="http://www.ote-
.cz/schema/common/market/request" date-time="2013-10-20T12:04:02" dtd-release="1"
dtd-version="1" id="000001" message-code="923">
<SenderIdentification coding-scheme="14" id="XXXXXXXXXXXX" />
<ReceiverIdentification coding-scheme="14" id="8591824000007" />
</COMMONMARKETREQ>
</SendRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Reply example

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soap env="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<wsse:Security xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-utility-1.0.xsd" xmlns:wsse="http://docs.oasis-
d1.4.3-en-web-services-interface-v1.14.doc
© 2020 OTE, a.s.
```

```
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
soapenv:mustUnderstand="1">
  <wsse:BinarySecurityToken EncodingType="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"
ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-
profile-1.0#X509v3" wsu:Id="X509-
ADC8F997FD2241EA361445342641986626059">MIIFijCCA...</wsse:BinarySecurityT
oken>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="SIG-509913">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
        <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-
c14n#" PrefixList="soapenv"/>
      </ds:CanonicalizationMethod>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1"/>
      <ds:Reference URI="#TS-509909">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
            <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-
c14n#" PrefixList="wsse soapenv"/>
          </ds:Transform>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>9wgiEUCZQbzIFGWhLsa9Z3nrJAs=</ds:DigestValue>
      </ds:Reference>
      <ds:Reference URI="#id-509911">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
            <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-
c14n#" PrefixList=""/>
          </ds:Transform>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>Hg3ezC7zFvEzgzki2fHH55lefRm8=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>mr35bjHjBL...</ds:SignatureValue>
    <ds:KeyInfo Id="KI-ADC8F997FD2241EA361445342641986626061">
      <wsse:SecurityTokenReference wsu:Id="STR-
ADC8F997FD2241EA361445342641986626062">
```

```
<wsse:Reference URI="#X509-ADC8F997FD2241EA361445342641986626059"
ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-
profile-1.0#X509v3"/>
  </wsse:SecurityTokenReference>
</ds:KeyInfo>
</ds:Signature>
<wsu:Timestamp wsu:Id="TS-509909">
  <wsu:Created>2013-10-20T12:04:01Z</wsu:Created>
  <wsu:Expires>2013-10-20T12:09:01Z</wsu:Expires>
</wsu:Timestamp>
</wsse:Security>
</soapenv:Header>
<soapenv:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" wsu:Id="id-509911">
  <SendResponse xmlns="http://www.ote-cr.cz/schema/service/common/market">
    <RETURN_CODE xmlns="http://www.ote-
cr.cz/schema/service/globals">0</RETURN_CODE>
    <RESPONSE xmlns="http://www.ote-cr.cz/schema/response" date-time="2013-10-
20T14:03:44" dtd-release="1" dtd-version="1" id="000001" message-code="932">
      <SenderIdentification coding-scheme="14" id="8591824000007"/>
      <ReceiverIdentification coding-scheme="14" id="XXXXXXXXXXXXXX"/>
      <Reason code="5521" result-code="M25521" type="A03">(MSG5521)</Reason>
    </RESPONSE>
    <ISOTEDATA xmlns="http://www.ote-
cr.cz/schema/market/data">...</ISOTEDATA>
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xm1-
c14n-20010315"/>
        <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1"/>
        <ds:Reference URI="">
          <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"/>
          </ds:Transforms>
          <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <ds:DigestValue>5pvliVTKB92EqK6JrdVy84PBJlo=</ds:DigestValue>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>E609lZJkmNqiYPTcYefo1HNIvySz...</ds:SignatureValue>
    </ds:Signature>
  </ds:KeyInfo>
  <ds:X509Data>
```

```
<ds:X509Certificate>MIIFijCC...</ds:X509Certificate>
</ds:X509Data>
<ds:KeyValue>
  <ds:RSAKeyValue>
    <ds:Modulus>sYSbEDLfhR...</ds:Modulus>
    <ds:Exponent>AQAB</ds:Exponent>
  </ds:RSAKeyValue>
</ds:KeyValue>
</ds:KeyInfo>
</ds:Signature>
</ISOTEDATA>
</SendResponse>
</soapenv:Body>
</soapenv:Envelope>
```

1.1.2.2 Services for EDI data exchange

For client support, that generate data in EDI format and send them to CS OTE system, the service interface is simplified.

1.1.2.2.1 *WS header*

This part is not used in the case of EDI message sending.

1.1.2.2.2 *Message body (Element Body)*

SOAP call has only 1 DATA parameter (element), which has to contain el. signed message in PKCS#7 format with base64 coding.

EDI document is inserted into DATA element. EDI message have to have binary electrocinal signature (in PKCS#7 format) a be coded by BASE64 algorithm. As an answer RETURN CODE element with following meaning is returned.

- 0 – data received correctly, internal record created
- 1 – error in signature verification – message is not signed correctly
- 2 – error in transformation – message is not in expected structure for processing
- 3 – error in internal record creation – general error

In the opposite case the request is rejected and standard SOAP: Fault error message is sent back with reason of refection.

Example of a request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" >
```

```
<SOAP-ENV:Body>
  <ns1:Z_CDS_DATA_TRANSFER xmlns:ns1="urn:sap-
com:document:sap:rfc:functions" >
  <ns1:DATA
xsi:type="xsd:string">MIAGCSqGSIB3DQEHAqCAMIACAQExCzAJBgUrDg
MCGGUAMIAGCSqGSIB3DQEHAaCAJIAEggIppD94
bWwgdMvyc21vbj0iMS4wIiBlbmNvZGluZz0iaXNvLTg4NTktMiI/PjxSRVNQ
T05TRSB4bWxucz0i
aHR0cDovL2Nkcy5vdGUtY3IuY3oiIHhtbG5zOnhzaT0iaHR0cDovL3d3dy53
My5vcmcvMjAwMS9Y
TUxTY2h1bWEtaW5zdGFuY2UiIHhzaTpzY2h1bWFMb2NhdGlvbj0iaHR0cDov
L2Nkcy5vdGUtY3Iu
Y3ogLi9SRVNQT05TRS54c2Q2IiG1kPSI4MTAwMDAwMDM5NzQzMyIgbWVzc2Fn
ZS1jb2RlPSI5NzIi
IGRhdGUtdGltZT0iMjAwOS0wNi0xNFQyMT0zMzoxNyIgaHR0cDovL3d3dy53
IjEiIGR0ZC1yZWxl
YXN1PSIxIj48U2VuZGVySWRlbnRpZmljYXRpb24gaWQ9IjI3WE9URS1DWkVD
SFJFUEIiIGNvZGlu
Zy1zY2h1bWU9IjE0Ii8+PFJlY2VpdMvYySWRlbnRpZmljYXRpb24gaWQ9Ijg1
OTE4MjQwMTA3MDki
IGNvZGluZy1zY2h1bWU9IjE0Ii8+PFJlZmVvZW5jZS8+PFJlYXNvbiBjb2Rl
PSIzNDIyIiB0eXB1
PSJBMDMiPiBCEwXhIHBYb3ZlZGVuYSBhZ3JlZ2FjZSAyNCB0b2RpbngvVkrU
IHBYbyBvYmNob2Ru
7SBkZW4gMTQuMDYuMjAwOS48L1JlYXNvbj48L1JFU1BPT1NFPgAAAAAAKCC
A2kwggN1MIICTaAD
AgECAgokjBJ+AAUAABJrMA0GCSqGSIB3DQEBAQUAMEIxCzAJBgNVBAYTAKNa
MQ8wDQYDVQQKEwZM
b2dpY2ExEjAQBGNVBAStTCVBLSSBHcm91cDEOMAwGA1UEAxMFT1RFQ0EwHhcN
MDgwODE4MTMxNTAw
WhcNMTAwODE4MTMxNTAwWjBgMRwwGgYJKoZIhvcNAQkBFg1jZHNAb3RlZGV2
LmN6MQswCQYDVQQG
EwJDWjEMMAoGA1UEChMFT1RFMRwEQYDVQQLEwPQS0kgU2VydMvYyMRAwDgYD
VQQDEwdDRFMgRGV2
MIGfMA0GCSqGSIB3DQEBAQUAA4GNADCBiQKBgQDlNKn6kornVKntg/12q4dl
cCg/XETa6ezwIqyf
Iw6Uht8878C2DvcqOAcNIRF71591vcLJ14t1wiY0P087PGd4VbRbpq8NjZCG
GXhZ1N9i7uXTRWs1
h2/11SWFZ+WpRG5w3wtQCi4DdCgxM+WINKWo/T1EIwDPrQk8Jf8r0vLceQID
AQABo4HCMIG/MA4G
A1UdDwEB/wQEAWIE8DATBgNVHSUEDDAKBggrBgEFBQcDBDAdBgNVHQ4EFgQU
f7iMNRGec9iFS42P
```

```
DERQyqJvodIweQYDVR0jBHIwcIAU9PFX94g6H15Uduw768RPh+XYj9GhRqRE
MEIx CzAJBgNVBAYT
AkNaMQ8wDQYDVQQKEwZMb2dpY2ExEjAQBgNVBAsTCVBLSSBHcm91cDEOMAwG
A1UEAxMFT1RFQ0GC
EHmDQl4+i3+jTXsUxM7X7XwwDQYJKoZIhvcNAQEFBQADggEBAGC11RY8JNxo
bemu00wkhJZiJnjg
GGWm39fjQbiyQnW5DdXpzbHVqOQ1f0qcmbU0C7SiAFgHf+D2Ob7rcODMBm+9
j06z2MSyXhID8h8j
h8icRwZH6tPYHNZSdQY+EfI6CsfAGgE7bi9fQGQy9j8gPTz2W+7PcW64ZBk
poHGZHEcnjkyR0G3
kui3SiSSxJ9mHgDtYoGx1pzhWnDNMefB89iF4gDRTLdRSeRZcy8e0r0naRU+
cWiwg5sl4M/Qvo/P
nbwAUcAQViMyasJVqhDl31LChz0mlTo05vEmCKt7+z058cD2r11tWOBwJyyL
4HsVm86Yze+2hGge
V4eCYzPydwMxggFVMIIBUQIBATBQMEIx CzAJBgNVBAYTAkNaMQ8wDQYDVQQK
EwZMb2dpY2ExEjAQBgNVBAsTCVBLSSBHcm91cDEOMAwGA1UEAxMFT1RFQ0ECCiSMEn4ABQAAEmsw
CQYFKw4DAhoFAKBd
MBgGCSqGSib3DQEJAzELBgkqhkiG9w0BBwEwHAYJKoZIhvcNAQkFMQ8XDTA5
MDYxNDE5MzYxNlow
IwYJKoZIhvcNAQkEMRYEFIKXKbZ9J3otwY320sSBNPTES5OrMA0GCSqGSib3
DQEBAQUABIGAuWm1
f+i59S7LtoCJG2//O31V2F1dvRa+WY2jyuHjofUWI7sPSKBzwr fHRlepUozc
i3SonifjqCcxnwdT
Q+1rdhfCtnWurGvcCS9hAK1LtHNPdXpC4Mgwf7g01cKXN/BHcagR2wy80+kw
8vXvy9aqtDjzUD/h
1mZtYjeGDkrQG9IAAAAAAAAA=</ns1:DATA>
  </ns1:Z_CDS_DATA_TRANSFER>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Exemple of a response:

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:edi="http://www.ote-cr.cz/schema/service/edi"
xmlns:globals="http://www.ote-cr.cz/schema/service/globals">
  <soapenv:Header/>
  <soapenv:Body>
    <edi:SendDataRequest>
      <globals:RETURN_CODE>2</globals:RETURN_CODE>
    </edi:SendDataRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

1.2 S/MIME

As back-up communication channel, it is possible to use SMTP protocol. Data have to be enclosed as an e-mail message attachment, which is sealed up according to S/MIME (<http://www.ietf.org/rfc/rfc2633.txt> resp. <http://www.ietf.org/rfc/rfc3851.txt>), which defines MIME expansion with use of el. signature and encryption according to PKCS#7 RSA standard.

Procedure for creating S/MIME message for CS OTE:

- 1) Data message (XML, EDI) has to be embeded into the MIME message in the form of attachment (Content-Disposition: attachment;).
- 2) Over all MIME message content it is applied el. signature according to PKCS#7 standard – „signed data“ format and Base64 encoded. Hash function SHA1 with RSA encryption is used. This content is indicated by corresponding header.
- 3) Thus created message is encrypted by receiver's public key. For symmetric message encryption, it is used 3DES block cipher in CBC mode for the reason of e-mail clients compatibility. Encrypted message have to be again encoded with Base64 procedure. Afterward it is supplied with appropriate S/MIME header and with other MIME message standard attributes.

Examples of message:

- 1) MIME message with data attachment

```
From: <cds@otedev.cz>  
Subject: VVT message  
To: <CDSADMIN@OTEDEV.CZ>  
Date: 13.05.2009 21:32:28 +0100  
MIME-Version: 1.0  
Content-Type: multipart/mixed;  
    boundary="-----_NextPart_000_13.05.2009_21:32:28_CDS"  
Importance: Normal  
X-Priority: 3 (Normal)  
X-Mailer: SAP Web Application Server 6.20
```

This is a multi-part message in MIME format.

```
-----=_NextPart_000_13.05.2009_21:32:28_CDS  
Content-Disposition: inline  
Content-Type: text/plain;  
    charset=us-ascii;
```

Content-Transfer-Encoding: quoted-printable
Content-Description: VVT message

-----=_NextPart_000_13.05.2009_21:32:28_CDS
Content-Type: application/octet-stream;
name="0081000000395197.edn"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
filename="0081000000395197.edn"

VU5BOisuPyAnVU5CK1VOT0M6Mys4NTkxODI0MDAwMDA3Oje0Kzg1OTE4MjQw
MDEwMDQ6MTQrMDkwNTEzOjIxMzIroDEwMDAwMDAzOTUxOTcrKysrKydvTkg
OTcyK0FQRVJBSzpeOjk2QTpaWjpfRElDWjEnQkdNKzEyRTo6OSs4MTAwMDAw
MDM5NTE5Nys0NSsnRFRNKzEzNzoyMDA5MDUxMzIxMzI6MjAzJ1JGRisrOidO
QUQrU08rODU5MTgyNDAwMDAwNzo6OSdFUkMrMDAwOjo2MCdGVFgrVFJEKysr
T1pOwU1FTs0gTyBQUk9WRURFTs0gQUdSRUdBQ0UgVkrUCgonRlRYK1RSRCsr
KzM0MjItIEJ5bGEgcHJvdmVkdW5hIGFncmVnYWw1IDl0IGhvZGlueSBWRFQg
cHJvIG9iY2hvZG467SBkZW4gMTMuMDUuMjAwOS4nrRlRYK1RSRCsrKwoKSVMg
T1RFCidVTlQrMTArOTcyJ1VOWisxKzk3Mic=

-----=_NextPart_000_13.05.2009_21:32:28_CDS--

2) signed S/MIME message header

Content-Type: application/x-pkcs7-mime; name=smime.p7m;
smime-type=signed-data
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7m"
Content-Description: S/MIME Cryptographic Signed Data

3) encrypted S/MIME message header

Content-Type: application/pkcs7-mime; name="smime.p7m";
smime-type=enveloped-data
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7m"
Content-Description: S/MIME Encrypted Message
To: cdsadmin@otedev.cz
From: cds@otedev.cz

2 Principles of communication scenarios via SOAP channel

2.1 Synchronous communication scenarios

The response / output message is provided in this scenarios directly as a output of the SOAP services call.

Via synchronous method the following groups of scenarios are solved:

- **Scenarios of „Elektricity“ area provided via „MarketService“ service, with exception of any requests for data** (i.e. allways when using the ISOTEDATA.xsd structure you will get the synchronous response)
- **Scenarios of „Elektricity“ prvided via services for ETSO messages CapacityService, ScheduleService, StatusRequestService.**
- **Scenarios of „Gas“ area for nominations via CDSEdigasService and Nomination.xsd structure**

2.2 Asynchronous communication scenarios

The response / output message is provided in this scenarios via one of the following methods according to the participant’s preferences:

- The response is handed over to the call-back service implemented on the participant’s system according to the OTE standards (push method). Definiton of callback services together with services for data receving are published on CS OTE public portal.
- Send to participant cue in CS OTE system and ready to be pickedup by participant (Pool method)
- The response is handed over via SMTP channel to participants email.

Via asynchronous method the following groups of scenarios are solved:

- **All scenarios solved via CDSService service.** The CS OTE response is provided on the CDSCallBackService or one of the above mentioned ways (client-server mode, SMTP)
- **Scenarios solving the request for data via MarketService service,** i.e. when using this service for ISOTEREQ.xsd structure transfer. The CS OTE response is provided on the MarketCallBackService or one of the above mentioned ways (client-server mode, SMTP)
- **All scenarios solved via ReportService service.** The CS OTE response is provided on the ReportCallBackService or one of the above mentioned ways (client-server mode, SMTP)
- **Scenarios solved via CDSEdigasService with exception of nominations (Nomination.xsd).** The CS OTE response is provided on the

-
- CDSEdigasCallbackService or one of the above mentioned ways (client-server mode, SMTP)
- **All scenarios solved via CDSGasservice.** The CS OTE response is provided on the CDSGasCallbackService or one of the above mentioned ways (client-server mode, SMTP)

Call-back services on the participant's system are used also for provisioning of the data, which emerge from the CS OTE initiatives (notifications, automated data hand-over etc.)

As an answer within asynchronous communication scenario RETURN CODE element with following meaning is returned.

- 0 – data received correctly, internal record created
- 1 – error in signature verification – message is not signed correctly
- 2 – error in transformation – message is not in expected structure for processing
- 3 – error in internal record creation – general error

In the opposite case the request is rejected and standard SOAP: Fault error message is sent back with reason of refection.

Data from CS OTE are provided to the subscriber either automatically as a result of the process (handover of data for invoicing after sending by distributor, handover of electronic invoice, trading results, etc.) Or at the explicit request of the participant.

Typical scenarios of asynchronous automatic communication are following:

1. Send message (data/request) to system CS OTE
2. Asynchronous processing of message in system CS OTE
3. Generating answer and passing it by preferred way to participant

2.2.1 Procedure communication in mode server-server

To support this type of communication scenario, participant has to implement WSDL service according to the relevant regulations callback service on his side.

A typical communication scenario for obtaining the data is as follows:

1. A participant sends an XML structure (CDSREQ, ISOTEQ, CDS GASQ ...) to the appropriate service.
2. Asynchronous message processing by system CS OTE
3. Answer processing is passed to the call-back service to the participants side.

In case of unsuccessful sending a message to the callback service, an attempt is made to resend the message to callback service subscriber (2x). In the case of unsuccessful delivery message is sent to the subscriber by alternative channel.

NOTE

Participant can execute pairing of messages question-answer based on elements of XML structure „message id“ (attribute „id“ root question element – for example <CDSGASREQ id=“XXXX“>) and „reference message id“ (attribute „id“ Reference Elementu – for example. CDSGASMASTERDATA / Reference@id).

This principle is currently applicable to all structures of XML according to OTE specification. XML structure according to standard specifications EDIGAS currently contain an attribute for the placing reference data exchange (except APERAK, where is used in the <OriginalMessageIdentification v = "xxx" />), so it is not that principle be used - ie. It is not possible to pair CDS EDIGASREQ and reports format EDIGAS (except APERAK where this is possible).

2.2.2 Procedure communication in mode client-server

To support this type of communication scenario, the participant does not need to implement callback service on their side, only periodically polls the system CS OTE.

A typical communication scenario for obtaining the data is as follows:

1. A participant sends an XML structure (CDSREQ, ISOTEQ, CDS GASQ ...) To the appropriate service.
2. Asynchronous message processing by system CS OTE
3. Answer is ready for pick up at system CS OTE
4. The participant makes a call to a Web service to pick up messages from CS OTE

CDS OTE exposes 3 services by area:

- **CommonService** (messages from area electricity/OZE except KT), structure COMMOREQ - msg code 921
- **CommonMarketService** (messages KT), structure COMMONMARKETREQ - msg code 923
- **CommonGasService** (messages from area of gas), structure COMMONGASREQ - msg code GX1

The outcome of the call is the first data report prepared in participant's queue of messages on OTE communication server.

If the queue is empty message is handed to the output of the service RESPONSE / GASRESPONSE with the information that the queue is empty - msg-code 922 (message outside KT) and 924 (reports from the field KT) respectively. GASRESPONSE msg-code GX2.

In this way subscriber is fetched from the queue sequentially all prepared messages regardless of whether they are provided automatically as a result of a process or as a response to the query.

NOTE

Participant can execute pairing of messages question-answer based on elements of XML structure „message id“ (attribute „id“ root question element – for example <CDSGASREQ id=“XXXX“>) and „reference message id“ (attribute „id“ Reference Elementu – for example. CDSGASMASTERDATA / Reference@id).

This principle is currently applicable to all structures of XML according to OTE specification. XML structure according to standard specifications EDIGAS currently contain an attribute for the placing reference data exchange (except APERAK, where is used in the <OriginalMessageIdentification v = "xxx" />), so it is not that principle be used - ie. It is not possible to pair CDSSEDIGASREQ and reports format EDIGAS (except APERAK where this is possible).

2.2.3 SOAP connection test

Messages COMMONREQ with code 991 and COMMONMARKETREQ with code 994 send to service CommonService → participant direction connection test. When receiving this messages, CS OTE communication server holds incoming connection and tries to deliver RESPONSE with code 995 resp. 996. If the RESPONSE message is delivered, CS OTE returns synchronously message 995 resp. 996 with Reason/code 997 (“Test message was sended OK.”), otherwise with Reason/code 998 (“Error sending test message.”). If the connection test was successful, CS OTE communication server tries to deliver all messages not older than 3 days for the participant not yet delivered. The 994 message invokes delivery of MARKET messages, 991 message is for all other messages.

2.2.4 TLS level connection test (HTTP)

In exceptional cases, the OTE operator performs a connection test to the market participant's service at the TSL level using a SOAP message, to which the participant does not need to respond in any way. This is just a test of establishing communication with the target server. In the event that such a message would be identified on the part of the subscriber, we enclose a simple XSD template, which this message meets:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="">  
<xsd:element name="TESTCONNECTION"/>  
</xsd:schema>
```

3 Services - Electricity

3.1 Services for Participant – OTE communication

Following services are designed for the purpose of synchronous communication

3.1.1 CDSService

The service includes XML structures from <http://www.ote-cr.cz/schema/service/cds...> namespace, containing following structures:

- CDSREQ.xsd
- CDSCLAIM.xsd
- CDSDATA.xsd
- CDSIDIS.xsd
- TDD.xsd
- MASTERDATA.xsd
- TEMPERATURE.xsd
- CDSINVOICE.xsd
- CDSLDSREPORT.xsd

Operation **send** is defined on the service with following structure:

Input:

```
<xsd:element name="SendRequest">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="request:CDSREQ" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="claim:CDSCLAIM" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="invoice:CDSINVOICE" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="data:CSDATA" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="idis:CDSIDIS" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="masterdata:MASTERDATA" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="tdd:TDD" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="temperature:TEMPERATURE" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="lds:CDSLDSREPORT" minOccurs="1"
maxOccurs="1"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

Output:

```
<xsd:element name="sendResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="globals:RETURN_CODE" minOccurs="1"
maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

3.1.2 MarketService

The service includes XML structures from [http://www.ote-cr.cz/schema/service/market ...](http://www.ote-cr.cz/schema/service/market...) namespace, containing following structures:

- ISOTEREQ.xsd
- ISOTEDATA.xsd
- ISOTEMASTERDATA.xsd
- RESPONSE.xsd

Operation **Send** is defined on the service with following structure:

Input:

```
<xsd:element name="SendRequest">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="request:ISOTEREQ" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="otedata:ISOTEDATA" minOccurs="1"
maxOccurs="1"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

Output:

```
<xsd:element name="SendResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="globals:RETURN_CODE" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="response:RESPONSE" minOccurs="0"
maxOccurs="1"/>
      <xsd:choice minOccurs="0" maxOccurs="1">
        <xsd:element ref="otedata:ISOTEDATA" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="otemasterdata:ISOTEMASTERDATA"
minOccurs="1" maxOccurs="1"/>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```
        </xsd:sequence>  
    </xsd:complexType>  
</xsd:element>
```

3.1.3 ReportService

The service includes XML structures from <http://www.ote-cr.cz/schema/report/...> namespace, containing following structures:

- SFVOTREQ.xsd
- SFVOTSETTINGS.xsd
- RESPONSE.xsd

Operation **Send** is defined on the service with following structure:

Input:

```
<xsd:element name="sendRequest">  
    <xsd:complexType>  
        <xsd:sequence>  
            <xsd:element ref="request:SFVOTREQ" minOccurs="1"  
maxOccurs="1"/>  
            <xsd:element ref="settings:SFVOTSETTINGS" minOccurs="1"  
maxOccurs="1"/>  
        </xsd:sequence>  
    </xsd:complexType>  
</xsd:element>
```

Output:

```
<xsd:element name="sendResponse">  
    <xsd:complexType>  
        <xsd:sequence>  
            <xsd:element ref="globals:RETURN_CODE" minOccurs="1"  
maxOccurs="1"/>  
            <xsd:element ref="response:RESPONSE" minOccurs="0"  
maxOccurs="1"/>  
        </xsd:sequence>  
    </xsd:complexType>  
</xsd:element>
```

3.1.4 CommonService

The service includes XML structures from prostoru <http://www.ote-cr.cz/schema/service/common...> The service contains operations with message-code 921 and 991. It contains following structures:

- COMMONREQ.xsd
- CDSCLAIM.xsd
- CDSINVOICE.xsd

-
- CDSDATA.xsd
 - CDSIDIS.xsd
 - MASTERDATA.xsd
 - TDD.xsd
 - SFVOTLIMITS.xsd
 - SFVOTTDDNETT.xsd
 - SFVOTTDDSUM.xsd
 - TEMPERATURE.xsd
 - SFVOTBILLING.xsd
 - SFVOTCLAIM.xsd
 - SFVOTCLAIMSUM.xsd
 - SFVOTCONFDATA.xsd
 - SFVOTDTEXPIMP.xsd
 - SFVOTBILLINGEMO.xsd
 - SFVOTBILLINGSUM.xsd
 - SFVOTTDD.xsd
 - SFVOTEXCHRATE.xsd
 - SFVOTLIMITCHANGE.xsd
 - SFVOTPSK.xsd
 - acknowledgement-v5r0.xsd
 - confirmation-v3r1.xsd
 - anomaly-v3r1.xsd
 - capacity-document-v4r0.xsd
 - RESPONSE.xsd
 - RESRESPONSE.xsd
 - RESCLAIM.xsd
 - RESDATA.xsd
 - RESSOURCE.xsd
 - RESSETTLDOC.xsd
 - RESFUEL.xsd
 - RESHEAT.xsd
 - RESDELEGATE.xsd

Operation **Send** is defined on the service with following structure:

Input:

```
<xsd:element name="sendRequest">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="request:COMMONREQ" minOccurs="1"
maxOccurs="1"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

d1.4.3-en-web-services-interface-v1.14.doc

© 2020 OTE, a.s.


```
</xsd:complexType>  
</xsd:element>
```

Output:

```
<xsd:element name="SendResponse">  
  <xsd:complexType>  
    <xsd:sequence>  
      <xsd:element ref="globals:RETURN_CODE" minOccurs="1"  
maxOccurs="1"/>  
      <xsd:element ref="response:RESPONSE" minOccurs="0"  
maxOccurs="1"/>  
      <xsd:choice minOccurs="0" maxOccurs="1">  
        <xsd:element ref="claim:CDSCLAIM" minOccurs="1"  
maxOccurs="1"/>  
        <xsd:element ref="invoice:CDSINVOICE" minOccurs="1"  
maxOccurs="1"/>  
        <xsd:element ref="data:CDSDATA" minOccurs="1"  
maxOccurs="1"/>  
        <xsd:element ref="idis:CDSIDIS" minOccurs="1"  
maxOccurs="1"/>  
        <xsd:element ref="masterdata:MASTERDATA" minOccurs="1"  
maxOccurs="1"/>  
        <xsd:element ref="tdd:TDD" minOccurs="1" maxOccurs="1"/>  
        <xsd:element ref="tddnett:SFVOTTDDNETT" minOccurs="1"  
maxOccurs="1"/>  
        <xsd:element ref="limits:SFVOTLIMITS" minOccurs="1"  
maxOccurs="1"/>  
        <xsd:element ref="tddsum:SFVOTTDDSUM" minOccurs="1"  
maxOccurs="1"/>  
        <xsd:element ref="temperature:TEMPERATURE" minOccurs="1"  
maxOccurs="1"/>  
        <xsd:element ref="billing:SFVOTBILLING" minOccurs="1"  
maxOccurs="1"/>  
        <xsd:element ref="reportclaim:SFVOTCLAIM" minOccurs="1"  
maxOccurs="1"/>  
        <xsd:element ref="claimsum:SFVOTCLAIMSUM" minOccurs="1"  
maxOccurs="1"/>  
        <xsd:element ref="reportconf:SFVOTCONFDATA" minOccurs="1"  
maxOccurs="1"/>  
        <xsd:element ref="expimp:SFVOTDTEXPIMP" minOccurs="1"  
maxOccurs="1"/>  
        <xsd:element ref="emobilling:SFVOTBILLINGEMO"  
minOccurs="1" maxOccurs="1"/>  
        <xsd:element ref="billingsum:SFVOTBILLINGSUM"  
minOccurs="1" maxOccurs="1"/>  
        <xsd:element ref="reporttdd:SFVOTTDD" minOccurs="1"  
maxOccurs="1"/>  
        <xsd:element ref="exchrates:SFVOTEXCHRATE" minOccurs="1"  
maxOccurs="1"/>  
        <xsd:element ref="psk:SFVOTPSK" minOccurs="1"  
maxOccurs="1"/>
```

```
        <xsd:element ref="limitchange:SFVOTLIMITCHANGE"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="ack:AcknowledgementDocument"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="confirm:ConfirmationReport"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="anomaly:AnomalyReport" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="capacity:CapacityDocument"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="resresponse:RESRESPONSE" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="resclaim:RESCLAIM" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="resdata:RESDATA" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="ressource:RESSOURCE" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="settlidoc:RESSETTLDOC" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="fuel:RESFUEL" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="heat:RESHEAT" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="delegate:RESDELEGATE" minOccurs="1"
maxOccurs="1"/>
    </xsd:choice>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
```

3.1.5 CommonMarketService

The service includes XML structures <http://www.ote-cr.cz/schema/service/common/market>. The service contains operations with message-code 923 and 994. It contains following structures:

- COMMONMARKETREQ.xsd
- ISOTEDATA.xsd
- ISOTEMASTERDATA.xsd
- RESPONSE.xsd

Operation **Send** is defined on the service with following structure:

Input:

```
<xsd:element name="sendRequest">
  <xsd:complexType>
    <xsd:choice>
```

```
        <xsd:element ref="request:COMMONMARKETREQ" minOccurs="1"
maxOccurs="1"/>
    </xsd:choice>
</xsd:complexType>
</xsd:element>
```

Output:

```
<xsd:element name="SendResponse">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="globals:RETURN_CODE" minOccurs="1"
maxOccurs="1"/>
            <xsd:element ref="response:RESPONSE" minOccurs="0"
maxOccurs="1"/>
            <xsd:choice minOccurs="0" maxOccurs="1">
                <xsd:element ref="otedata:ISOTEDATA" minOccurs="1"
maxOccurs="1"/>
                <xsd:element ref="otemasterdata:ISOTEMASTERDATA"
minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

3.2 Services for communication OTE -> Participant

Three following callback services are designed for the support of asynchronous communication. They have to be implemented in a target system, which the communication server will communicate with. They represent counterparts to presented functions. Each service has a dual **Send** operation. Returning structure contains one single RETURN_CODE element indicating operation call score.

3.2.1 CDSCallbackService

The service includes asynchronous receiving of XML structures from <http://www.ote-cr.cz/schema/service/callback/cds> namespace, containing following structures:

- CDSCLAIM.xsd
- CDSDATA.xsd
- CDSIDIS.xsd
- TDD.xsd
- MASTERDATA.xsd
- TEMPERATURE.xsd
- CDSINVOICE.xsd
- CDSSYSTSERVREPORT.xsd
- CDSLDSREPORT.xsd

- RESPONSE.xsd

Operation **receive** is defined on the service with following structure:

Input:

```
<xsd:element name="SendRequest">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="response:RESPONSE" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="invoice:CDSINVOICE" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="claim:CDSCLAIM" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="data:CDSDATA" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="idis:CDSIDIS" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="masterdata:MASTERDATA" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="tdd:TDD" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="temperature:TEMPERATURE" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="sysstserv:CDSSYSTSERVREPORT" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="lds:CDSLDSREPORT" minOccurs="1"
maxOccurs="1"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

Output:

```
<xsd:element name="SendResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="globals:RETURN_CODE" minOccurs="1"
maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

3.2.2 MarketCallbackService

The service includes asynchronous receiving of XML structures from <http://www.ote-cr.cz/schema/service/callback/market>... namespace, containing following structures:

- ISOTEDATA.xsd
- ISOTEMASTERDATA.xsd
- RESPONSE.xsd

Operation **Send** is defined on the service with following structure:

Input:

```
<xsd:element name="SendRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="response:RESPONSE"
minOccurs="1" maxOccurs="1"/>
      <xsd:choice minOccurs="0" maxOccurs="1">
        <xsd:element ref="otedata:ISOTEDATA"
minOccurs="1" maxOccurs="1"/>
        <xsd:element
ref="otemasterdata:ISOTEMASTERDATA" minOccurs="1" maxOccurs="1"/>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Output:

```
<xsd:element name="receiveResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="globals:RETURN_CODE" minOccurs="1"
maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

3.2.3 ReportCallbackService

The service includes asynchronous receiving of XML structures from <http://www.ote-cr.cz/schema/service/callback/report> namespace, containing following structures:

- SFVOTBILLING.xsd
- SFVOTCLAIM.xsd
- SFVOTCLAIMSUM.xsd
- SFVOTBILLINGEMO.xsd
- SFVOTBILLINGSUM.xsd
- SFVOTDTEXPIMP.xsd
- SFVOTCONFDATA.xsd
- SFVOTTDD.xsd
- SFVOTEXCHRATE.xsd
- SFVOTTDDSUM.xsd
- SFVOTTDDNETT.xsd
- SFVOTLIMITS.xsd
- RESPONSE.xsd

Operation **Send** is defined on the service with following structure:

Input:

```
<xsd:element name="SendRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="response:RESPONSE" minOccurs="0"
maxOccurs="1"/>
      <xsd:choice minOccurs="0" maxOccurs="1">
        <xsd:element ref="billing:SFVOTBILLING" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="reportclaim:SFVOTCLAIM" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="claimsum:SFVOTCLAIMSUM" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="reportconf:SFVOTCONFDATA" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="expimp:SFVOTDTEXPIMP" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="emobilling:SFVOTBILLINGEMO"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="billingsum:SFVOTBILLINGSUM"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="reporttdd:SFVOTTDD" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="tddnett:SFVOTTDDNETT" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="tddsum:SFVOTTDDSUM" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="exchrates:SFVOTEXCHRATE" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="limits:SFVOTLIMITS" minOccurs="1"
maxOccurs="1"/>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Output:

```
<xsd:element name="receiveResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="globals:RETURN_CODE" minOccurs="1"
maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

3.2.4 CommonCallbackService

The service includes asynchronous receiving of XML structures from <http://www.ote-cr.cz/schema/service/callback/common>.. namespace. The service is designed for realisation of message with the message-code 995. It contains following structures:

- RESPONSE.xsd

Operation **Send** is defined on the service with following structure:

Input:

```
<xsd:element name="receiveRequest">  
  <xsd:complexType>  
    <xsd:sequence>  
      <xsd:element ref="response:RESPONSE" minOccurs="1"  
maxOccurs="1"/>  
    </xsd:sequence>  
  </xsd:complexType>  
</xsd:element>
```

Output:

```
<xsd:element name="receiveResponse">  
  <xsd:complexType>  
    <xsd:sequence>  
      <xsd:element ref="globals:RETURN_CODE" minOccurs="1"  
maxOccurs="1"/>  
    </xsd:sequence>  
  </xsd:complexType>  
</xsd:element>
```

3.3 Services for ETSO standard support

For ETSO standard support, these services are designed, respecting ETSO message type.

3.3.1 CapacityService

The service serve for sending CapacityDocument message. “MCC for DM” is implemented. It contains **SendCapacityDocument** operation, with following parameters structure:

Input:

```
<xsd:element name="SendCapacityDocumentRequest">  
  <xsd:complexType>  
    <xsd:sequence>  
      <xsd:element ref="capacity:CapacityDocument"  
minOccurs="1" maxOccurs="1"/>  
    </xsd:sequence>  
  </xsd:complexType>  
</xsd:element>
```

Output:

```
<xsd:element name="SendCapacityDocumentResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="ack:AcknowledgementDocument"
minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

where CapacityDocument and AcknowledgementDocument structures being of ETSO standart structures

3.3.2 ScheduleService

The service serve for sending ScheduleMessage message. “RD load“ is implemented. It contains **SendScheduleMessage** operation, with following parameters structure:

Input:

```
<xsd:element name="SendScheduleMessageRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="schedule:ScheduleMessage"
minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Output:

```
<xsd:element name="SendScheduleMessageResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="ack:AcknowledgementDocument"
minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

where ScheduleMessage and AcknowledgementDocument structures being of ETSO standart structures

3.3.3 ScheduleCallbackService

This is a call-back service for asynchronous transfer of data initiated from CS OTE system it is registered in namespace <http://www.ote-cr.cz/schema/service/callback/etso/schedule-v1>

It defines operation **Send** in the following structure

Input:

```
<xsd:element name="SendRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="confirm:ConfirmationReport" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="anomaly:AnomalyReport" minOccurs="1"
maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Output:

```
<xsd:element name="SendResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="globals:RETURN_CODE" minOccurs="1"
maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Where ConfirmationReport and AnomalyReport are standard ETSO structures taken over to the CSOTE defined namespace.

3.3.4 StatusRequestService

The service serve for sending StatusRequest message. “Request for MCC” and “Request for RD“ is implemented. It contains **GetStatus** operation, with following parameters structure:

Input:

```
<xsd:element name="GetStatusReq">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="esr:StatusRequest" minOccurs="1"
maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Output

```
<xsd:element name="GetStatusResponse">
  <xsd:complexType>
    <xsd:sequence>
```

```
        <xsd:element ref="ack:AcknowledgementDocument" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="confirm:ConfirmationReport" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="anomaly:AnomalyReport" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="capacity:CapacityDocument" minOccurs="1"
maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
```

where StatusRequest, AnomalyReport, ConfirmationReport, CapacityDocument and AcknowledgementDocument structures being of ETSO standart structures

3.4 Services for EDI standard support

Following service is designed for support of EDI format data sending:

3.4.1 EDIService

The service serve for sending base64 encoded data, representing sned EDI message to system OTE. The service contains **SendData** operation, with following parameters structure:

Input:

```
<xsd:element name="SendDataRequest">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="DATA" type="xsd:base64Binary"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

Output:

```
<xsd:element name="SendDataResponse">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="globals:RETURN_CODE" minOccurs="1"
maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

where DATA structures is a base64 chain encoding the message and RETURN_CODE is a type representing the return code

3.4.2 EDICallbackService

This is service type „Serices for communication with OTE -> participant (call-back service)“

<http://www.ote-cr.cz/schema/service/callback/edi>

Used for base64 coded EDI messages sending

SendData operation with following structure is defined for this service

Input:

```
<xsd:element name="SendRequest">  
  <xsd:complexType>  
    <xsd:sequence>  
      <xsd:element name="DATA" type="xsd:base64Binary"/>  
    </xsd:sequence>  
  </xsd:complexType>  
</xsd:element>
```

Output:

```
<xsd:element name="SendResponse">  
  <xsd:complexType>  
    <xsd:sequence>  
      <xsd:element ref="globals:RETURN_CODE" minOccurs="1"  
maxOccurs="1"/>  
    </xsd:sequence>  
  </xsd:complexType>  
</xsd:element>
```

kde struktury DATA je base64 řetězec kódující zprávu a RETURN_CODE je typ představující návratový kód.

4 Services - Gas

Service design is based on structures defined in the document D1.4.2 XML Formats, published on the public OTE websites.

Complete services definition are published on public OTE website.

4.1 Services for communication Participant ->OTE

4.1.1 CDSGasService

The service implements functionality of the data transmission eventually the request for data. It is registered in the namespace <http://www.ote-cr.cz/schema/service/cdsgas> and involves the following structures:

- CDSGASMASTERDATA.xsd
- CDSGASPOF.xsd
- CDSGASCLAIM.xsd
- CDSGASREQ.xsd
- CDSGASINVOICE.xsd

On this service one Send operation with the following structure is defined:

Input:

```
<xsd:element name="SendRequest">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="gasmasterdata:CDSGASMASTERDATA"
minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="gaspof:CDSGASPOF" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="gasclaim:CDSGASCLAIM" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="gasrequest:CDSGASREQ" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="gasinvoice:CDSGASINVOICE" minOccurs="1"
maxOccurs="1"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

Output:

```
<xsd:element name="SendResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="globals:RETURN_CODE" minOccurs="1"
maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
```

```
</xsd:element>
```

The service enables the asynchronous sending of the request / data to the OTE

4.1.2 CDSEdigasService

The service implements functionality of the data transmission eventually the request for data in EDIGAS format. It is registered in the namespace <http://www.ote-cr.cz/schema/service/cdsgas/edigas> and involves the following structures:

- Allocation.xsd
- GasdatDocument.xsd
- Aperak.xsd
- Nomination.xsd
- NominationResponse.xsd
- + special structure CDSEDIGASREQ.xsd for inquiry
- + special structure SHPCDS for shipper code

On this service the SendAsync operation with the following structure is defined:

Input:

```
<xsd:element name="SendAsyncRequest">  
  <xsd:complexType>  
    <xsd:choice>  
      <xsd:element ref="alloc:Allocation" minOccurs="1"  
maxOccurs="1" />  
      <xsd:element ref="gas:GasdatDocument" minOccurs="1"  
maxOccurs="1" />  
      <xsd:element ref="req:CDSEDIGASREQ" minOccurs="1"  
maxOccurs="1" />  
      <xsd:element ref="shpcds:SHPCDS" minOccurs="1" maxOccurs="1"  
/>  
    </xsd:choice>  
  </xsd:complexType>  
</xsd:element>
```

Output:

```
<xsd:element name="SendAsyncResponse">  
  <xsd:complexType>  
    <xsd:sequence>  
      <xsd:element ref="globals:RETURN_CODE" minOccurs="1"  
maxOccurs="1" />  
    </xsd:sequence>  
  </xsd:complexType>  
</xsd:element>
```

The operation enables the asynchronous sending of the request / data. As a response the return_code is handed over in synchronous mode and the additional data in asynchronous model on the call-back service or via other described asynchronous methods.

For the synchronous messages the SendSync operation is defined with the following structure:

```
<xsd:element name="SendSyncRequest">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="nom:Nomination" minOccurs="1"
maxOccurs="1" />
      <xsd:element ref="nomres:NominationResponse" minOccurs="1"
maxOccurs="1"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

Output:

```
<xsd:element name="SendSyncResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="globals:RETURN_CODE" minOccurs="1"
maxOccurs="1" />
      <xsd:element ref="aper:Aperak" minOccurs="0" maxOccurs="1"
/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

The operation enables the synchronous sending of the data to OTE. As a response the return_code is handed over and when there's no error (RETURN_CODE != 0) the Aperak message is tranfered also.

4.1.3 CommonGasService

This service includes the XML structures from <http://www.ote-cr.cz/schema/service/cdsgas/common> The service includes msg codes GX1 and GX2 and following structures:

- COMMONGASREQ.xsd
- GASRESPONSE.xsd
- CDSGASMASTERDATA.xsd
- CDSGASPOF.xsd
- CDSGASCLAIM.xsd
- Aperak.xsd

- Nomination.xsd
- NominationResponse.xsd
- ImbalanceNotice.xsd
- Allocation.xsd
- GasdatDocument.xsd
- BalanceAction.xsd
- SFVOTGASBILLING.xsd
- SFVOTGASCLAIM.xsd
- SFVOTGASCLAIMSUM.xsd
- SFVOTGASBILLINGSUM.xsd
- SFVOTGASTDD.xsd
- SFVOTGASTDDNETT.xsd
- SFVOTGASTDDSUM.xsd
- SFVOTGASEXCHRATE.xsd
- CDSGASINVOICE.xsd

One operation Send is defined with the following structure:

Input:

```
<xsd:element name="SendRequest">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="request:COMMONGASREQ" minOccurs="1"
maxOccurs="1"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

Output:

```
<xsd:element name="SendResp">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="globals:RETURN_CODE" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="gasresponse:GASRESPONSE" minOccurs="0"
maxOccurs="1"/>
      <xsd:choice minOccurs="0" maxOccurs="1">
        <xsd:element ref="gasmasterdata:CDSGASMASTERDATA"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="gaspof:CDSGASPOF" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="gasclaim:CDSGASCLAIM" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="alloc:Allocation" minOccurs="1"
maxOccurs="1" />
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```
        <xsd:element ref="nom:Nomination" minOccurs="1"
maxOccurs="1" />
        <xsd:element ref="gas:GasdatDocument" minOccurs="1"
maxOccurs="1" />
        <xsd:element ref="not:ImbalanceNotice" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="nomres:NominationResponse"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="aper:Aperak" minOccurs="1"
maxOccurs="1" />
        <xsd:element ref="balact:BalanceAction" minOccurs="1"
maxOccurs="1" />
        <xsd:element ref="shpcds:SHPCDS" minOccurs="1"
maxOccurs="1" />
        <xsd:element ref="billing:SFBVOTGASBILLING" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="reportclaim:SFBVOTGASCLAIM"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="claimsum:SFBVOTGASCLAIMSUM"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="billingsum:SFBVOTGASBILLINGSUM"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="reporttdd:SFBVOTGASTDD" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="gastddnett:SFBVOTGASTDDNETT"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="gastddsum:SFBVOTGASTDDSUM" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="gasinvoice:CDSGASINVOICE" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="gasexchrates:SFBVOTGASEXCHRATE"
minOccurs="1" maxOccurs="1"/>
    </xsd:choice>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
```

4.1.4 ReportGasService

This service includes XML structures from <http://www.ote-cr.cz/schema/service/reportgas>:

- SFVOTGASREQ.xsd
- GASRESPONSE.xsd

One operation Send is defined with the following structure:

Input:

```
<xsd:element name="SendRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="request:SFVOTGASREQ" minOccurs="1"
maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Output:

```
<xsd:element name="SendResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="globals:RETURN_CODE" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="response:GASRESPONSE" minOccurs="0"
maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

4.2 Services for communication OTE->Participant (call-back services)

For the support of the asynchronous communications the following call-back services are defined. These services must be implemented on the participant's system. Every of the following services has the Send operations. The response structure includes only RETURN_CODE element indicating whether the service call was successful or not.

4.2.1 CDSGasCallbackService

This service covers the asynchronous transfer of XML messages in OTE standard. The service is registered in the namespace <http://www.ote-cr.cz/schema/service/callback/cdsgas> and includes the following structures:

- GASRESPONSE.xsd
- CDSGASMASTERDATA.xsd
- CDSGASPOF.xsd
- CDSGASCLAIM.xsd
- CDSGASINVOICE.xsd

On this service one operation Send with the following structure is defined:

Input:

```
<xsd:element name="SendRequest">
  <xsd:complexType>
    <xsd:choice>
```

d1.4.3-en-web-services-interface-v1.14.doc

© 2020 OTE, a.s.

```
        <xsd:element ref="gasmasterdata:CDSGASMASTERDATA"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="gaspof:CDSGASPOF" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="gasclaim:CDSGASCLAIM" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="gasresponse:GASRESPONSE" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="gasinvoice:CDSGASINVOICE" minOccurs="1"
maxOccurs="1"/>
        </xsd:choice>
    </xsd:complexType>
</xsd:element>
```

Output:

```
<xsd:element name="SendResponse">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="globals:RETURN_CODE" minOccurs="1"
maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

4.2.2 CDSEdigasCallbackService

This service covers the asynchronous transfer of XML structures in EDIGAS format. It is registered in the name space <http://www.ote-cr.cz/schema/service/callback/cdsgas/edigas> and includes the following structures:

- Aperak.xsd
- NominationResponse.xsd
- ImbalanceNotice.xsd
- Nomination.xsd
- Allocation.xsd
- GasdatDocument.xsd
- BalanceAction.xsd
- SHPCDS.xsd
- GASRESPONSE.xsd

On this service one operation Send with the following structure is defined:

Input:

```
<xsd:element name="SendRequest">
    <xsd:complexType>
        <xsd:choice>
```

```

        <xsd:element ref="aper:Aperak" minOccurs="1" maxOccurs="1"
/>
        <xsd:element ref="not:ImbalanceNotice" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="nom:Nomination" minOccurs="1"
maxOccurs="1" />
        <xsd:element ref="nomres:NominationResponse" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="alloc:Allocation" minOccurs="1"
maxOccurs="1" />
        <xsd:element ref="gas:GasdatDocument" minOccurs="1"
maxOccurs="1" />
        <xsd:element ref="balact:BalanceAction" minOccurs="1"
maxOccurs="1" />
        <xsd:element ref="shpcds:SHPCDS" minOccurs="1" maxOccurs="1"
/>
        <xsd:element ref="gasresponse:GASRESPONSE" minOccurs="1"
maxOccurs="1"/>
        </xsd:choice>
    </xsd:complexType>
</xsd:element>
```

Output:

```
<xsd:element name="SendResp">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="globals:RETURN_CODE" minOccurs="1"
maxOccurs="1" />
            <xsd:element ref="aper:Aperak" minOccurs="0" maxOccurs="1"
/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

4.2.3 ReportGasCallbackService

This service covers the asynchronous transfer of XML structures in EDIGAS format. It is registered in the name space <http://www.ote-cr.cz/schema/service/callback/reportgas> and includes the following structures:

- SFVOTGASBILLING.xsd
- SFVOTGASCLAIM.xsd
- SFVOTGASCLAIMSUM.xsd
- SFVOTGASBILLINGSUM.xsd
- SFVOTGASTDD.xsd
- SFVOTGASTDDNETT.xsd

- SFVOTGASTDDSUM.xsd
- GASRESPONSE.xsd
- SFVOTGASIMGNETT.xsd
- SFVOTGASEXCHRATE.xsd

On this service one operation Send with the following structure is defined:

Input:

```
<xsd:element name="SendRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="response:GASRESPONSE" minOccurs="0"
maxOccurs="1"/>
      <xsd:choice minOccurs="0" maxOccurs="1">
        <xsd:element ref="billing:SFVOTGASBILLING" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="reportclaim:SFVOTGASCLAIM"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="claimsum:SFVOTGASCLAIMSUM"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="billingsum:SFVOTGASBILLINGSUM"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="reporttdd:SFVOTGASTDD" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="gastddnett:SFVOTGASTDDNETT"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="gastddsum:SFVOTGASTDDSUM" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="gasimgnett:SFVOTGASIMGNETT"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="gasexchrate:SFVOTGASEXCHRATE"
minOccurs="1" maxOccurs="1"/>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Output:

```
<xsd:element name="SendResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="globals:RETURN_CODE" minOccurs="1"
maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

5 Services - OZE

Service design is based on structures defined in the document D1.4.2 XML Formats, published on the public OTE websites.

Complete services definition are published on public OTE website.

5.1 Services for communication Participant -> OTE

5.1.1 RESService

The service implements functionality of the data transmission eventually the request for data. It is registered in the namespace <http://www.ote-cr.cz/schema/service/res> and involves the following structures:

- RESREQ.xsd
- RESCLAIM.xsd
- RESDATA.xsd
- RESSOURCE.xsd
- RESFUEL.xsd
- RESHEAT.xsd

On this service one Send operation with the following structure is defined:

Input:

```
<xsd:element name="SendRequest">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="request:RESREQ" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="claim:RESCLAIM" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="data:RESDATA" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="source:RESSOURCE" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="fuel:RESFUEL" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="heat:RESHEAT" minOccurs="1"
maxOccurs="1"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

Output:

```
<xsd:element name="SendResponse">
  <xsd:complexType>
```

```
<xsd:sequence>
  <xsd:element ref="globals:RETURN_CODE" minOccurs="1"
maxOccurs="1"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
```

Umožňuje asynchronní zaslání požadavku/dat na OTE přičemž jako odpověď je zaslán synchronně chybový kód a asynchronně dodatečná data zaregistrovaným kanálem (na Callback službu, případně emailem).

5.2 Services for communication OTE->Participant (call-back services)

For the support of the asynchronous communications the following call-back services are defined. These services must be implemented on the participant's system. Every of the following services has the Send operations. The response structure includes only RETURN_CODE element indicating whether the service call was successful or not.

5.2.1 RESCallbackService

This service includes asynchronous Reception of XML structure in different than EDIGAS format. It is registered in the namespace <http://www.ote-cr.cz/schema/service/callback/cdsgas> and involves the following structures:

- RESRESPONSE.xsd
- RESCLAIM.xsd
- RESDATA.xsd
- RESSOURCE.xsd
- RESSETTLDOC.xsd
- RESFUEL.xsd
- RESHEAT.xsd
- RESDELEGATE.xsd

On this service one Send operation with the following structure is defined:

Input:

```
<xsd:element name="SendRequest">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="response:RESRESPONSE" minOccurs="1"
maxOccurs="1"/>
      <xsd:element ref="claim:RESCLAIM" minOccurs="1"
maxOccurs="1"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

d1.4.3-en-web-services-interface-v1.14.doc

© 2020 OTE, a.s.

```
        <xsd:element ref="data:RESDATA" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="source:RESSOURCE" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="settlldoc:RESSETTLDOC" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="fuel:RESFUEL" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="heat:RESHEAT" minOccurs="1"
maxOccurs="1"/>
        <xsd:element ref="delegate:RESDELEGATE" minOccurs="1"
maxOccurs="1"/>
    </xsd:choice>
</xsd:complexType>
</xsd:element>
```

Output:

```
<xsd:element name="SendResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="globals:RETURN_CODE" minOccurs="1"
maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

6 Example of signed document

The annex demonstrates a practical example of signed XML document, as it is specified in the chapter Message body. ISOTEDATA element contains as the last subelement {<http://www.w3.org/2000/09/xmlsig#>} Signature, representing digital signature of ISOTEDATA message.

```
<data:ISOTEDATA id="?" message-code="?" date-time="?" dtd-version="1"
dtd-release="1" answer-required="?" interval="?" err-reaction="?">
  <data:SenderIdentification id="?" coding-scheme="?"/>
  <data:ReceiverIdentification id="?" coding-scheme="?"/>
  <!--Optional:-->
  <data:DataProvider id="?" coding-scheme="?"/>
  <!--Optional:-->
  <data:Reference id="?"/>
  <!--1 or more repetitions:-->
  <data:Trade id="?" id-definition="?" trade-day="?" trade-
day-to="?" version="?" trade-state="?" trade-type="?" trade-order="?"
error-code="?" replacement="?" acceptance="?" market="?" trade-flag="?"
trade-stage="?" short-description="?" anulation-proposer="?" anulation-
type="?" sett-curr="?" source-sys="?" version-a="?" market-area="?">
    <!--Zero or more repetitions:-->
    <data:TimeData datetime="?" datetime-type="?"
timeattribute="?"/>
    <!--Zero or more repetitions:-->
    <data:ProfileData profile-role="?">
      <!--1 or more repetitions:-->
      <data:Data period="?" value="?" unit="?" splitting="?"
perflag-cancel="?" perflag-aggregation="?" timeattribute="?" anulation-
proposer="?" period-stage="?">
        <!--Zero or more repetitions:-->
        <data:ProcReason code="?"/>
      </data:Data>
    </data:ProfileData>
  <!--Optional:-->
  <data:Comment>?</data:Comment>
  <!--0 to 4 repetitions:-->
  <data:Party id="?" role="?"/>
  <!--Zero or more repetitions:-->
  <data:ProcReason code="?"/>
</data:Trade>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmlsig#">
  <ds:SignedInfo
xmlns:ds="http://www.w3.org/2000/09/xmlsig#">
    <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"
xmlns:ds="http://www.w3.org/2000/09/xmlsig#">
      <ds:SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmlsig#rsa-sha1"
xmlns:ds="http://www.w3.org/2000/09/xmlsig#">
```



```
<ds:Reference URI=""
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:Transforms
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" />
  </ds:Transforms>
  <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" />
    <ds:DigestValue
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">hSyyBsLmZbORDXG/qFObVMDl00
U=
    </ds:DigestValue>
  </ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
g82dQcKq/aM5CFoerU0Ee5Qf4ImvF8UO9cb+nAiq3Gic9nhgxG1/Y6y5QTiqPUkN8KHiu6VB
o5M8
WmFxVKv7qev8Ru4meIe+66AqQ7jpsWfwYku2G2TnyagwZNubUuyK34o5siTg+I7ERmpENZx6
to/y
8Yr+blhf4hUCdwo4b20=
  </ds:SignatureValue>
<ds:KeyInfo
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:X509Data
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:X509Certificate
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
MIIDRTCCAi2gAwIBAgIKKL+faQAFAAASTDANBgkqhkiG9w0BAQUFADBGMQswCQYDVQQGEwJD
WjEP
MA0GA1UEChMGTG9naWNhMRIwEAYDVQQLEwlQS0kgR3JvdXAxZDjAMBgnVBAAMTBURUNBMB4X
DTA4
MDYzMDU1NDcwMFoXDSAwMDYzMDU1NTcwMFowQDELMAkGA1UEBhMCQ1oxDDAKBgNVBAoTA09U
RTEQ
MA4GA1UECXMHUGVyc29uczERMA8GA1UEAxMIZWludm9pY2UwgZ8wDQYJKoZIhvcNAQEBBQAD
gY0A
MIGJAoGBAM1ltRACfYcAYEay3wEs1tqgr6QMuPm2yOz7pfrLrrfSBH77jSZ82UUePHE8QfPS
P9QJ
MH1hZKy2yN1IbR3mQo7PwBsBwbPiBU4ynVSMSP5YCNywuVcUdEejS4bdTV7VQ+a6Kw1DlkqP
8d17
OO8J08kT/ys8a2WhDwr4pRi9itLtAgMBAAGjgcIwgb8wDgYDVR0PAQH/BAQDAgTwMBMGA1Ud
JQQM
MAoGCCsGAQUFBwMCMB0GA1UdDgQWBRRhP/9LuhLsaEmlBXdmB34BY8ZpdDB5BgNVHSMEcjBw
gBT0
8Vf3iDofXlR27DvrxE+H5diP0aFGpEQwQjELMAkGA1UEBhMCQ1oxDzANBgNVBAoTBkxvZ21j
YTES
MBAGA1UECXMJUeEtJIEdyb3VwMQ4wDAYDVQQDEwVPEVVDQYIQeYNCXj6Lf6NNexTEztfDAN
Bgkq
```

```
hkiG9w0BAQUFAAOCAQEAKikZTqWOBvVh5/Rv1+1Cad55CH+9zbBrI4v7BdWKHW3KBUV+Z7jD
FTEH
CQLiWjAjJ8bpfmb2r3UH9pG/ojzJDNxfAc8/eX62iDrDO6IdKCknca7c6ZCkCEK716KHJQPi
1eCD
Cw8mQA1BN0/mv1BOE75mYmqIs/1oIPpMelqgLXk2mbOvRKeEQWdDFL1LG0dJCjC/OIhbguN2
n4RD
Y6XTAzsSJBvRu3PKfA+8JG+0SKx0sAaOZjYtUv4fZ1gcgLkDwXuJUmq1cxbMs4X32BHXGtPf
h8C7
DB8LCXwlqlFFoj4JJdBGWxYOjIpiawXD/6VFodo3V+Wj6uXGCdMYNrTsNw==
      </ds:X509Certificate>
    </ds:X509Data>
  <ds:KeyValue
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:RSAKeyValue
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:Modulus
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
zWW1EAJ9gIBgRrLfASzW2qCvpAy4+bbI7Pul+suut9IEfvuNjnzZRR48cTxB89I/1AkwfWfk
rLbI
3UhtHeZCjs/AGwHBs+IFTjKdVIxKnlgI1jC5VxR0R6NLht1NXtVD5rorDUOWSo/x3Xs47wnT
yRP/
KzxrZaEPCvilGL2K0u0=
      </ds:Modulus>
    <ds:Exponent
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">AQAB
    </ds:Exponent>
  </ds:RSAKeyValue>
</ds:KeyValue>
</ds:KeyInfo>
</ds:Signature>
</data:ISOTEDATA>
```