Fiscalization Service

(Version v03)

Technical Specification

Versions

Version	Description of Change
v01	Initial Version
v02	Shortened filed names.
	Added additional error codes.
	Added WTNIC code generation chapters.
	Added test and production URL-s.
	Terminology which is used in the Law inserted into the introduction chapter
v03	Changed endpoint URL-s.
	QR code generation modifications, terminology changed
	Register invoice chapter changes:
	• Check date and time sent control moved from mandatory to optional controls
	Register invoice request data message changes:
	• Issuer element reamed to Seller (in below changes renamed element will be referred with a new name)
	• Seller NUIS attribute replaced with IDNum and IDType attributes
	• Seller Address, Town and Country attributes changed to optional
	• Buyer NUIS attribute replaced with IDNum and IDType attributes
	• CashRegister attribute renamed to TCRCode
	• TCRCode attribute changed to optional
	• Fees list element added
	• PayDeadline attribute added
	• PaymentMethod attribute replaced with PayMethods element list
	• BusinUnit attribute renamed to BusinUnitCode
	• SoftNum attribute renamed to SoftCode
	• Currency element added, Currency Code and ExRate attributes added
	• SupplyDateOrPeriod element added, SupplyDateOrPeriod Start and End attributes added
	• SameTaxItems element renamed to SameTaxes
	• SameTaxes Items element renamed to SameTax
	• ConsTaxItems element renamed to ConsTaxes
	• ConsTaxes Items element renamed to ConsTax
	• SameTaxes element changed to optional
	• Simplified invoice TypeOfInv type added
	• TypeOfInv enumerations changed
	• TypeOfSelfIss enumerations changed
	• SelfIssuing attribute removed
	• SumCompCard attribute added
	• GoodsExport attribute renamed to GoodsExAmt
	• RR changed to mandatory
	• InvNum attribute removed
	• InvOrdNum attribute renamed to InvNum
	• BadDept attribute renamed to IsBadDept
	• IsBadDept attribute changed to mandatory
	• ReverseCharge attribute renamed to IsReverseCharge
	• IssuerInVAT attribute renamed to IsIssuerInVAT
	• Item VR, VA, R, RR changed to optional
	• Item EX attribute added
	• SameTax ExemptFromVAT attribute added
	• Renamed UP attribute to UPB
	• Added UPA attribute
	• Buyer attributes changed to optional

Version	Description of Change
	• IsSubsequentDelivery moved to the Header element
	• Added check is issuer active and VAT obligated
	Register TCR chapter changes:
	• Renamed to Register TCR
	• Added additional description.
	• Added general mandatory controls
	• Added send time mandatory control
	Register TCR request data message changes:
	• RegDateTime attribute removed
	• BusinUnit attribute renamed to BusinUnitCode
	• TCROrdNum attribute renamed to TCRIntID
	• TCRIntID data type changed to string
	• SoftNum attribute renamed to SoftCode
	• ManufacNum attribute renamed to MaintainerCode
	• Added ValidFrom and ValidTo attributes
	• SoftCode and MaintainerCode attributes changed to optional
	Register TCR response data message changes:
	• TCRNumber element renamed to TCRCode
	• TCRCode data type pattern changed
	Register TCR cash balance chapter changes:
	• Renamed to Register cash deposit
	• Added additional description.
	• Added general mandatory controls
	• Added send time mandatory control
	Register cash deposit request data message changes:
	• TCRNumber attribute renamed to TCRCode
	• TCRCode data type pattern changed
	• Operation enumeration changed to INITIAL, INOUT
	• IsSubsequentDelivery moved to the Header element
	Register cash deposit response data message changes:
	• FCDC element added
	Chapter PKI renamed to Digital certificates and provided additional description.
	Error message changes:
	• RequestUUID element changed to optional
	• ResponseUUID element added
	Added deregistration TCR method
	Controls:
	• Common mandatory controls added to separate chapter.
	• Optional controls renamed to additional controls
	• Removed list of additional controls
	Warehouse transfer note chapter content removed.
	Updated XML samples.
	Added additional error codes.
	Added .NET C# sample code for signature generation.
	Added Id and Version attributes.
	Modified schema patterns to take into account eager validation.

Related Documents

Document Name	Description
Fiscalization Service - Functional specification	All processes described.

CONTENTS

1. Introduction

This document provides a description of the data interface for invoice and acknowledgement of data messages containing information on sales which the taxpayers are obliged to send for every sale made and subject to registration of Sales, i.e. invoices issued.

Files containing definition of the XML schema and the Web service (WSDL), which describe the structure of the registered invoice data messages and the Web service used to receive them are provided as Annexes 6 and 7 to this document.

This document provides specification for the fiscalization service version 2.

Old deprecated service version will be available for 4 weeks after the new version is released. Date when a new version is released can be found at GDT website https://www.tatime.gov.al/eng/c/320/327/fiscalization-service-technical-specifications.

1.1 USED ABBREVIATIONS

Abbreviation	Description	Terminology used in the Law (if it is different)
CA	Certificate Authority	-
CIS	Central Invoice System	-
CPCM	CPCM is central portal for the control and management of taxpayers in the cash transaction subsystem	-
CRL	Certificate Revocation List	-
FIC	Fiscal Identification Code (generated at server side after successful verification of the invoice)	UII – Unique invoice identifier
FNIC	Fiscal Note Identification Code (generated at server side after successful verification of the warehouse transfer note)	UWTNI - Unique Warehouse Transfer Note Identifier
GUID	Global Unique Identifier	-
IIC	Issuer's invoice code	ISC - Invoice Issuer's Security Code
NIC	Note Identification Code (warehouse transfer note identification code)	WTNISC - Warehouse Transfer Note Issuer's Security Code
NUIS	National Unique Identification Number	NUIS/NIPT
OCSP	On-Line Certificate Status Protocol	-
SOAP	Message exchange protocol for XML messages as specified at: https://www.w3.org/TR/soap/	-
TCR	Taxpayer Cash Register. The same as billing device or electronic cash device.	Taxpayer's electronic cash device
TCRCode	Taxpayer Cash Register Code.	Taxpayer's electronic cash device code
UC	Use case	-
UUID	Universally Unique Identifier	-
WSDL	Web Services Description Language –XML-based language for description of functions offered by a WWW service as specified at http://www.w3.org/TR/wsdl	-

Abbreviation	Description	Terminology used in the Law (if it is different)
XML Schema	A XML-based language intended for definition of XML document structure as specified at http://www.w3.org/TR/xmlschema11-1/ and https://www.w3.org/TR/xmlschema11-2/	-

Table 1 – Used abbreviations

1.2 TERMINOLOGY

Term	Definition	Terminology used in the Law (if it is different)
Response data message	A data structure in a defined format prescribed by the financial authority, which contains the Fiscal Identification Code (FIC) and is used as acknowledgement of invoice and formal correctness of the registered invoice data message sent.	A data structure in a defined format prescribed by the financial authority, which contains Unique invoice identifier (UII) and is used as acknowledgement of invoice and formal correctness of the registered invoice data message sent.
Error Data Message	A data structure in a defined format prescribed by the financial authority, which contains an error code and its text description as a reaction to a registered invoice data message received containing critical errors preventing it from being processed, or when another error occurs which prevents the message being processed at the tax authority's side.	-
Invoice	An invoice is a proof of sale issued (in paper form or electronically) by a taxpayer to a person or entity making a purchase, which contains all information regarding totals of the sale and items. Invoice shall mean any document issued in paper or in electronic form, which satisfies the requirements provided under the draft Law "ON INVOICES AND THE SYSTEM FOR MONITORING TRANSACTIONS"	-
Issuer	Person who is issuing the invoice. Issuer of the invoice is responsible for the fiscalization of the invoice in CIS. This person is in most cases the seller of goods and services but in case of self-billing invoice, the issuer is the buyer of goods and services.	-
Registered Invoice	Invoice which is registered on CIS containing FIC.	Invoice which is registered on CIS containing UII.
Registered invoice data message	A data structure in a defined format prescribed by the fiscal authority, which contains information about the sale and other technical information necessary. This is a complete XML message containing information described in the relevant Web service standards: SOAP/WSDL/WS-Security, etc. A registered invoice data message is sent by an electronic cash device to the tax authority's common technical equipment (Central invoice system)..	-
Central invoice platform	Central invoice platform is a web application for taxpayers providing support for invoice fiscalization processes.	-
Taxpayer's cash register/electronic cash device	Taxpayer´s cash register or electronic cash device is a device on the taxpayer's side, which sends information on registered invoices to the tax authority (to the Central	-

Term	Definition	Terminology used in the Law (if it is different)
	Invoice System). This may signify, depending on the context, an end device such as a cash register, or additional SW and HW actually sending the registered invoices information. The data messages include an item marked as "Electronic cash device code", which identifies the end device (electronic cash device). In other parts of the text, this term usually means the end device and the relevant SW and HW sending the data messages.	

Table 2 - Terminology

2. Environments

The government will publish Web service addresses for two types of environments: production environment and one or more test environments:

- **Non-production environment** will be used solely by software developers (developing software for cash registers), not by cash registers' end users. Sending a data message to the non-production environment shall not be considered sending of registered invoice information. The FIC returned by the non-production environment is not a valid FIC (it is different per prefix). In the non-production environment, digital certificates for cash registers may be issued using a simplified process.
- **Production environment** is intended for the taxpayers and will be used for routine operations, i.e. receipt and acknowledgement of data messages containing information on registered sales.

Endpoints:

- o Test environment:
    - o https://eFiskalizimi-test.tatime.gov.al/FiscalizationService-v2/FiscalizationService.wsdl
- o Production environment:
    - o https://eFiskalizimi.tatime.gov.al/FiscalizationService-v2/FiscalizationService.wsdl

2.1 PREPARATION WORKS FOR FISCALIZATION SERVICE USE

Details on this matter can be found in Fiscalization Service - Functional specification, chapter that covers this subject. Below is the process diagram.
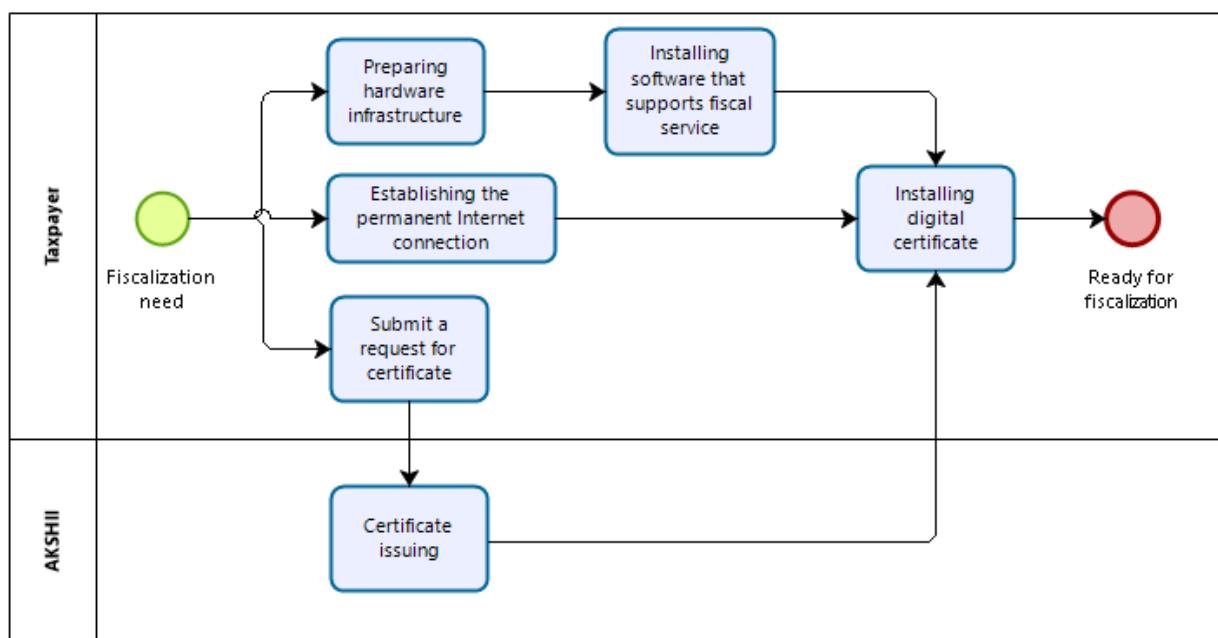
Figure 1 – Preparation activities for using a fiscal service

2.2   TOPOLOGY

Users access the CIS by initiating 1-way TLS connection. Clients exchange messages with Tax administration's access point using TLS channel by described procedure. Data exchange is synchronous, meaning access point answers on user's request immediately. Request and response messages formats are specified through XML schema.

2.2.1   CIS ACCESS POINT

Implementation and maintenance of the access point is a TBD's task. TBD company will provide its users connection to the access point in two environments: production and test.

2.2.2   INTERNET CONNECTION

Access point will be available through internet networks in HTTPS protocol.

2.2.3   INFORMATION SYSTEM OF THE CLIENT

Clients are obliged to provide hardware and software support for messages exchange with access point. As shown on image below, there is no mediatory component development planned. Development of the hardware-software solutions is in client's domain of business. Client is also obliged to secure internet connection to CIS access point with needed bandwidth. Platform choice and software solution implementation is in client's domain and such information is not needed to be reported to TBD company.
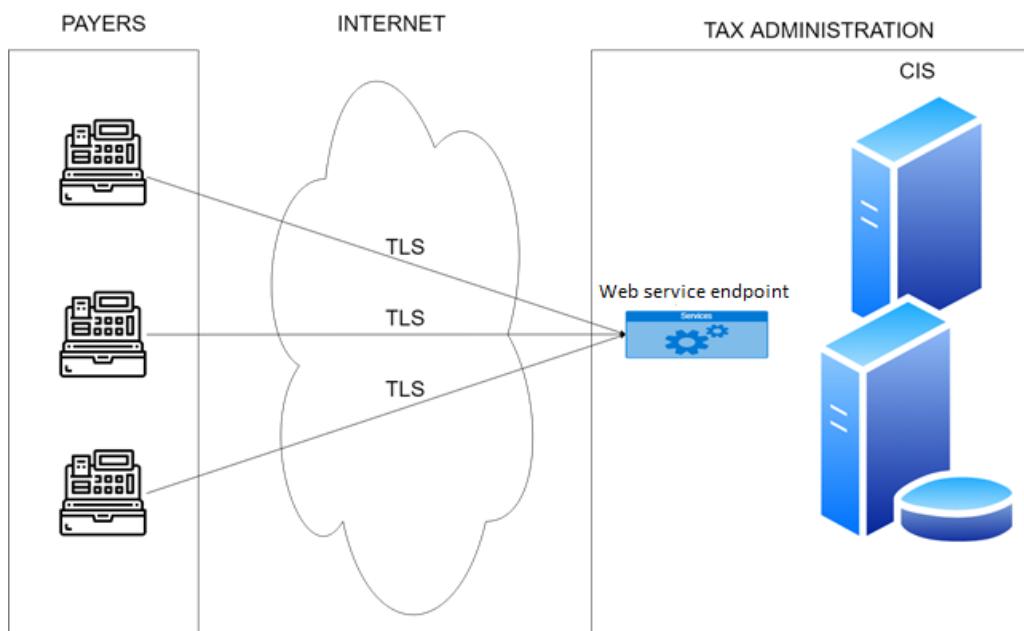
Figure 2 – Information system of the client

2.3   CONDITIONS FOR CONNECTION TO CIS

Central information system (CIS) of the Tax Administration will be available in two environments: production and test.

Connection conditions are similar but differ in addresses of their access points and certificates. Both, in production and test environment certificates are issued by NAIS. Environments are not different in its functionality (besides new functionalities development), only difference is in data – test environment uses test data.

2.3.1 NETWORK PRECONDITIONS AND RECOMMENDATIONS

To connect to the CIS of the Tax Administration, client system needs to fulfil these conditions:

Network type	Internet
Recommended open TCP ports to CIS	443

Network recommendations for client system are:

Link characteristics	Permanent symmetric link
Bandwidth	2 Mb/s at minimum (up to 40 messages per second with assumption that message takes 6 KB)

2.3.2 SECURITY PRECONDITIONS

All communication with CIS of the Tax Administration is protected by 1-way TLS encryption at the transport layer. In production environment CIS presents itself to client with a TLS certificate issued by NAIS production CA, while in test environment the certificate is issued by NAIS test CA.

Protection at the transport layer	HTTPS (TLS v1.1 and v1.2, AES_256 encryption at least)
Certificates for the electronic signing	Certificate type: application digital certificate for fiscalization

2.3.3 APLICATION PRECONDITIONS

CIS functionality is available to its clients using web-service technology. For that reason client´s application (or infrastructure, depending on realization) needs to fulfil these preconditions:

Client creation standards	WS-1
Service type	Document-literal
Application protocol	SOAP/HTTPS (SOAP 1.1)
Code site of the request message XML	UTF-8

3. Interface

Interface for exchanging the data between the taxpayer and CIS regarding the fiscalization will be SOAP web service. Messages are in XML format according to the standards of SOAP messages.

The web service has several operations which will be used by the taxpayer who needs to do the fiscalization of the invoices. Invoices are issued by the electronic cash device represented by its code (for cash transactions). The code is assigned in operation of registration of electronic cash device which needs to be executed during the installation of each electronic cash device. At the beginning of each day, electronic cash device which handles cash transactions must register the amount of cash in the deposit and only then it should start to issue invoices. Each invoice must be registered to the fiscalization service and upon successful registration the invoice is assigned with a FIC which is printed on the invoice together with other mandatory invoice elements. In case that the invoice needs to be corrected, new corrective invoice is issued with a reference to the invoice which needs to be corrected. During the day, electronic cash devices for cash payments should have the possibility for the taxpayer to see the current cash balance (it is recommended to do it when the operators of electronic cash device changes) as well as notify the Tax Administration through fiscalization service of any withdrawal or extra deposit of cash in the cash register. Each of this operation is explained in its chapter together with the list of elements of the exchanged messages.

Taxpayers should also register warehouse transfer notes for all goods transferred between warehouses and sale premises inside territory of Republic of Albania in situations specified by the Law "on invoice and system for monitoring transactions" and its bylaws.

Message sent by the taxpayer to CIS is the request message to which CIS replies by sending the response message. In case of an error, the error message is sent in the response with its structure. Request and response messages (except for the error message) all have the following parts: header (general info about the message), data (data specific for the operation), signature (digital signature signed by the person who is sending the message which provides the identity of the sender and info to verify that the data of the message is not changed). Signature is explained in chapter 4.3.

3.1 INTERFACE VERSIONING

Versioning of the fiscalization service will be based on semantic versioning schema. Each version has a version number assigned expressed as "MAJOR.MINOR.PATCH" each of which are integers incremented according to these rules:

- MAJOR version is increased when there are incompatible API changes. New interface will be provided, and old interface will remain for some period. Clients are expected to upgrade to new version as described in release notes of the new version.
- MINOR version is increased when thefunctionality is added in a backwards-compatible manner. Current interface remains compatible with current clients, but new functionalities are added which can or should be used. Clients are expected to upgrade to new version as described in release notes of the new version.
- PATCH version is increased when there are backwards-compatible bug fixes. Current interface remains the same.

Service endpoint will have a context suffix -vMAJOR, e.g. /FiscalizationService-v2. This means that at one moment there might be several active service endpoints with different MAJOR versions but each of them will always have the latest MINOR and PATCH versions.

3.2 DATA MESSAGE CODING

All items in all data messages will only use selected characters encoded as a single byte in a standard decimal ASCII character set. The allowed decimal codes are 9, 10, 13, or 32 to 126.

UTF-8 must be used for encoding the data messages as XML documents, i.e. first line of the XML SOAP envelope will always be:

<?xml version="1.0" encoding="UTF-8"?>

All XML elements of the fiscalization service are part of the same namespace, referenced in the Web service definition (WSDL).

The data format mask for individual items, which is listed along with their detailed description below, is a regular expression in the sense of the XML Schema, which defines the required syntax of the given item.

3.3 DATA MESSAGE STRUCTURE

All types of data messages have a common basic data format based on the SOAP 1.1 (Simple Object Access Protocol) protocol, i.e. application XML data structures are inserted into the body of the SOAP envelope. Unlike SOAP envelope header which remains empty.

Every request and response data message shall be signed with a private key belonging to the issuer or fiscalization service respectably. Exception to that rule are error messages (described in the chapter 3.7.6) which are not signed by the fiscalization service.

Digital signature is calculated only for the data message that resides inside SOAP envelope body element and is incorporated inside that data message as a envelop signature XML element.

3.4 CONTROLS

There are three types of controls:

- Mandatory controls (in real time)
- Additional controls (during back-up verification)

Mandatory controls shall be performed by CIP system in all methods in real time. In case that the control is not passed, an error message will be returned with error code defined here. The mandatory controls include the following:

Control Name	Control Description	Error code
XML format	XML format must be valid	0, 20
Check size of the data	Size should not exceed 150kB	1
XML structure validation	Check of the individual registered invoice data message's in XML against the XSD schema (*.xsd). XSD schema contains an exact definition of the data and format structure for the individual data items and a check of presence of individual items	11
Certificate validation	Check that certificate is not expired. Check that the certificate is issued by the trusting CA. Check that the identification number in the certificate corresponds to the invoice issuer identification number (tax number) in the XML message. Check that certificate is not listed in CRL.	34, 35, 36, 37, 38, 39

Electronic signature check	Check that the hash of the message calculated by CIS corresponds to the hash listed in the message.	22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
	Check that the signature corresponds to the hash of the message and to the public key of the certificate.	
Client time differs	Client time differs from a server's time by more than allowed time in minutes	2
Other controls	Other controls as specified in the functional specification and for control purposes of the Tax Administration	

Table 3

Additional controls are not performed at the moment of registration of the invoice but are instead postponed for later processing of the invoices. Errors detected here will be available to taxpayer over central invoice platform and to tax officials through CPCM.

3.5 REGISTER TCR

Each electronic cash device should be registered on CIS in order to receive the code which represents that electronic cash device. This code is used for identification of electronic cash device in messages which are exchanged between CIS and the electronic cash device. This registration should be done only once when the electronic cash device is installed in the business premise where it is used.

Before this, taxpayer needs to be registered in Tax Administration in active Registry of taxpayers. Taxpayer must also register the business premise (in application Central invoice platform) in which the TCR is located prior to registration of the TCR.

If this registration is done for the same internal identification of the TCR and business unit code, CIS will update information with new data provided in the request message and return same TCRCode. Deregistration is possible by providing new ValidTo field date field in RegisterTCRRequest message.
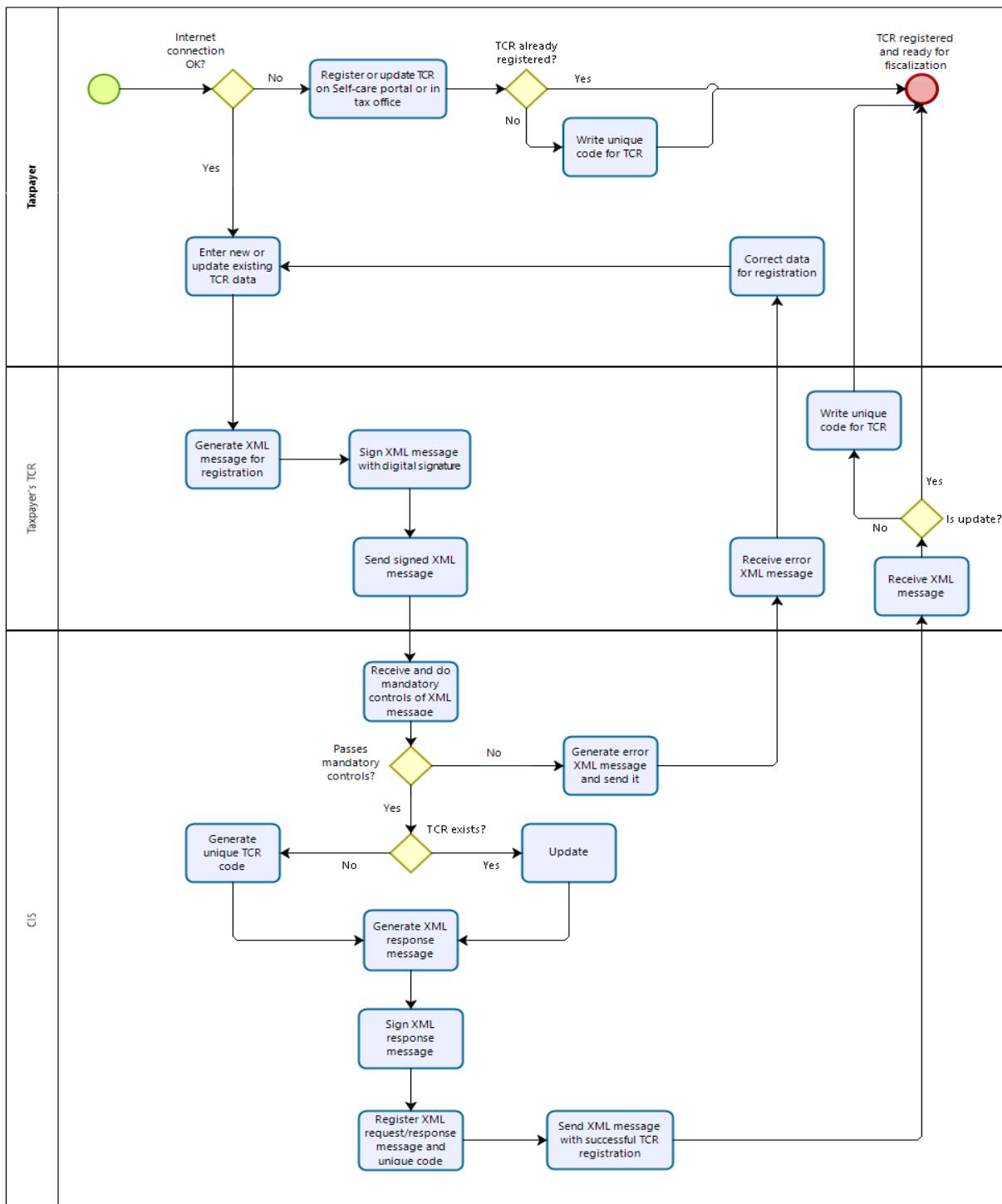
Figure 3 - Registration of taxpayer's cash register

3.5.1 REGISTER TCR REQUEST DATA MESSAGE

Name	Field type	Occurrence [Min, Max]	Description
RegisterTCRRequest	Element	[1, 1]	Root XML element representing registration of TCR.
Id	Attribute	[1, 1]	Attribute used for signature creation and verification. Fixed value "Request".
Version	Attribute	[1,1]	Attribute used to specify compliance with XSD schema. For this version fixed value is "2".
Header	Element	[1, 1]	XML element representing header…
UUID	Attribute	[1, 1]	ID of the message.
SendDateTime	Attribute	[1, 1]	Date and time of sending the message to the Tax administration.
TCR	Element	[1, 1]	XML element representing a single TCR registration message.
IssuerNUIS	Attribute	[1, 1]	Taxpayer's NUIS/NIPT.
BusinUnitCode	Attribute	[1, 1]	Business unit (premise) code.
TCRIntID	Attribute	[1, 1]	TCR internal identification.
SoftCode	Attribute	[0, 1]	Code of the software used by TCR.
MaintainerCode	Attribute	[0, 1]	Code of the maintainer for the TCR software.
ValidFrom	Attribute	[0, 1]	Date from which the TCR will be used.
ValidTo	Attribute	[0,1]	Date until which the TCR will be used.
Signature	Element	[1, 1]	XML element with digital signature.

Table 4

3.5.1.1 Header

Element representing the header of the request data message.

3.5.1.2 Header UUID

Element generated by the TCR. It uniquely identifies the request message sent from TCR to CIS. UUID should be constructed according to the RFC4122 version 4.

Data type	string
Length	36 characters
Pattern	[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[1-5][0-9a-fA-F]{3}-[89abAB][0-9a-fA-F]{3}-[0-9a-fA-F]{12}
Example	58e0a7d7-eebc-41d8-9669-0800200c9a66 58E0A7D7-EEBC-41D8-9669-0800200C9A66

Table 5

3.5.1.3 Header SendDateTime

Element represents date and time of sending the request message to the CIS. Date and time should be in ISO 8601 format.

Data type	dateTime
Length	23 characters
Pattern	[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}[+-][0-9]{2}:[0-9]{2}
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 6

3.5.1.4 TCR

Element representing a single TCR registration request.

3.5.1.5 TCR IssuerNUIS

Element representing issuer's NUIS/NIPT.

Data type	string
Length	10
Pattern	[a-zA-Z]{1}[0-9]{8}[a-zA-Z]{1}
Example	K72001008V

Table 7

3.5.1.6 TCR BusinUnitCode

Code of the business unit (premise) in which the invoice is issued.

Data type	string
Length	10 characters
Pattern	[a-z]{2}[0-9]{3}[a-z]{2}[0-9]{3}
Example	ab123ab123

Table 8

3.5.1.7 TCR TCRIntID

Attribute representing the internal identification of the TCR.

Data type	string
Length	50
Example	2

Table 9

3.5.1.8 TCR SoftCode

Code of the software used for invoice issuing.

Data type	string
Length	10 characters
Pattern	[a-z]{2}[0-9]{3}[a-z]{2}[0-9]{3}
Example	ab123ab123

Table 10

3.5.1.9 TCR MaintainerCode

Code of the maintainer for the TCR software.

Data type	string

Length	10 characters
Pattern	[a-z]{2}[0-9]{3}[a-z]{2}[0-9]{3}
Example	ab123ab123

Table 11

3.5.1.10   TCR ValidFrom

Element represents date from which the TCR will be used or is valid.

Data type	Date
Length	10 characters
Pattern	[0-9]{4}-[0-9]{2}-[0-9]{2}
Example	2019-01-24

Table 12

3.5.1.11   TCR ValidTo

Element represents date until which the TCR will be used or is valid.

Data type	date
Length	10 characters
Pattern	[0-9]{4}-[0-9]{2}-[0-9]{2}
Example	2019-01-24

Table 13

3.5.1.12   Signature

XML element stores enveloped digital signature described in the chapter 4.3.1.

3.5.2   REGISTER TCR RESPONSE DATA MESSAGE

Name			Field type	Occurrence [Min, Max]	Description
RegisterTCRResponse			Element	[1, 1]	Root XML element representing registration of TCR.
	Id		Attribute	[1, 1]	Attribute used for signature creation and verification. Fixed value "Response".
	Version		Attribute	[1,1]	Attribute used to specify compliance with XSD schema. For this version fixed value is "2".
	Header		Element	[1, 1]	XML element representing header…
		UUID	Attribute	[1, 1]	ID of the message.
		RequestUUID	Attribute	[1, 1]	UUID of the request message for which this response message was sent.
		SendDateTime	Attribute	[1, 1]	Date and time of sending the message to the Tax administration.
	TCRCode		Element	[1, 1]	Code of the TCR generated by the CIS.
	Signature		Element	[1, 1]	XML element with signature.

Table 14

3.5.2.1   Header

Element representing the header of the response data message.

3.5.2.2   Header UUID

Element generated by the CIS. It uniquely identifies the response message sent from CIS to TCR. UUID should be constructed according to the RFC4122 version 4.

Data type	string
Length	36 characters
Pattern	[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[1-5][0-9a-fA-F]{3}-[89abAB][0-9a-fA-F]{3}-[0-9a-fA-F]{12}
Example	58e0a7d7-eebc-41d8-9669-0800200c9a66 58E0A7D7-EEBC-41D8-9669-0800200C9A66

Table 15

3.5.2.3   Header RequestUUID

Element generated by the TCR and referenced by the CIS. It uniquely identifies the request message for which the response message was sent to the TCR. UUID should be constructed according to the RFC4122 version 4.

Data type	String
Length	36 characters
Pattern	[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[1-5][0-9a-fA-F]{3}-[89abAB][0-9a-fA-F]{3}-[0-9a-fA-F]{12}
Example	58e0a7d7-eebc-41d8-9669-0800200c9a66 58E0A7D7-EEBC-41D8-9669-0800200C9A66

Table 16

3.5.2.4   Header SendDateTime

Element represents date and time of sending the response message to the TCR. Date and time should be in ISO 8601 format.

Data type	dateTime
Length	23 characters
Pattern	[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}[+-][0-9]{2}:[0-9]{2}
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 17

3.5.2.5   TCRCode

Code of the TCR device, generated by the CIS.

Data type	string
Length	10 characters
Pattern	[a-z]{2}[0-9]{3}[a-z]{2}[0-9]{3}
Example	ab123ab123

Table 18

3.5.2.6   Signature

XML element stores enveloped digital signature described in the chapter 4.3.1.

3.5.3 MANDATORY CONTROLS

Mandatory controls shall be performed by CIP system in process of registering the TCR. In case that the control is not passed, an error message will be returned with an error code.

Upon identifying a critical error, CIS will return an error data message containing the error's numeric code and its text description (see chapter 3.9). When errors which the system can interpret as a cyber-attack are identified, the system does not send any response to the client (the taxpayer's TCR).

The mandatory controls include all the controls from chapter 3.4 and the following:

Control Name	Control description (Error if)	Error code
Valid from in the past	TCR.ValidFrom cannot be in the past.	45
Valid to in the past	TCR.ValidTo cannot be in the past.	46
Valid to before valid from	TCR.ValidTo cannot be before TCR.ValidFrom.	47
Active TCR valid from change	TCR.ValidFrom different from active TCR valid from. Value of the valid from the active TCR cannot be changed.	48
Registered taxpayer	TCR.IssuerNUIS doesn't reference active taxpayer in the Registry of taxpayers.	52
Registered business unit	TCR.BusinUnitCode doesn't reference active business unit (premise) of the taxpayer.	41
Registered software	TCR.SoftCode doesn't references active software.	42
Registered maintainer	TCR.MaintainerCode doesn't references active maintainer.	43

Table 19

3.5.4 ERROR MESSAGE

Error message is defined in chapter 3.9

3.5.5 EXAMPLE XML

3.5.5.1 Request XML

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <RegisterTCRRequest xmlns="https://eFiskalizimi.tatime.gov.al/FiscalizationService/schema"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#" Id="Request" Version="2">
            <Header SendDateTime="2019-12-02T10:57:01+01:00" UUID="78b37523-3677-416a-8bc0-e0dd77296fc7"/>
            <TCR BusinUnitCode="bb123bb123" IssuerNUIS="L91806031N" MaintainerCode="mm123mm123" SoftCode="ss123ss123" TCRIntID="1"
ValidFrom="2019-12-05"/>
            <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/2001/04/xmldsig-more#rsa-sha256"/>
                    <Reference URI="#Request">
                        <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/2001/04/xmlenc#sha256"/>
                        <DigestValue>0ilvSabbyg8..........GzADJ0DO8=</DigestValue>
                    </Reference>
                </SignedInfo>
                <SignatureValue>tpGg48z..........0HmXIVOA==</SignatureValue>
                <KeyInfo>
                    <X509Data>
                        <X509Certificate>MIIFYDCCB..........Pifz0UlKJAanmqN3</X509Certificate>
                    </X509Data>
                </KeyInfo>
            </Signature>
        </RegisterTCRRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.5.5.2 Response XML

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
```

```xml
<SOAP-ENV:Body>
    <ns2:RegisterTCRResponse Id="Response" Version="2"
xmlns:ns2="https://eFiskalizimi.tatime.gov.al/FiscalizationService/schema" xmlns:ns3="http://www.w3.org/2000/09/xmldsig#">
        <ns2:Header RequestUUID="07986698-1502-4150-9520-1918a5bbbf49" SendDateTime="2019-11-27T09:49:22.168+01:00"
UUID="12839912-75b6-4cd4-ba2d-fca3e6395226"/>
        <ns2:TCRCode>vk822xf503</ns2:TCRCode>
        <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/2001/04/xmldsig-more#rsa-sha256"/>
                <Reference URI="#Response">
                    <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/2001/04/xmlenc#sha256"/>
                    <DigestValue>p+TSatU9..........+yD8ff87U=</DigestValue>
                </Reference>
            </SignedInfo>
            <SignatureValue>kFXWTPybI..........W6kKA3ojg==</SignatureValue>
            <KeyInfo>
                <X509Data>
                    <X509Certificate>MIIFRjCCBC..........gpNT2r23YQ==</X509Certificate>
                </X509Data>
            </KeyInfo>
        </Signature>
    </ns2:RegisterTCRResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.6   REGISTER CASH DEPOSIT

Every day before registering the first invoice of the day on CIS, each TCR for handling cash transactions must register the initial amount of cash deposit (Operation INITIAL). If needed TCR can correct initial deposit by registering cash amount again (Operation INITIAL).

During the day, the operator can put or take cash from the TCR and each of these actions must be registered (Operation INOUT).

In special cases defined by the Law (when there is failure of internet connection or if the taxpayer operates in the area without internet connection), information about opening deposit or withdrawal must be stored in TCR memory. If there is failure of TCR, data of the deposit/withdrawal is written in the bound invoice book and submitted to the Tax Administration later together with all invoices that are going to be fiscalized through that process.

The open cash deposit can be 0.00.

Cash balance registration is not required in the business premises that operates only (i.e. issues only) with non-cash invoices or in case of web-shop.
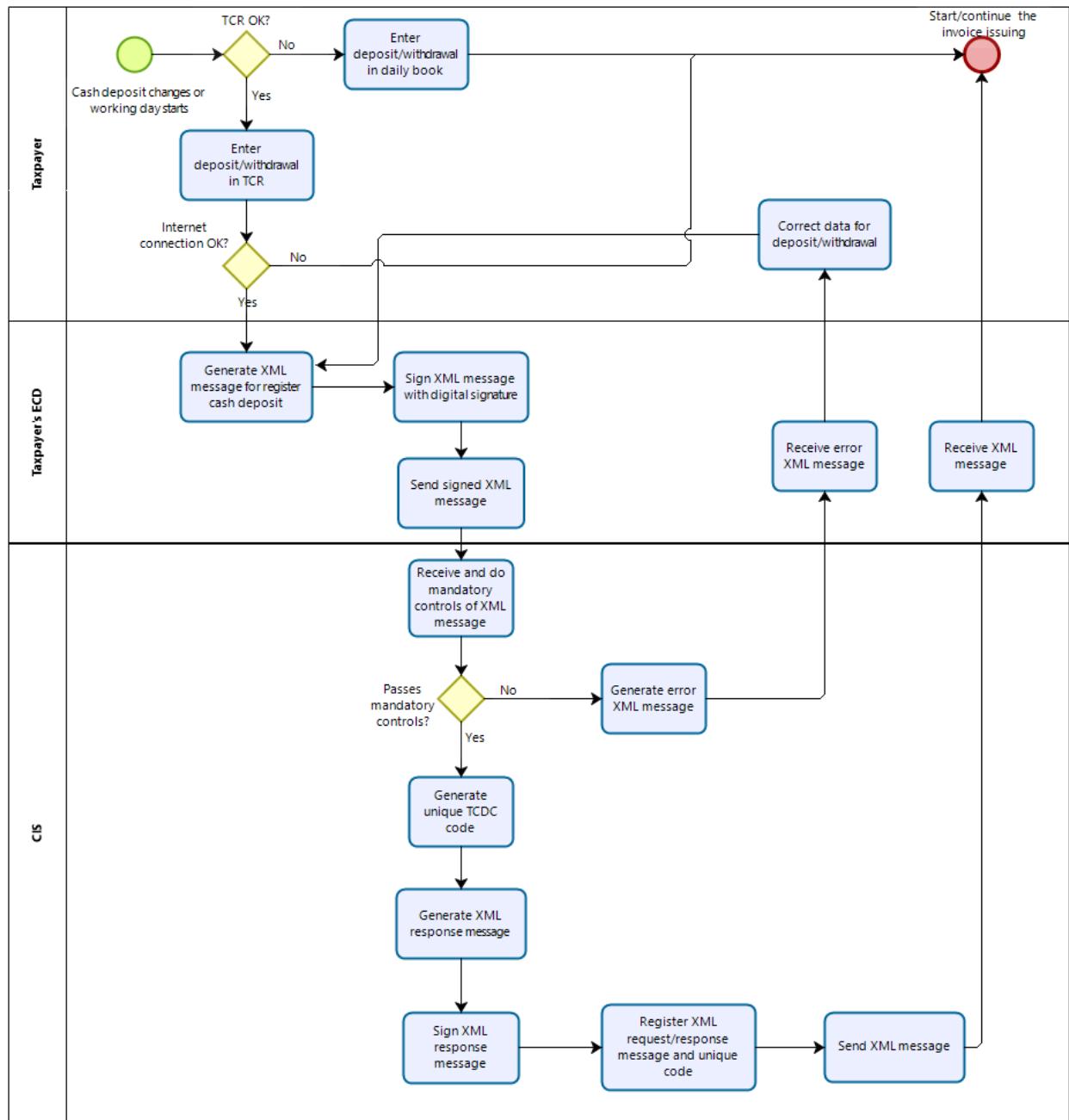
Figure 4 – Registration of the cash deposit

3.6.1 REGISTER CASH DEPOSIT REQUEST DATA MESSAGE

Name	Field type	Occurrence [Min, Max]	Description
RegisterCashDepositRequest	Element	[1, 1]	Root XML element representing registration of TCR first deposit.
Id	Attribute	[1, 1]	Attribute used for signature creation and verification. Fixed value "Request".
Version	Attribute	[1,1]	Attribute used to specify compliance with XSD schema. For this version fixed value is "2".
Header	Element	[1, 1]	XML element representing header of the message.
UUID	Attribute	[1, 1]	UUID generated by a TCR for every data message sent to the CIS.
SendDateTime	Attribute	[1, 1]	Date and time of sending the message from a TCR to the CIS.
IsSubseqDeliv	Attribute	[1, 1]	Is the request message delivered subsequently?
CashDeposit	Element	[1, 1]	XML element representing a single cash deposit request.
ChangeDateTime	Attribute	[1, 1]	Date and time when the cash deposit was changed.
Operation	Attribute	[1, 1]	Cash deposit operation made at the register.
CashAmt	Attribute	[1, 1]	Amount of the cash deposit in the TCR.
TCRCode	Attribute	[1, 1]	Code of the TCR for which the cash deposit is registered.
IssuerNUIS	Attribute	[1, 1]	Taxpayer's NUIS/NIPT.
Signature	Element	[1, 1]	XML element with signature.

Table 20

3.6.1.1 Header

Element representing the header of the request data message.

3.6.1.2 Header UUID

Element generated by the TCR. It uniquely identifies the request message sent from TCR to CIS. UUID should be constructed according to the RFC4122 version 4.

Data type	string
Length	36 characters
Pattern	[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[1-5][0-9a-fA-F]{3}-[89abAB][0-9a-fA-F]{3}-[0-9a-fA-F]{12}
Example	58e0a7d7-eebc-41d8-9669-0800200c9a66 58E0A7D7-EEBC-41D8-9669-0800200C9A66

Table 21

3.6.1.3 Header SendDateTime

Element represents date and time of sending the request message to the CIS. Date and time should be in ISO 8601 format.

Data type	dateTime
Length	23 characters
Pattern	[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}[+-][0-9]{2}:[0-9]{2}
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00
Mandatory controls	Error if Header.SendDateTime differences from server time more than 60 minutes.

Table 22

3.6.1.4 Header IsSubseqDeliv

Element that says if the request message is being delivered subsequently. For example, when an issuer has sent a message to the CIS but he didn't receive a response from the CIS. He can resend exactly the same message as before but only with flag IsSubseqDeliv. This flag is indication for CIS that an Issuer sent the message again.

Data type	boolean
Values	true, false
Example	true

Table 23

3.6.1.5 CashDeposit

Element representing a single cash deposit registration.

3.6.1.6 CashDeposit ChangeDateTime

Element representing date and time when the deposit of cash was changed in the cash register.

Data type	dateTime
Length	23 characters
Pattern	[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}[+-][0-9]{2}:[0-9]{2}
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00
Mandatory controls	Error if CashDeposit.ChangeDateTime differences from server time more than 60 minutes.

Table 24

3.6.1.7 CashDeposit Operation

This is an XML element that represents the operation made at the cash register.

Data type	string
Constraint	Enumeration, described in the table below.
Example	INITIAL

Table 25

Enumeration values for the operation are listed in the table below.

Value	Description
INITIAL	Initial balance in the TCR.
INOUT	Amount of cash deposited or withdrawn from the TCR.

Table 26

3.6.1.8 CashDeposit CashAmt

Element representing the amount of cash found in the cash register after the operation.

Data type	decimal
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0

Example	212.12 -212.12
Mandatory controls	Error if CashDeposit.CashAmt is negative and CashDeposit.Operation equals INITIAL. Error if CashDeposit.CashAmt is zero and CashDeposit.Operation equals INOUT.

Table 27

3.6.1.9 CashDeposit TCRCode

Code representing the unique number of the TCR in question.

Data type	string
Length	10 characters
Pattern	[a-z]{2}[0-9]{3}[a-z]{2}[0-9]{3}
Example	ab123ab123

Table 28

3.6.1.10 CashDeposit IssuerNUIS

Element representing issuer's NUIS/NIPT (tax number).

Data type	string
Length	10
Pattern	[a-zA-Z]{1}[0-9]{8}[a-zA-Z]{1}
Example	K72001008V

Table 29

3.6.1.11 Signature

XML element stores enveloped digital signature described in the chapter 4.3.1.

3.6.2 REGISTER CASH DEPOSIT RESPONSE DATA MESSAGE

Name	Field type	Occurrence [Min, Max]	Description
RegisterCashDepositResponse	Element	[1, 1]	Root XML element representing registration of cash deposit.
Id	Attribute	[1, 1]	Attribute used for signature creation and verification. Fixed value "Response".
Version	Attribute	[1,1]	Attribute used to specify compliance with XSD schema. For this version fixed value is "2".
Header	Element	[1, 1]	XML element representing header…
UUID	Attribute	[1, 1]	ID of the message.
RequestUUID	Attribute	[1, 1]	UUID of the request message for which this response message was sent.
SendDateTime	Attribute	[1, 1]	Date and time of sending the message to the Tax Administration.
FCDC	Element	[1, 1]	Fiscalization cash deposit code generated by the CIS.
Signature	Element	[1, 1]	XML element with signature.

Table 30

3.6.2.1 Header

Element representing the header of the response data message.

3.6.2.2 Header UUID

Element generated by the CIS. It uniquely identifies the response message sent from CIS to TCR. UUID should be constructed according to the RFC4122 version 4.

Data type	string
Length	36 characters
Pattern	[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[1-5][0-9a-fA-F]{3}-[89abAB][0-9a-fA-F]{3}-[0-9a-fA-F]{12}
Example	58e0a7d7-eebc-41d8-9669-0800200c9a66 58E0A7D7-EEBC-41D8-9669-0800200C9A66

Table 31

3.6.2.3 Header RequestUUID

Element generated by the TCR and referenced by the CIS. It uniquely identifies the request message for which response message was sent to the TCR. UUID should be constructed according to the RFC4122 version 4.

Data type	String
Length	36 characters
Pattern	[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[1-5][0-9a-fA-F]{3}-[89abAB][0-9a-fA-F]{3}-[0-9a-fA-F]{12}
Example	58e0a7d7-eebc-41d8-9669-0800200c9a66 58E0A7D7-EEBC-41D8-9669-0800200C9A66

Table 32

3.6.2.4 Header SendDateTime

Element represents date and time of sending the response message to the TCR. Date and time should be in ISO 8601 format.

Data type	dateTime
Length	23 characters
Pattern	[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}[+-][0-9]{2}:[0-9]{2}
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 33

3.6.2.5 FCDC

Fiscalization cash deposit code. Unique code generated by the CIS for every successful cash deposit registration.

Data type	String
Length	36 characters
Pattern	[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[1-5][0-9a-fA-F]{3}-[89abAB][0-9a-fA-F]{3}-[0-9a-fA-F]{12}
Example	58e0a7d7-eebc-41d8-9669-0800200c9a66 58E0A7D7-EEBC-41D8-9669-0800200C9A66

Table 34

3.6.2.6 Signature

XML element stores enveloped digital signature described in the chapter 4.3.1.

3.6.3 MANDATORY CONTROLS

Mandatory controls shall be performed by CIP system in process of registering the cash deposit of TCR. In case that the control is not passed, an error message will be returned with an error code.

Upon identifying a critical error, CIS will return an error data message containing the error's numeric code and its text description (see chapter 3.9). When errors which the system can interpret as a cyber-attack are identified, the system does not send any response to the client (the taxpayer's TCR).

The mandatory controls include all the controls from chapter 3.4 and the following:

Control Name	Control Description (Error if)	Error code
Check changed date and time	CashDeposit.ChangeDateTime differences from CIS time more than allowed time in minutes.	49
Check INITIAL amount	CashDeposit.CashAmt is negative and CashDeposit.Operation equals INITIAL.	50
Check INOUT amount	CashDeposit.CashAmt is zero and CashDeposit.Operation equals INOUT.	51
Registered taxpayer	CashDeposit.IssuerNUIS doesn't reference active taxpayer in the Registry of taxpayers.	52
Registered TCR	CashDeposit.TCRCode doesn't reference registered or active TCR or the TCR doesn't belongs to the referenced issuer.	53

Table 35

3.6.4 ERROR MESSAGE

Error message is defined in chapter 3.9.

3.6.5 EXAMPLE XML

3.6.5.1 Request XML

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <RegisterCashDepositRequest xmlns="https://eFiskalizimi.tatime.gov.al/FiscalizationService/schema"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#" Id="Request" Version="2">
            <Header SendDateTime="2019-12-05T14:35:00+01:00" UUID="3389b9c4-bb24-4673-b952-456e451cd3c3" IsSubseqDeliv="false"/>
            <CashDeposit CashAmt="2000.00" ChangeDateTime="2019-12-05T14:35:00+01:00" IssuerNUIS="L91806031N" Operation="INITIAL"
TCRCode="cc123cc123"/>
            <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/2001/04/xmldsig-more#rsa-sha256"/>
                    <Reference URI="#Request">
                        <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/2001/04/xmlenc#sha256"/>
                        <DigestValue>PTmGp/uQH......... LR+4IH2/hODo=</DigestValue>
                    </Reference>
                </SignedInfo>
                <SignatureValue>PDVluTMr5..........SspzFpjYkEA==</SignatureValue>
                <KeyInfo>
                    <X509Data>
                        <X509Certificate>MIIFYDCCB.......... ..........Pifz0UlKJAanmqN3</X509Certificate>
                    </X509Data>
                </KeyInfo>
            </Signature>
        </RegisterCashDepositRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.6.5.2 Response XML

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:RegisterCashDepositResponse Id="Response" Version="2"
xmlns:ns2="https://eFiskalizimi.tatime.gov.al/FiscalizationService/schema" xmlns:ns3="http://www.w3.org/2000/09/xmldsig#">
            <ns2:Header RequestUUID="5b685bba-fe1e-405c-ac8c-2243502dc9db" SendDateTime="2019-11-27T09:44:14.952+01:00"
UUID="ed3cbe5c-08ad-44e2-a6e1-a811e3c80137"/>
            <ns2:FCBC>c8a197f1-b7dc-4e82-8729-4a506aecf9b9</ns2:FCBC>
```

```xml
<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/2001/04/xmldsig-more#rsa-sha256"/>
        <Reference URI="#Response">
            <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/2001/04/xmlenc#sha256"/>
            <DigestValue>giebs7Pyu.........rs+xz0iS3f0=</DigestValue>
        </Reference>
    </SignedInfo>
    <SignatureValue>BwiADbuZFaa.........fd2Uo3LEiClw==</SignatureValue>
    <KeyInfo>
        <X509Data>
            <X509Certificate>MIIFRjCCBC6g.........JfgpNT2r23YQ==</X509Certificate>
        </X509Data>
    </KeyInfo>
</Signature>
        </ns2:RegisterCashDepositResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.7 REGISTER INVOICE

Issuer of the invoice is obliged to deliver information on each invoice he issues. Information has to be delivered at the moment of issuing. Exceptionally, it can be delivered afterwards (as stipulated in the Law).

Data exchange process starts at the moment when the issuer is about to issue an invoice to the customer. The TCR prepares invoice data and based on that data it creates IIC. After that it prepares XML invoice request message and signs it electronically by its certificate private key (using a certificate that was issued to issuer by CIS with purpose being the implementation of fiscalization). After that the 1-way TLS communication is started and once successful it calls the service.

Central information system receives and processes request message. If the request is successfully processed, central information system prepares XML message that contains FIC, which is unique for every invoice, signs it electronically with its certificate, and sends it back to the TCR.

TCR receives answer message and checks its electronic signature. After that, cashier (operator) issues the invoice and hands it to the buyer.

Corrective invoice is a special type of invoice which contains a reference to the original invoice and is issued in order to change some data in the original invoice, e.g. some items from the original invoice should be removed because they are returned from the buyer to the seller.

If there were errors during the operation (invalid XML, invalid certificate or similar), central information system shows the error as XML message. If that is the case, there is no FIC so issuer will issue the invoice without FIC. The invoice issuing process must not be halted because of the error, but the issuer is obliged to correct the error and deliver it after he receives the correct message.

In all situations when issuer does not get FIC for invoice he issued (loss of Internet connection, computer breakdown, central information system unavailability or similar), he is obliged to make another invoice request. Invoice is found to be properly sent and reported to the CIS once issuer gets FIC for it.

In cases when there are invoices without FIC, those should be sent again later (and in timeframe defined in laws), as invoices processed at the moment have advantage over invoices issued prior. Invoices without FIC should be delivered when traffic load is smaller or immediately and automatically when Internet connection becomes available again.

When the taxpayer is operating in the area without internet connection, he can export invoice registration request into special file format which can be then delivered through the Central invoice platform with bulk upload of invoice or brought to the Tax Administration office.

Maximum time-out for machine to wait for the answer that contains FIC is set by issuer himself. Issuer needs to check the Internet connection quality and time needed for issuing one receipt so that will not affect his business. When calculating maximum time-out, issuer should count in additional two seconds (time needed for request to come in and get out of process).

Invoice registration can be verified by online application. Each invoice contains a QR-code which contains a link to the web page which displays information about the invoice, if it was successfully registered. Details are explained in subchapter 3.7.7.
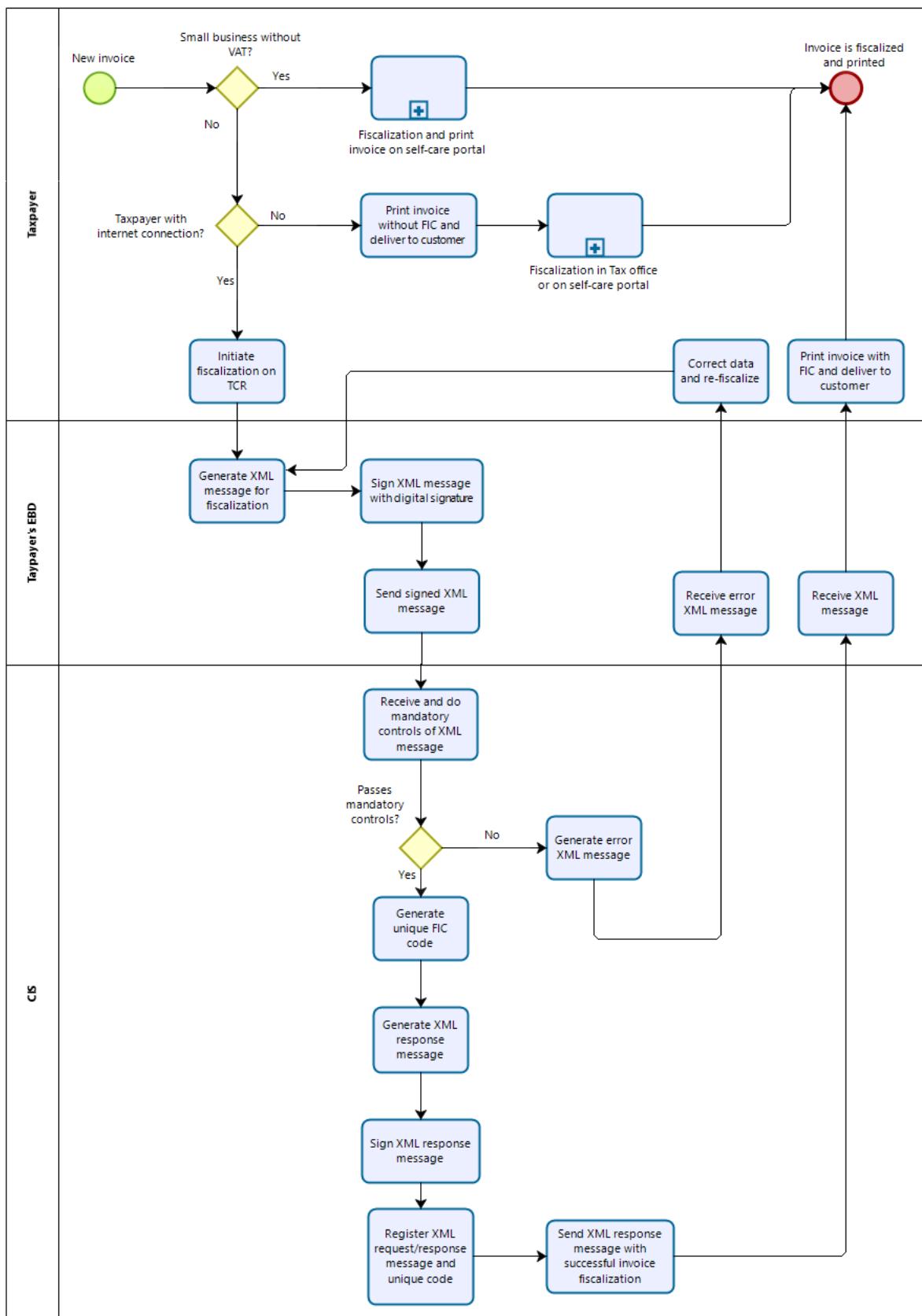
Figure 5 – Fiscalization of invoices
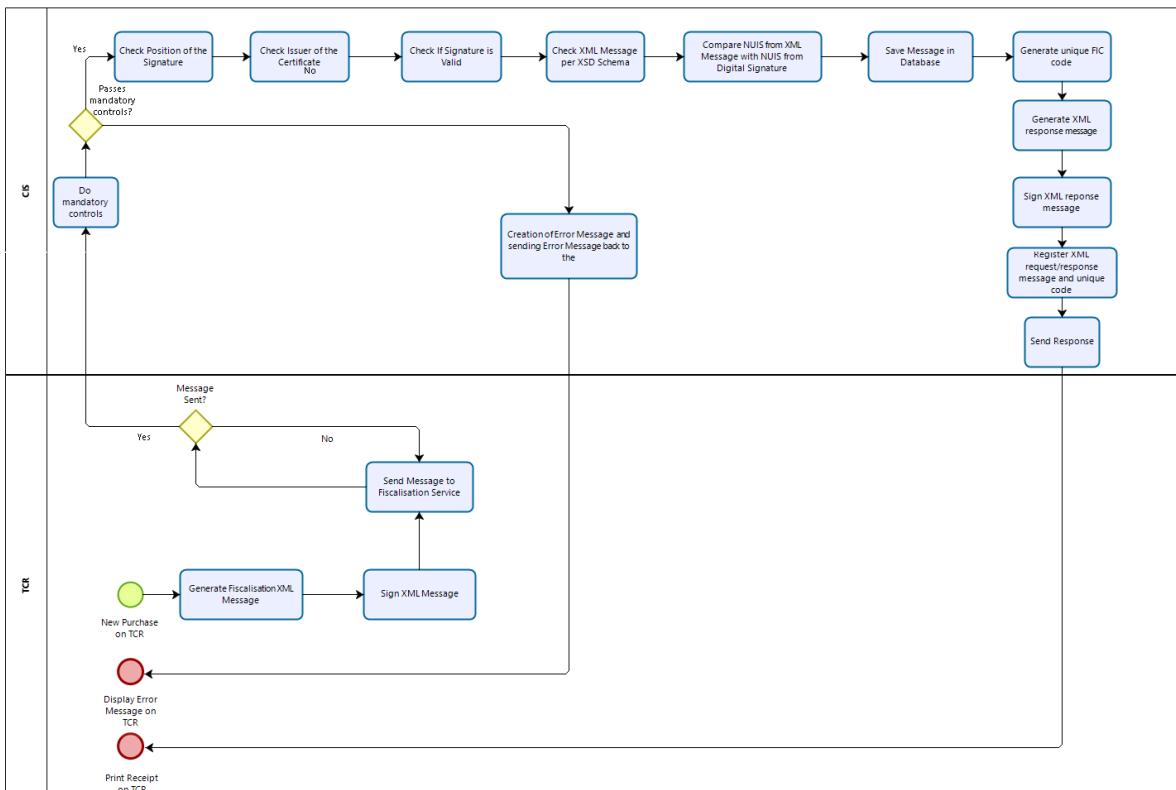
Figure 6 - XML validations

CIS processes the registration of the invoice in the following way:

- TLS communication is initiated between the invoice issuer and CIS
  - In case that certificate provided by the invoice issuer is invalid, communication protocol fails
- CIS checks the size of the message
  - If the message is larger than allowed, an error message is sent in the response and no further processing is done.
- Further controls are performed. For each passed control, process continues to the next control. If the control fails, no more controls are performed, and error message is prepared. Following controls are performed:
  - SOAP message is a valid XML document
  - SOAP message is structured according to the defined schema
  - Certificate in the signature of the message is checked
  - Signature of the message is checked
  - IIC is verified
  - Date and time written in the message is checked
  - Is issuer in the VAT register
- If one of the controls failed, an error message will be sent in the reply to the request message with the error code related to the nature of the error.
- If all of the controls have passed successfully, register invoice response message will be sent.
  - FIC code is generated.
  - Response message is prepared with FIC included.
  - Response message is signed, and signature is put into the message.
- Invoice data is inserted into the database.
  - Data includes request SOAP message, response SOAP message (or error message).

- Prepared response message is sent back to the taxpayer.

3.7.1 REGISTER INVOICE REQUEST DATA MESSAGE

This is the request message sent by the issuer of the invoice to CIS.

Name	Field type	Occurrence [Min, Max]	Description
RegisterInvoiceRequest	Root	[1, 1]	Root XML element representing register invoice message.
Id	Attribute	[1, 1]	Attribute used for signature creation and verification. Fixed value "Request".
Version	Attribute	[1,1]	Attribute used to specify compliance with XSD schema. For this version fixed value is "2".
Header	Element	[1, 1]	XML element representing header of the invoice containing data about the message (request) sent.
UUID	Attribute	[1, 1]	UUID generated by a TCR for every register sale data message send to the CIS.
SendDateTime	Attribute	[1, 1]	Date and time of sending the register invoice data message from a TCR to the CIS.
IsSubseqDeliv	Attribute	[1, 1]	Is the request message delivered subsequently?
Invoice	Element	[1, 1]	XML element representing a single invoice.
TypeOfInv	Attribute	[1, 1]	Type of the invoice (cash, non-cash).
TypeOfSelfIss	Attribute	[0, 1]	Entered only if invoice is self-issued.
IsSimplifiedInv	Attribute	[1, 1]	Is invoice simplified
IssueDateTime	Attribute	[1, 1]	Date and time when the invoice is created and issued at TCR.
InvNum	Attribute	[1, 1]	Invoice number composed of invoice ordinal number, year of invoice issuance and code of TCR that issued invoice.
InvOrdNum	Attribute	[1, 1]	Invoice ordinal number.
TCRCode	Attribute	[0, 1]	Code of the device that issued the invoice. Mandatory if the TypeOfInv is different from NONCASH.
IsIssuerInVAT	Attribute	[1, 1]	Issuer is in VAT register
TaxFreeAmt	Attribute	[0, 1]	The total amount of goods and services delivered when VAT is not charged
MarkUpAmt	Attribute	[0, 1]	Amount related to special procedure for margin scheme
GoodsExAmt	Attribute	[0, 1]	Amount of goods for export from the Republic of Albania.
TotPriceWoVAT	Attribute	[1, 1]	Total price of the invoice excluding VAT.
TotVATAmt	Attribute	[0, 1]	Total VAT amount of the invoice. Mandatory if IsIssuerInVAT equals true. Mandatory if TypeOfSelfIss exists and IsIssuerInVAT equals true. Mandatory if IsReverseCharge equals true.
TotPrice	Attribute	[1, 1]	Total price of all items including taxes and discounts
OperatorCode	Attribute	[1, 1]	Reference to the operator code, who is operating on TCR and issues invoices.
BusinUnitCode	Attribute	[1, 1]	Business unit (premise) code.
SoftCode	Attribute	[1, 1]	Software code.
IIC	Attribute	[1, 1]	Issuer's invoice code calculated as MD5 hash from IICSignature attribute.
IICSignature	Attribute	[1, 1]	Signed issuer's invoice code concatenated parameters.
CorrectiveInv	Element	[0, 1]	XML element groups data for an original invoice that will be corrected with current invoice.
IICRef	Attribute	[1, 1]	IIC reference on the original invoice.
IssueDateTime	Attribute	[1, 1]	Date and time the original invoice is created and issued at TCR.
Type	Attribute	[1, 1]	Type of the corrective invoice.
IsReverseCharge	Attribute	[1, 1]	If true, the buyer is obliged to pay the VAT.
IsBadDebt	Attribute	[1, 1]	If an invoice is declared as uncollectible, it must have this attribute set to true
PayDeadline	Attribute	[0, 1]	Last day for payment.
SumInvIICRefs	Element	[0, 1]	XML element that contains list of IIC-s to which this invoice referred to, e.g. if this is a summary invoice it shall contain a reference to each individual invoice issued and fiscalized before and included in this summary invoice.
SumInvIICRef	Element	[1, 1000]	XML element that contains one IIC reference, e.g. reference of the invoice that is part of the summary invoice.

				IIC	Attribute	[1, 1]	IIC of the invoice that is referenced in the summary invoice.
				IssueDateTime	Attribute	[1, 1]	Date and time the invoice referenced by the summary invoice is created and issued at TCR.
			SupplyDateOrPeriod		Element	[0, 1]	XML element representing supply date or period of supply, if it is different from the date when the invoice was issued.
				Start	Attribute	[1, 1]	Start day of the supply.
				End	Attribute	[1, 1]	End day of the supply.
			PayMethods		Element	[1, 1]	XML element representing list of payment methods.
				PayMethod	Element	[1, 10]	XML element representing one payment method.
					Type	Attribute	[1, 1]	Type of the payment method.
					Amt	Attribute	[0, 1]	Amount payed by payment method in the currency in which the invoice was issued. Mandatory if multiple payment methods exist.
					CompCard	Attribute	[0, 1]	Company card number if the payment method is company card. Mandatory if PayMethod.Type equals COMPANY.
					Vouchers	Element	[0, 1]	XML element that contains list of voucher numbers if the payment method is voucher. Mandatory if PayMethod.Type equals SVOUCHER or MVOUCHER.
						Voucher	Element	[1, 20]	XML element that contains one voucher number.
							Num	Attribute	[1,1]	Voucher number
			Currency		Element	[0, 1]	XML element representing currency in which the amount on the invoice is expressed.
				Code	Attribute	[1, 1]	Currency code in which the amount on the invoice is expressed.
				ExRate	Attribute	[1, 1]	Exchange rate applied to calculate the equivalent amount of the total amount expressed in currency different from ALL.
			Seller		Element	[1, 1]	XML element representing seller's data.
				IDType	Attribute	[1, 1]	Seller's identification number type.
				IDNum	Attribute	[1, 1]	Seller's identification number.
				Name	Attribute	[1, 1]	Seller's name.
				Address	Attribute	[0, 1]	Seller's address. Mandatory if the seller is foreigner.
				Town	Attribute	[0, 1]	Seller's town. Mandatory if the seller is foreigner.
				Country	Attribute	[0, 1]	Seller's country. Mandatory if the seller is foreigner.
			Buyer		Element	[0, 1]	XML element representing buyer's data.
				IDType	Attribute	[0, 1]	Buyer's identification number type. Mandatory if the Invoice.TypeOfSelfIss exists. Mandatory if the Invoice.GoodsExAmt exists.
				IDNum	Attribute	[0, 1]	Buyer's identification number. Mandatory if the Invoice.TypeOfSelfIss exists. Mandatory if the Invoice.GoodsExAmt exists.
				Name	Attribute	[0, 1]	Buyer's name. Mandatory if the Invoice.Buyer.IDType and Invoice.Buyer.IDNum exists.
				Address	Attribute	[0, 1]	Buyer's address. Mandatory if the Invoice.Buyer.IDType and Invoice.Buyer.IDNum exists.
				Town	Attribute	[0, 1]	Buyer's town. Mandatory if the Invoice.Buyer.IDType and Invoice.Buyer.IDNum exists.
				Country	Attribute	[0, 1]	Buyer's country. Mandatory if the Invoice.Buyer.IDType and Invoice.Buyer.IDNum exists.
			Items		Element	[1, 1]	XML element representing list of invoice items.
				I (Item)	Element	[1, 1000]	XML element representing one item.
					N (Name)	Attribute	[1, 1]	Name of the item (goods or services).
					C (Code)	Attribute	[0, 1]	Code of the item from the barcode or similar representation.
					U (Unit of measure)	Attribute	[1, 1]	What is the item's unit of measure (piece, weight measure, length measure, etc.)
					Q (Quantity)	Attribute	[1, 1]	Amount or number (quantity) of items
					UPB (Unit price without VAT)	Attribute	[1,1]	Unit price before Value added tax is applied
					UPA (Unit Price with VAT)	Attribute	[0, 1]	Unit price after Value added tax is applied. Mandatory if IsIssuerInVAT equals true. Mandatory if TypeOfSelfIss exists and IsIssuerInVAT equals true. Mandatory if IsReverseCharge equals true.
					R (Rebate)	Attribute	[0, 1]	Percentage of the rebate.
					RR (Rebate Reducing base price)	Attribute	[0, 1]	Is rebate reducing base amount?
					PB (Price Before VAT)	Attribute	[1, 1]	Total price of goods and services before the tax
					VR (VAT Rate)	Attribute	[0, 1]	Rate of value added tax. Mandatory if IsIssuerInVAT equals true. Mandatory if TypeOfSelfIss exists and IsIssuerInVAT equals true.

						Mandatory if IsReverseCharge equals true.
		EX (Exempt from VAT)		Attribute	[0, 1]	Exempt from VAT.
		VA (VAT Amount)		Attribute	[0, 1]	Amount of value added tax for goods and services. Mandatory if IsIssuerInVAT equals true. Mandatory if TypeOfSelfIss exists and IsIssuerInVAT equals true. Mandatory if IsReverseCharge equals true.
		PA (Price After applying VAT)		Attribute	[1, 1]	Total price of goods after the tax and applying discounts
	SameTaxescount			Element	[0, 1]	XML element representing list of the aggregated items that go under same tax rate/exemption. Mandatory if IsIssuerInVAT equals true. Mandatory if TypeOfSelfIss exists and IsIssuerInVAT equals true. Mandatory if IsReverseCharge equals true.
		SameTax		Element	[1, 20]	XML element representing one same tax item.
			NumOfItems	Attribute	[1, 1]	Number of items.
			PriceBefVAT	Attribute	[1, 1]	Price before VAT.
			VATRate	Attribute	[0, 1]	VAT rate.
			ExemptFromVAT	Attribute	[0, 1]	Exempt from VAT.
			VATAmt	Attribute	[0, 1]	VAT amount.
	ConsTaxes			Element	[0, 1]	XML element representing list of special, consumption taxes.
		ConsTax		Element	[1, 20]	XML element representing one cons tax item.
			NumOfItems	Attribute	[1, 1]	Number of items under consumption tax.
			PriceBefConsTax	Attribute	[1, 1]	Price before adding consumption tax.
			ConsTaxRate	Attribute	[1, 1]	Rate of the consumption tax.
			ConsTaxAmt	Attribute	[1, 1]	Amount of consumption tax.
	Fees			Element	[0, 1]	XML element representing list of fees.
		Fee		Element	[1, 20]	XML element representing one fee.
			Type	Attribute	[1, 1]	Type of the fee.
			Amt	Attribute	[1, 1]	Amount of the fee.
	Signature			Element	[1, 1]	XML element representing signature for the invoice.

Table 36

3.7.1.1 Header

XML element representing header of the request data message.

3.7.1.2 Header UUID

Element generated by the TCR. It uniquely identifies the request message sent from TCR to CIS. UUID should be constructed according to the RFC4122 version 4.

Data type	string
Length	36 characters
Pattern	[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[1-5][0-9a-fA-F]{3}-[89abAB][0-9a-fA-F]{3}-[0-9a-fA-F]{12}
Example	58e0a7d7-eebc-41d8-9669-0800200c9a66 58E0A7D7-EEBC-41D8-9669-0800200C9A66

3.7.1.3 Header SendDateTime

Element represents date and time of sending the request message to the CIS. Date and time should be in ISO 8601 format.

Data type	dateTime
Length	23 characters

Pattern	[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}[+-][0-9]{2}:[0-9]{2}
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

<p align="center">Table 37</p>

3.7.1.4 Header IsSubseqDeliv

Element that says if the request message is being delivered subsequently. For example, when an issuer has sent a message to the CIS but he didn't receive a response from the CIS. He can resend exactly the same message as before but only with flag IsSubseqDeliv. This flag is indication for CIS that an Issuer sent the message again.

Data type	boolean
Values	true, false
Example	true

<p align="center">Table 38</p>

3.7.1.5 Invoice

XML element representing a single invoice.

3.7.1.6 Invoice TypeOfInv

Type of the item represents the type of invoice item, e.g. regular sale or a returned item.

Data type	string
Values	Enumeration, described in the table below.
Example	CASH

<p align="center">Table 39</p>

Following table shows the list of allowed values inside TypeOfInv attribute.

Value	Description
CASH	Cash
NONCASH	Non-cash

<p align="center">Table 40</p>

3.7.1.7 Invoice TypeOfSelfIss

This element shows the type of self-issuing.

Data type	string
Values	Enumeration, described in the table below.
Example	ABROAD

<p align="center">Table 41</p>

Following table shows the list of allowed values inside TypeOfSelfIss attribute.

Value	Description
AGREEMENT	The previous agreement between the parties.
DOMESTIC	Purchase from domestic farmers.

ABROAD	Purchase of services from abroad.
OTHER	Other

Table 42

3.7.1.8   Invoice IsSimplifiedInv

Is invoice simplified.

Data type	boolean
Values	true, false
Example	true

Table 43

3.7.1.9   Invoice IssueDateTime

Time and date when the invoice is created and issued at TCR.

Data type	string
Length	23 characters
Pattern	[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}[+-][0-9]{2}:[0-9]{2}
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 44

3.7.1.10   Invoice InvNum

Invoice number composed of invoice ordinal number, year of invoice issuance and code of the TCR that issued invoice if the invoice is is not equal to NONCASH. Invoice ordinal number is a sequence that is assigned to each new invoice so that the invoices can be counted. The sequence is reset at the beginning of each year.

Data type	string
Pattern	[0-9][1-9]{0,14}\/[0-9]{4}(\/[a-z]{2}[0-9]{3}[a-z]{2}[0-9]{3})?
Example	TypeOfInv is not equal to NONCASH: 9934/2019/ab123ab123 TypeOfInv is equal to NONCASH: 9934/2019

Table 45

3.7.1.11   Invoice InvOrdNum

Invoice ordinal number. Invoice ordinal number is a sequence that is assigned to each new invoice so that the invoices can be counted. The sequence is reset at the beginning of each year.

Data type	string
Pattern	[0-9][1-9]{0,14}
Example	9934

Table 46

3.7.1.12   Invoice TCRCode

Code of the device that issued the invoice.

Data type	string
Pattern	[a-z]{2}[0-9]{3}[a-z]{2}[0-9]{3}
Example	ab123ab123

Table 47

3.7.1.13   Invoice IsIssuerInVAT

Is taxpayer in the VAT register.

Data type	boolean
Values	true, false
Example	true

Table 48

3.7.1.14   Invoice TaxFreeAmt

Invoice amount that is exempted from VAT, either because the taxpayer is not in the VAT register or some other exemption applied that is different from the one in other fields of this XML message.

Data type	decimal
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	17.24

Table 49

3.7.1.15   Invoice MarkUpAmt

The total amount pertaining to the special margin scheme procedure in the invoice in decimal form (the taxable amount). The margin for used goods, works of art, collectibles or antiques.

Data type	decimal
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	23.10

Table 50

3.7.1.16   Invoice GoodsExAmt

Total price of delivery of exported goods. There is no VAT on the invoice.

Data type	decimal
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	246.00

Table 51

3.7.1.17   Invoice TotPriceWoVAT

Total amount of the invoice without VAT.

Data type	decimal

Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	212.12

Table 52

3.7.1.18   Invoice TotVATAmt

Total amount of VAT (value added tax) which needs to be payed for all groups of items listed in this invoice.

Data type	decimal
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	242.23

Table 53

3.7.1.19   Invoice TotPrice

Total price which needs to be payed by the customer for all groups of items listed in this invoice including VAT.

Data type	decimal
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	212.12

Table 54

3.7.1.20   Invoice OperatorCode

Reference to the operator who is operating on TCR. Value represents code of the operator.

Data type	string
Length	10 characters
Pattern	[a-z]{2}[0-9]{3}[a-z]{2}[0-9]{3}
Example	ab123ab123

Table 55

3.7.1.21   Invoice BusinUnitCode

Code of the business unit (premise) in which the invoice is issued.

Data type	string
Length	10 characters
Pattern	[a-z]{2}[0-9]{3}[a-z]{2}[0-9]{3}
Example	ab123ab123

Table 56

3.7.1.22   Invoice SoftCode

Code of the software used for invoice issuing.

Data type	string

Length	10 characters
Pattern	[a-z]{2}[0-9]{3}[a-z]{2}[0-9]{3}
Example	ab123ab123

Table 57

3.7.1.23 Invoice IIC

Issuers's invoice code which is generated by the TCR of the issuer of the invoice. This is a unique code for every invoice. The code is formed by concatenating the fields, signing with issuer's private key and calculating MD5 hash. Further description can be found in the chapter 4.3.2.3.

Data type	string
Max length	32
Pattern	[0-9a-fA-F]{32}
Example	C701FB4839E7D2C3D8DBC81BBAC06164 c701fb4839e7d2c3d8dbc81bbac06164

Table 58

3.7.1.24 Invoice IICSignature

Signed issuer's invoice code concatenated parameters. Further description can be found tin the chapter 4.3.2.2.

Data type	string
Max length	512
Pattern	[0-9a-fA-F]{512}
Example	B2C218486302EC553EE1AB9124E1A14705742E870E8872EF34E63617AB252E189ACDF7A3E3F5C82061FFFF8AC2826A5588596A8807F648410899B6193F77F4BDCDFA875 53A62079A2EF9E6E6F0B8DA1038968D2FCB920B580EBF33ACEEDFEA0DAA78067F916ADC5D278CC237EFD53A6156EABAFBE98A8F3CE99E854818822FA20C0FF46E5B380 5264BBCD085F0A8A9BD503A1304E9202D7304FF93541FB7FAA4629EE0BD7ED566F610DCD047721AEAA828DFECA651087CDE5AF95C125793D4CD8E83B801DE171335A8 66D7E31F1473BF0C93EBFD994326C0FE97ACB8DA722F788EA27B8D9E15E8E7B6EF772AB7534060F2BCAF1C3E82645235C9D1857B0790C2

Table 59

3.7.1.25 Invoice CorrectiveInv

XML element groups data for an original invoice that is will be corrected with current invoice.

3.7.1.26 Invoice CorrectiveInv IICRef

Reference to the invoice IIC of the original invoice. It is entered only if this is a corrective invoice of the original invoice that has to be changed.

Data type	string
Max length	32
Pattern	[0-9a-fA-F]{32}
Example	C701FB4839E7D2C3D8DBC81BBAC06164 c701fb4839e7d2c3d8dbc81bbac06164

Table 60

3.7.1.27 Invoice CorrectiveInv IssueDateTime

Date and time the original invoice is created and issued at TCR.

Data type	string
Length	23 characters
Pattern	[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}[+-][0-9]{2}:[0-9]{2}
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 61

3.7.1.28 Invoice CorrectiveInv Type

Type of the corrective invoice.

Data type	string
Constraint	Enumeration, described in the table below.
Example	DEBIT

Table 62

Enumeration values for corrective invoice types are listed in table below.

Value	Description
CORRECTIVE	Corrective invoice
DEBIT	Debit note
CREDIT	Credit note

Table 63

3.7.1.29 Invoice IsReverseCharge

Buyer is obliged to pay VAT by himself instead of the seller.

Data type	boolean
Values	true, false
Example	true

Table 64

3.7.1.30 Invoice IsBadDebt

If the invoice is marked as unpayable, it gets "bad debt" note.

Data type	boolean
Values	true, false
Example	true

Table 65

3.7.1.31 Invoice PayDeadline

Last day for payment.

Data type	string
Length	10 characters
Pattern	[0-9]{4}-[0-9]{2}-[0-9]{2}

Example	2019-01-24

Table 66

3.7.1.32   Invoice SumInvIICRefs

XML element that contains list of IIC-s referenced by this Summary invoice, e.g. individual invoices (or invoice/orders) that are included in the Summary invoice and which had payment method type COMPANY or ORDER.

3.7.1.33   Invoice SumInvIICRefs SumInvIICRef

XML element that contains one IIC reference, e.g. reference of the invoice that is part of the summary invoice.

3.7.1.34   Invoice SumInvIICRefs SumInvIICRef IIC

IIC of the invoice that is referenced in the summary invoice.

Data type	string
Max length	32
Pattern	[0-9a-fA-F]{32}
Example	C701FB4839E7D2C3D8DBC81BBAC06164 c701fb4839e7d2c3d8dbc81bbac06164

Table 67

3.7.1.35   Invoice SumInvIICRefs SumInvIICRef IssueDateTime

Date and time the invoice referenced by the summary invoice is created and issued at TCR.

Data type	string
Length	23 characters
Pattern	[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}[+-][0-9]{2}:[0-9]{2}
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 68

3.7.1.36   Invoice SupplyDateOrPeriod

XML element representing supply date or period of supply, if different from the date when the invoice was issued.

3.7.1.37   Invoice SupplyDateOrPeriod Start

Start date of the supply. To represent a specific date, start date must be same as the end date. To represent a period, start date must be before end date and in the same month as the end date.

Data type	string
Length	10
Pattern	[0-9]{4}-[0-9]{2}-[0-9]{2}
Example	2019-01-24

Table 69

3.7.1.38 Invoice SupplyDateOrPeriod End

End date of the supply. To represent a specific date, end date must be same as the start date. To represent a period, end date must be after start date, and in the same month as start date.

Data type	string
Length	10
Pattern	[0-9]{4}-[0-9]{2}-[0-9]{2}
Example	2019-01-24

Table 70

3.7.1.39 Invoice PayMethods

XML element representing list of invoice payment methods.

3.7.1.40 Invoice PayMethods PayMethod

XML element representing a single payment method on the list of payment methods.

3.7.1.41 Invoice PayMethods PayMethod Type

Method of payment.

Data type	string
Constraint	Enumeration, described in the table below.
Example	BANKNOTE

Table 71

Enumeration values for the method of payment are listed in table below.

Value	Description
BANKNOTE	Banknotes and coins
CARD	Credit and debit card
CHECK	Bank check
SVOUCHER	Single-purpose voucher
MVOUCHER	Multi-purpose voucher
COMPANY	Seller's company cards and similar
ORDER	Invoice not yet paid. It will be paid by summary invoice.
ACCOUNT	Transaction account
FACTORING	Factoring
COMPENSATION	Compensation
TRANSFER	Transfer of rights or debts
WAIVER	Waiver of debts
KIND	Payment in kind (clearing)
OTHER	Other cashless payments

Table 72

3.7.1.42　Invoice PayMethods PayMethod Amt

Total price payed with one payment method.

Data type	decimal
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	212.12

3.7.1.43　Invoice PayMethods PayMethod CompCard

Company card number if the payment method is company card.

Data type	string
Length	50
Example	1257896

3.7.1.44　Invoice PayMethods PayMethod Vouchers

XML element that contains list of voucher numbers if the payment method is voucher. It can be more than one voucher used for the same invoice (i.e. for the same supply, but one voucher cannot be split for 2 or more supplies).

3.7.1.45　Invoice PayMethods PayMethod Vouchers Voucher

XML element that contains one voucher number.

3.7.1.46　Invoice PayMethods PayMethod Vouchers Voucher Num

Voucher number.

Data type	string
Length	17-25
Pattern	[1-9][0-9]{0,7}-[0-9]{4}-[a-zA-Z]{1}[0-9]{8}[a-zA-Z]{1}
Example	2-2020-J43675678H 34564-2020-J43675678H

3.7.1.47　Invoice Currency

XML element representing currency in which the amount on the invoice is expressed.

3.7.1.48　Invoice Currency Code

Currency code in which the amount on the invoice is expressed. Code is expressed by the ISO 4217 standard.

Data type	string
Constraint	Enumeration, described in the table below.
Example	EUR

Enumeration values for selected currency codes are listed in table below. Other values can be found in the fiscalization service shema in the chapter 7.

Value	Description
ALL	Albanian lek
EUR	Euro
GRD	Greek drachma
MKD	Macedonian denar
TRY	Turkish lira
BGN	Bulgarian lev
BAM	Bosnia and Herzegovina convertible mark
HRK	Croatian kuna
…	…

Table 77

3.7.1.49   Invoice Currency ExRate

Exchange rate applied to calculate the equivalent amount in ALL of the total amount expressed in foreign currency.

Data type	Double
Constraints	Must be positive number.
Example	3.500 0.375

Table 78

3.7.1.50   Invoice Seller

XML element representing a seller.

3.7.1.51   Invoice Seller IDNum

Seller's identification number.

Data type	string
Length	20
Example	For NUIS: K72001008V For Social security number: 123-45-6789

Table 79

3.7.1.52   Invoice Seller IDType

Seller's identification number type.

Data type	string
Constraint	Enumeration, described in the table below.
Example	NUIS

Table 80

Enumeration values for the identification number type are listed in table below.

Value	Description
NUIS	NUIS number
ID	Personal ID number
PASS	Passport number
VAT	VAT number
TAX	TAX number
SOC	Social security number

Table 81

3.7.1.53 Invoice Seller Name

Seller's name.

Data type	string
Length	100 characters
Example	Name Surname

Table 82

3.7.1.54 Invoice Seller Address

Seller's address.

Data type	string
Length	100 characters
Example	Plaza Tirana 1

Table 83

3.7.1.55 Invoice Seller Town

Seller's town.

Data type	string
Length	100 characters
Example	Tirana

Table 84

3.7.1.56 Invoice Seller Country

Seller's country represented as ISO 3166-1 Alpha-3 code.

Data type	string
Constraint	Enumeration, described in the table below.
Example	ALB

Table 85

Enumeration values for selected countries are listed in table below.

Value	Description

ALB	Albania
GRC	Greece
MKD	North Macedonia
XKX	Kosovo
MNE	Montenegro
ITA	Italy
…	…

3.7.1.57   Invoice Buyer

XML element representing a buyer that purchase goods.

3.7.1.58   Invoice Buyer IDNum

Buyer's identification number.

Data type	string
Length	20
Example	For NUIS: K72001008V For Social security number: 123-45-6789

3.7.1.59   Invoice Buyer IDType

Buyer's identification number type.

Data type	string
Constraint	Enumeration, described in the table below.
Example	NUIS

Enumeration values for the identification number type are listed in table below.

Value	Description
NUIS	NUIS number
ID	Personal ID number
PASS	Passport number
VAT	VAT number
TAX	TAX number
SOC	Social security number

3.7.1.60   Invoice Buyer Name

Buyer's name.

Data type	string
Length	100 characters

Example	Name Surname

Table 90

3.7.1.61 Invoice Buyer Address

Buyer's address.

Data type	string
Length	100 characters
Example	Street Name 888

3.7.1.62 Invoice Buyer Town

Buyer's town.

Data type	string
Length	100 characters
Example	Tirana

Table 91

3.7.1.63 Invoice Buyer Country

Buyer's country represented as ISO 3166-1 Alpha-3 code.

Data type	string
Constraint	Enumeration, described in the table below.
Example	ALB

Table 92

Enumeration values for selected countries are listed in table below.

Value	Description
ALB	Albania
GRC	Greece
MKD	North Macedonia
XKX	Kosovo
MNE	Montenegro
ITA	Italy
…	…

Table 93

3.7.1.64 Invoice Items

XML element representing list of invoice items (goods or services). Items which are the same should be grouped as one item (one XML element called "Item") with the appropriate amount (sum of the samev items).

3.7.1.65 Invoice Items I (Item)

XML element representing a single item on the list of items.

3.7.1.66 Invoice Items I N (Item Name)

Name of the item.

Data type	String
Max length	50
Example	Wine 1.5L

Table 94

3.7.1.67 Invoice Items I C (Item Code)

Code of the item from the barcode or similar representation. It helps in identification of the product (item). For example, when a mobile phone is sold or imported, this field must contain IMEI number of that phone.

Data type	String
Max length	50
Example	978020137962

Table 95

3.7.1.68 Invoice Items I U (Item Unit of measure)

Unit of measure for specific item – piece, weight, length...

Data type	String
Max length	50
Example	Kg

Table 96

3.7.1.69 Invoice Items I Q (Item Quantity)

Amount or number (quantity) of items.

Data type	Double
Constraints	Must be positive number, at least 0.001.
Example	3.500 0.375

Table 97

3.7.1.70 Invoice Items I UPB (Item Unit Price Before VAT)

Price of one item before Value added tax is applied (unit price without VAT).

Data type	Decimal
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	3.50

Table 98

3.7.1.71 Invoice Items I UPA (Item Unit Price After VAT)

Price of one item after Value added tax is applied (unit price with VAT).

Data type	Decimal
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	3.85

Table 99

3.7.1.72 Invoice Items I R (Item Rebate)

Rebate percentage.

Data type	Decimal
Example	12 33.17

Table 100

3.7.1.73 Invoice Items I RR (Item Rebate Reducing base price)

Is rebate reducing base price?

Data type	Boolean
Values	true, false
Example	True

Table 101

3.7.1.74 Invoice Items I PB (Item Price Before VAT)

Price before VAT for the items in this group of items. This is not the unit price of the item. It is the unit price multiplied by the quantity of items.

Data type	Decimal
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	134.34

Table 102

3.7.1.75 Invoice Items I VR (Item VAT Rate)

Rate of value added tax expressed as percentage.

Data type	Decimal
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	3.50

Table 103

3.7.1.76   Invoice Items I EX (Item Exempt from VAT)

Exemption from VAT type

Data type	string	
Constraint	Enumeration, described in the table below.	
Example	TYPE_1	

Table 104

Enumeration values for the exempt from VAT types are listed in table below.

Value	Description
TYPE_1	Exempt type 1. Exempted on the basis of Article 51 of the VAT law
TYPE_2	Exempt type 2. Exempted on the basis of Articles 53 and 54 of the VAT law

Table 105

3.7.1.77   Invoice Items I VA (Item VAT Amount)

Amount of value added tax for the items in this group of items.

Data type	Decimal
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	3.50

Table 106

3.7.1.78   Invoice Items I PA (Item Price After applying VAT)

Price including VAT for the items in this group of items.

Data type	Decimal
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	3.50

Table 107

3.7.1.79   Invoice SameTaxes

XML element representing list of invoice items (goods or services) that are under same VAT rate/or exempted form VAT. All items of same VAT rate/exemption are grouped together.

3.7.1.80   Invoice SameTaxes SameTax

XML element representing several goods or services that are under same VAT rate/exemption.

3.7.1.81   Invoice SameTaxes SameTax NumOfItems

Number of items of same tax rate/exemption.

Data type	Integer
Pattern	([1-9][0-9]*)

Example	2

Table 108

3.7.1.82   Invoice SameTaxes SameTax PriceBefVAT

Total Price of items of the same tax rate/exemption before VAT

Data type	decimal
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	12.20

Table 109

3.7.1.83   Invoice SameTaxes SameTax VATRate

VAT rate applied on items with the same tax rate/exemption, expressed as percentage.

Data type	String
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	10.00

Table 110

3.7.1.84   Invoice SameTaxes SameTax ExemptFromVAT

Exemption from VAT type

Data type	string
Constraint	Enumeration, described in the table below.
Example	TYPE_1

Table 111

Enumeration values for the exempt from VAT types are listed in table below.

Value	Description
TYPE_1	Exempt type 1. Exempted on the basis of Article 51 of the VAT law
TYPE_2	Exempt type 2. Exempted on the basis of Articles 53 and 54 of the VAT law

Table 112

3.7.1.85   Invoice SameTaxes SameTax VATAmt

VAT amount for items from the same tax rate/exemption .

Data type	decimal
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	246.00

Table 113

3.7.1.86   Invoice ConsTaxes

XML element representing list of invoice items (goods or services) that are under consumption tax.

3.7.1.87 Invoice ConsTaxes ConsTax

XML element representing a single goods or services that are under same consumption tax.

3.7.1.88 Invoice ConsTaxesConsTax NumOfItems

Number of items under the consumption tax.

Data type	integer
Pattern	([1-9][0-9]*)
Example	2

Table 114

3.7.1.89 Invoice ConsTaxesConsTax PriceBefConsTax

Price of the item before consumption tax.

Data type	decimal
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	12.20

Table 115

3.7.1.90 Invoice ConsTaxesConsTax ConsTaxRate

Consumption tax rate.

Data type	String
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	10.00

Table 116

3.7.1.91 Invoice ConsTaxesConsTax ConsTaxAmt

Consumption tax amount.

Data type	decimal
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	246.00

Table 117

3.7.1.92 Invoice Fees

XML element representing list of invoice fee items.

3.7.1.93 Invoice Fees Fee

XML element representing a single fee.

3.7.1.94   Invoice Fees Fee Type

Type of the fee.

Data type	string
Constraint	Enumeration, described in the table below.
Example	PACK

Table 118

Enumeration values for fees are listed in table below.

Value	Description
PACK	Packaging fee
BOTTLE	Fee for the return of glass bottles
COMMISSION	Commission for currency exchange activities
OTHER	Other fees that are not listed here.

Table 119

3.7.1.95   Invoice Fees Fee Amt

The decimal amount of the fee.

Data type	decimal
Pattern	(-?[1-9][0-9]*|0)\.[0-9]{2}|0
Example	12.20

Table 120

3.7.1.96   Signature

XML element stores enveloped digital signature described in the chapter 4.3.1.

3.7.2   REGISTER INVOICE RESPONSE DATA MESSAGE

Name			Field type	Occurrence [Min, Max]	Description
RegisterInvoiceResponse			Element	[1, 1]	Root XML element representing register invoice response message.
	Id		Attribute	[1, 1]	Attribute used for signature creation and verification. Fixed value "Response".
	Version		Attribute	[1,1]	Attribute used to specify compliance with XSD schema. For this version fixed value is "2".
	Header		Element	[1, 1]	XML element representing generic message data about the response sent.
		UUID	Attribute	[1, 1]	UUID generated by a CIS for every register invoice response data message send to the TCR.
		RequestUUID	Attribute	[1, 1]	UUID of the request message for which this response message was sent.
		SendDateTime	Attribute	[1, 1]	Date and time of sending the register invoice response data message from a CIS to the TCR.
	FIC		Element	[1, 1]	CIS generated verification code that can be used to uniquely identify registered invoice.
	Signature		Element	[1, 1]	XML element with signature.

Table 121

3.7.2.1   Header

XML element representing header of the response data message.

3.7.2.2   Header UUID

Element generated by the CIS for every message sent to the TCR. It uniquely identifies the message sent to the TCR. UUID should be constructed according to the RFC4122 version 4.

Data type	string
Length	36 characters
Pattern	[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[1-5][0-9a-fA-F]{3}-[89abAB][0-9a-fA-F]{3}-[0-9a-fA-F]{12}
Example	58e0a7d7-eebc-41d8-9669-0800200c9a66 58E0A7D7-EEBC-41D8-9669-0800200C9A66

Table 122

3.7.2.3   Header RequestUUID

Element generated by the TCR and referenced by the CIS. It uniquely identifies the request message for which response message was sent to the TCR. UUID should be constructed according to the RFC4122 version 4.

Data type	string
Length	36 characters
Pattern	[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[1-5][0-9a-fA-F]{3}-[89abAB][0-9a-fA-F]{3}-[0-9a-fA-F]{12}
Example	58e0a7d7-eebc-41d8-9669-0800200c9a66 58E0A7D7-EEBC-41D8-9669-0800200C9A66

Table 123

3.7.2.4   Header SendDateTime

Element represents date and time of sending the response message to the TCR. Date and time should be in ISO 8601 format.

Data type	dateTime
Length	23 characters
Pattern	[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}[+-][0-9]{2}:[0-9]{2}
Example	2019-01-24T22:00:58+01:00 2019-01-24T22:00:58-01:00

Table 124

3.7.2.5   FIC

Element represents unique number generated by the CIS under which the requested invoice is registered.

Data type	string
Length	36 characters
Pattern	[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[1-5][0-9a-fA-F]{3}-[89abAB][0-9a-fA-F]{3}-[0-9a-fA-F]{12}
Example	58e0a7d7-eebc-41d8-9669-0800200c9a66 58E0A7D7-EEBC-41D8-9669-0800200C9A66

Table 125

3.7.2.6   Signature

XML element stores enveloped digital signature described in the chapter 4.3.1.

3.7.3   MANDATORY CONTROLS

Mandatory controls shall be performed on received invoices data messages in the CIP system in real time. When any of the critical controls return a failure, the invoice data message shall not be approved, and FIC shall not be issued.

Upon identifying a critical error, CIS will return an error data message containing the error's numeric code and its text description (see chapter 3.9). When errors which the system can interpret as a cyber-attack are identified, the system does not send any response to the client (the TCR).

The mandatory controls include the following:

Control Name	Control Description (Error if)	Error code
NUIS ID type	Invoice.TypeOfSelfIss exists and Inovice.Buyer.IDType is not NUIS. Invoice.TypeOfSelfIss doesn't exists and Inovice.Seller.IDType is not NUIS.	54
Issuer status invalid	Issuer is not in the Registry of taxpayers.	55
Issuer VAT status invalid	Invoice.IsIssuerInVAT status is different from the real issuer VAT status.	44
Invoice VAT status invalid	Invoice.IsIssuerInVAT is true and there is no VAT stated on the invoice. Invoice.IsIssuerInVAT is true and there is no Invoice.SameTaxes element. Invoice.IsIssuerInVAT is false, there is VAT stated on the invoice and Invoice.IsReverseCharge is false. Invoice.IsIssuerInVAT is false and there is Invoice.SameTaxes element. Invoice.TypeOfSelfIss exists and there is no Invoice.SameTaxes element. Invoice.IsReverseCharge is true there is no Invoice.SameTaxes element.	55
Cash invoice limit	Invoice.TypeOfInv is CASH, Invoice.PayMethods.PayMethod is BANKNOTE and Invoice.TotPrice is more than allowed amount.	40

Table 126

3.7.4   ERROR MESSAGE

Error message is defined in chapter 3.9.

3.7.5   EXAMPLE XML

3.7.5.1   Request XML

```xml
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <RegisterInvoiceRequest xmlns="https://eFiskalizimi.tatime.gov.al/FiscalizationService/schema"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#" Id="Request" Version="2">
            <Header SendDateTime="2019-12-05T14:30:13+01:00" UUID="8d216f9a-55bb-445a-be32-30137f11b964" IsSubseqDeliv="false"/>
            <Invoice BusinUnitCode="bb123bb123" IssueDateTime="2019-12-05T14:30:13+01:00" IIC="4AD5A215BEAF85B0416235736A6DACAB"
IICSignature="83D728C8E10BA04C430BE64CE98612B0256C0FE618C167F28BF62A0C0CB38C51824F152AB00510AE076508E53ACE4F877D25D51C7830F043E09BB15
00D3A0AEA233ECC6175A45FE58CBF53E517FD9EA1D06CBABC055EEE6B430A16560C96D3A27720A6E5C9BA5C8D18A7AE5C2A7F1D8E46B293F56D32847FCEE199D2AFDC
6E5BC1164BA974A6E29D6F40FBD8C51D40A99BC97DD6DB2AE9EC0582F2E74E9C7841AC5A854DE92B1D778A809CACCBBEF4DC325C852487BCF035AA2D54594DC6BDD85
9E250782CCCDD7CC89EE80A2FE1030AAAD615DA5D728322F8590D9F56E6DDE5975A738F304F56BB832996763624B72C77E97881D9C647B50709F20AFBFA0602"
InvNum="1/2019/cc123cc123" InvOrdNum="1" IsBadDebt="false" IsIssuerInVAT="true" IsReverseCharge="false" IsSimplifiedInv="false"
OperatorCode="oo123oo123" SoftCode="ss123ss123" TCRCode="cc123cc123" TotPrice="20.00" TotPriceWoVAT="16.00" TotVATAmt="4.00"
TypeOfInv="CASH">
                <PayMethods>
                    <PayMethod Amt="20.00" Type="BANKNOTE"/>
                </PayMethods>
                <Seller Address="Seller address" Country="ALB" IDNum="L91806031N" IDType="NUIS" Name="Seller name" Town="Seller
town"/>
                <Items>
                    <I C="501234567890" N="Item name" PA="20.00" PB="16.00" Q="1.0" R="0" RR="true" U="piece" UP="20.00" VA="4.00"
VR="25.00"/>
                </Items>
                <SameTaxes>
                    <SameTax NumOfItems="1" PriceBefVAT="16.00" VATAmt="4.00" VATRate="25.00"/>
                </SameTaxes>
            </Invoice>
```

```
                    <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/2001/04/xmldsig-more#rsa-sha256"/>
                            <Reference URI="#Request">
                                <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/2001/04/xmlenc#sha256"/>
                                <DigestValue>ECWby0FKaNQ6..........m8BPjyk6g=</DigestValue>
                            </Reference>
                        </SignedInfo>
                        <SignatureValue>LHFUg3xNi..........tjpeYNw==</SignatureValue>
                        <KeyInfo>
                            <X509Data>
                                <X509Certificate>MIIFYDCC..........Pifz0UlKJAanmqN3</X509Certificate>
                            </X509Data>
                        </KeyInfo>
                    </Signature>
                </RegisterInvoiceRequest>
        </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.7.5.2   Response XML

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:RegisterInvoiceResponse Id="Response" Version="2"
xmlns:ns2="https://eFiskalizimi.tatime.gov.al/FiscalizationService/schema" xmlns:ns3="http://www.w3.org/2000/09/xmldsig#">
            <ns2:Header RequestUUID="8d216f9a-55bb-445a-be32-30137f11b964" SendDateTime="2019-11-27T09:41:12.479+01:00"
UUID="f8bcb5ae-59fb-41ac-9011-f4db86bbce26"/>
            <ns2:FIC>a592e7ec-9517-4f02-8d54-ac965f679a8c</ns2:FIC>
            <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/2001/04/xmldsig-more#rsa-sha256"/>
                    <Reference URI="#Response">
                        <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/2001/04/xmlenc#sha256"/>
                        <DigestValue>WYXOkHAd..........SLOIbwDdHCQk=</DigestValue>
                    </Reference>
                </SignedInfo>
                <SignatureValue>Bnr1bB8Ic..........bfAjdVUpA==</SignatureValue>
                <KeyInfo>
                    <X509Data>
                        <X509Certificate>MIIFRjCCB..........gpNT2r23YQ==</X509Certificate>
                    </X509Data>
                </KeyInfo>
            </Signature>
        </ns2:RegisterInvoiceResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.7.6   DATA EXPORT AND UPLOAD

When the taxpayer is operating in the area where there is no internet connection, he can use the alternative way of invoice registration. Invoice registration requests should be exported into files in a special format. These files can then be uploaded on Central invoice platform which will register invoices. Responses from the registration service will be generated and the taxpayer will download files with responses and bring them back to the TCR for import.

Instead of using the Central invoice platform web application, the taxpayer can bring the files on USB flash drive to the local Tax administration office where the officer will do the same process as in the Central invoice platform.

What taxpayer does in that case is that he uses TCR to creates a file for each of those invoices, and files should be named in `<yyyyMMddHHmmSS>_<TCRCode>_<IIC>_request.xml` format.

When those .xml files (WS messages) are created, they will be created in the way that only main part of the XML, the one that contains data, will be preserved. It will omit WS header and WS envelope from the message. The file should contain only content of envelope body element. Content of the body element must not be altered to preserve valid signature.

Then each of those files should be saved in a zipped (archived) folder, named `<yyyyMMddHHmmSS>_<TCRCode>_request.zip`. Size of the ZIP archive should not exceed 15 MB.

After that the taxpayer transfers that archive to the USB flash drive or other transferable media and takes it to the place where there is an active internet connection. After logging in to Central invoice platform using the certificate, taxpayer imports ZIP archive with invoices represented as XML files. Central invoice platform should extract XML files from the ZIP archive, form a web service message request and send it to fiscalization service. The fiscalization service will make a response, which the Central invoice platform will transform into XML file by extracting the content of SOAP envelope body element. XML file representing the response will be named `<yyyyMMddHHmmSS>_<TCRCode>_<IIC>_response.xml`. XML files will be compressed in a ZIP archive with corresponding name `<yyyyMMddHHmmSS>_<TCRCode>_response.zip` where `<yyyyMMddHHmmSS>` is the same as in the ZIP archive with request XML files. That means that each request in XML file named `<yyyyMMddHHmmSS>_<TCRCode>_<IIC>_request.xml` will be paired with response in XML file `<yyyyMMddHHmmSS>_<TCRCode>_<IIC>_response.xml,` and each archive containing XML request named `<yyyyMMddHHmmSS>_<TCRCode>_request.zip` will be paired with archive named `<yyyyMMddHHmmSS>_<TCRCode>_response.zip` containing XML responses.

If there are errors, they will also be stored inside XML responses and the Central Invoice Platform will notify the user about it.

Application should log to application log name of the incoming ZIP archive, names of the extracted request files, name of the generated response files, name of the generated ZIP archive, TIN of the user.

3.7.6.1 XML example

As mentioned in previous chapter, when there is upload of the invoices, the XML file will be created in a way that only main part of it – the content of the body (without SOAP body element) - the one containing the message – will be preserved, while other parts will be omitted. Here is the example of it (SOAP envelope elements are removed and in the example they are strikethrough):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <RegisterInvoiceRequest xmlns="https://eFiskalizimi.tatime.gov.al/FiscalizationService/schema"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#" Id="Request" Version="2">
            <Header SendDateTime="2019-12-05T14:30:13+01:00" UUID="8d216f9a-55bb-445a-be32-30137f11b964" IsSubseqDeliv="false"/>
            <Invoice BusinUnitCode="bb123bb123" IssueDateTime="2019-12-05T14:30:13+01:00" IIC="4AD5A215BEAF85B0416235736A6DACAB"
IICSignature="83D728C8E10BA04C430BE64CE98612B0256C0FE618C167F28BF62A0C0CB38C51824F152AB00510AE076508E53ACE4F877D25D51C7830F043E09BB15
00D3A0AEA233ECC6175A45FE58CBF53E517FD9EA1D06CBABC055EEE6B430A16560C96D3A27720A6E5C9BA5C8D18A7AE5C2A7F1D8E46B293F56D32847FCEE199D2AFDC
6E5BC1164BA974A6E29D6F40FBD8C51D40A99BC97DD6DB2AE9EC0582F2E74E9C7841AC5A854DE92B1D778A809CACCBBEF4DC325C852487BCF035AA2D54594DC6BDD85
9E250782CCCDD7CC89EE80A2FE1030AAAD615DA5D728322F8590D9F56E6DDE5975A738F304F56BB832996763624B72C77E97881D9C647B50709F20AFBFA0602"
InvNum="1/2019/cc123cc123" InvOrdNum="1" IsBadDebt="false" IsIssuerInVAT="true" IsReverseCharge="false" IsSimplifiedInv="false"
OperatorCode="oo123oo123" SoftCode="ss123ss123" TCRCode="cc123cc123" TotPrice="20.00" TotPriceWoVAT="16.00" TotVATAmt="4.00"
TypeOfInv="CASH">
                <PayMethods>
                    <PayMethod Amt="20.00" Type="BANKNOTE"/>
                </PayMethods>
                <Seller Address="Seller address" Country="ALB" IDNum="L91806031N" IDType="NUIS" Name="Seller name" Town="Seller
town"/>
                <Items>
                    <I C="501234567890" N="Item name" PA="20.00" PB="16.00" Q="1.0" R="0" RR="true" U="piece" UP="20.00" VA="4.00"
VR="25.00"/>
                </Items>
                <SameTaxes>
                    <SameTax NumOfItems="1" PriceBefVAT="16.00" VATAmt="4.00" VATRate="25.00"/>
                </SameTaxes>
            </Invoice>
            <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/2001/04/xmldsig-more#rsa-sha256"/>
                    <Reference URI="#Request">
                        <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/2001/04/xmlenc#sha256"/>
                        <DigestValue>ECWby0FKaNQ6..........m8BPjyk6g=</DigestValue>
                    </Reference>
                </SignedInfo>
                <SignatureValue>LHFUg3xNi..........tjpeYNw==</SignatureValue>
                <KeyInfo>
                    <X509Data>
                        <X509Certificate>MIIFYDCC..........Pifz0UlKJAanmqN3</X509Certificate>
                    </X509Data>
                </KeyInfo>
```

```
        </Signature>
      </RegisterInvoiceRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.7.7   QR CODE GENERATION

Each invoice should contain a QR-code which has a URL inside. The URL leads to the web application "Invoice check" and displays information about the invoice if the invoice is successfully registered or instructs user to report an issue if it is not registered within the required timeframe.

QR-code contains a URL with special query string which identifies the invoice. First part of the URL is fixed and contains the protocol, hostname and path followed by the query parameters.

Fixed part of the URL:

- Test environment:
  - https://efiskalizimi-app-test.tatime.gov.al/invoice-check/#/verify
- Production environment:
  - https://efiskalizimi-app.tatime.gov.al/invoice-check/#/verify

Query parameters:

Parameter name	Description	Sample value
iic	Invoice IIC (issuer's identification number)	EA26D5BE7F45827026108F825A8A512B
tin	Taxpayer identification number (issuer's TIN)	L91806031N
crtd	Date and time when the invoice was created. Value is displayed in special format "yyyy-mm-ddThh:mi:ss" where *yyyy*, *mm* is month, *dd* is day of the month, *T* is fixed value, *hh* are hours in 24-hours format, *mi* are minutes, *ss* are seconds.	2019-09-26T13:50:13
ord	Invoice ordinal number	6
bu	Code of the business unit (premise)	bg517kw842
cr	Code of the TCR	xb131ap287
sw	Code of the software installed on the TCR	gz434bv927
prc	Total price of the invoice	2354.84

Table 127 - QR code URL query parameters

QR code must be created with at least M error correction level that guaranties that at least 15% of character can be restored.

3.7.7.1   QR code example

Example URL encoded in a QR-code with sample values from the Table 127 is:

https://efiskalizimi-app-test.tatime.gov.al/invoice-check/#/verify?iic=EA26D5BE7F45827026108F825A8A512B&tin=L91806031N&crtd=2019-09-26T13:50:13+01:00&ord=6&bu=bg517kw842&cr=xb131ap287&sw=gz434bv927&prc=199.00

Example QR-code which is formed from the URL above is:

Figure 7 - QR code sample

3.8   WAREHOUSE TRANSFER NOTE

To be defined by 01.04.2020.

3.9   ERROR MESSAGES

In case of an error in the process of the request message, an error message is sent as a reply by CIS. Error messages share the same general format which is based on SOAP fault message version 1.1 and extended with the code XML element which represents numeric error code.

3.9.1   XML FORMAT

Name	Occurrence [Min, Max]	Description
fault	[1, 1]	XML element representing error message.
faultCode	[1, 1]	XML element representing class of errors.
faultString	[1, 1]	XML element where the error explanation is written.
detail	[1, 1]	XML element that carries error messages. It can contain multiple child elements.
responseUUID	[1, 1]	XML element that specifies UUID of this error response.
requestUUID	[0, 1]	XML element that specifies UUID of the request for which error occurred if available.
code	[1, 1]	XML element that describes the error with a numeric code. List of codes can be found in the chapter 3.9.3.

Table 128

3.9.1.1   Header

This is an XML root element representing the header of the error message.

3.9.1.2   Header UUID

This is an attribute that uniquely describes the message and gives it the unique identification.

Data type	string
Length	36 characters

Pattern	[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[1-5][0-9a-fA-F]{3}-[89abAB][0-9a-fA-F]{3}-[0-9a-fA-F]{12}
Example	58e0a7d7-eebc-41d8-9669-0800200c9a66 58E0A7D7-EEBC-41D8-9669-0800200C9A66

Table 129

3.9.1.3 Fault

This is an XML element that will appear only if error happens.

3.9.1.4 FaultCode

This is an XML element that represents origin of error that occurred.

Data type	string
Constraint	Enumeration, described in the table below.
Example	Client

Table 130

Enumeration values for the method of payment are listed in table below.

Value	Description
Client	Received message was incorrectly formed or contained incorrect information.
Server	There was a problem with the server, so the message could not proceed.

Table 131

3.9.1.5 FaultString

This is an XML element that contains textual explanation for error that occurred.

Data type	string
Length	Undefined
Example	Validation failed with digest wrong.

Table 132

3.9.1.6 Detail

This is an XML element that carries numeric error code.

3.9.1.7 Code

This is a Detail's child element, that describes the numeric error code. Numeric error codes are listed in the chapter 3.9.2.

Data type	int
Length	3
Pattern	[1-9][0-9]{0,2}
Example	21

Table 133

3.9.1.8 ResponseUUID

This is a Detail's child element, that specifies UUID of this error message.

Data type	string
Length	36 characters
Pattern	[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[1-5][0-9a-fA-F]{3}-[89abAB][0-9a-fA-F]{3}-[0-9a-fA-F]{12}
Example	58e0a7d7-eebc-41d8-9669-0800200c9a66 58E0A7D7-EEBC-41D8-9669-0800200C9A66

Table 134

3.9.1.9 RequestUUID

This is a Detail's child element, that specifies UUID of the request message that generated an error if available.

Data type	string
Length	36 characters
Pattern	[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[1-5][0-9a-fA-F]{3}-[89abAB][0-9a-fA-F]{3}-[0-9a-fA-F]{12}
Example	58e0a7d7-eebc-41d8-9669-0800200c9a66 58E0A7D7-EEBC-41D8-9669-0800200C9A66

Table 135

3.9.2 ERROR CODES

Following table lists the error codes that a fiscalization service can return to the invoice issuer. It's not an exhaustive list.

Error code	Error origin	Error description
0	Client	Exception occurred while extracting received XML message during size check.
1	Client	Received XML message exceed allowed size.
2	Client	Client time differs from a server's time by more than allowed time in minutes
10	Client	Exception occurred while extracting received XML message during XML validation against the XSD.
11	Client	Received XML message failed XSD validation.
20	Client	Exception occurred while extracting received XML message during signature check.
21	Client	Received XML message missing Signature XML element.
22	Client	Received XML message missing Request XML element.
23	Client	Exception occurred while extracting Signature XML element during signature check.
24	Client	Provided more than one Signature XML element.
25	Client	Signed wrong XML element.
26	Client	Wrong signature method specified.
27	Client	Wrong canonicalization method specified.
28	Client	Wrong digest method specified.
29	Client	Cryptographic signature wrong.
30	Client	Digest calculation wrong.
31	Client	Overall signature wrong.
32	Client	There are more keyInfo elements than needed.
33	Client	Certificate provided is not of X509 type of certificate.
34	Client	Certificate provided is not valid.
35	Client	Certificate is not issued by NAIS.
36	Client	Certificate has expired.
37	Client	Compare the NIPT in XML with the NIPT in the certificate
38	Client	Certificate status revoked
39	Client	Certificate status unknown
40	Client	Invoice amount too large to be paid by cash.
41	Client	Business unit code doesn't reference active business unit (premise) of the taxpayer.
42	Client	Software code doesn't references active software.
43	Client	Maintainer code doesn't references active maintainer.
44	Client	Issuer VAT status doesn't correspond to the IsIssuerInVAT attribute.
45	Client	ValidFrom cannot be in the past.

46	Client	ValidTo cannot be in the past.
47	Client	ValidTo cannot be before ValidFrom.
48	Client	Active TCR ValidFrom cannot be changed.
49	Client	Change date and time differences from CISs time more than allowed time in minutes.
50	Client	Cash amount cannot be negative for INITIAL operation.
51	Client	Cash amount cannot be zero for INOUT operation.
52	Client	Issuer doesn't reference active taxpayer in the Registry of taxpayers.
53	Client	TCR code doesn't reference registered or active TCR or the TCR doesn't belongs to the referenced issuer.
54	Client	Issuer ID type must be NUIS.
55	Client	Issuer not active in the Registry of taxpayers.
9xx	Server	Internal server exceptions

Table 136

3.9.3   EXAMPLE XML

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <SOAP-ENV:Fault>
            <faultcode>SOAP-ENV:CLIENT</faultcode>
            <faultstring xml:lang="en">Validation failed with digest wrong.</faultstring>
            <detail>
                <code>30</code>
                <requestUUID>78dde160-2b33-40e4-98fa-f6a2c34475a3</requestUUID>
                <responseUUID>9416dcca-499a-4724-933d-40d115ea4fc7</requestUUID>
            </detail>
        </SOAP-ENV:Fault>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4. Security

Following chapter describes security principles used by fiscalization service and parties that communicate with it.

4.1 DIGITAL CERTIFICATES

For the purpose of the securing fiscalization service and identifying clients that consumes the service, three types of certificates shall be issued:

- Certificate for transport security with common name *.tatime.gov.al issued to the service.
  Used to secure transport between the service and an issuer.
  Issued by public CA different from NAIS CA.
- Certificate for message security issued to the service.
  Used to digitally sign response data messages.
  Issued by NAIS CA.
- Certificate for message security issued to an issuer.
  Used to digitally sign request data messages and IIC data elements.
  Issued by NAIS CA.

A digital certificate for fiscalization purposes is issued by the competent authority for issuing digital certificates (CAs). In the case of the Republic of Albania it is NAIS. A detailed description of the submission of a request for issuing a digital certificate and obtaining a certificate is (or will have to be) defined in the instructions of NAIS and is not the subject of this documentation. Certificates issued by NAIS will have following certificate authorities:

- Test environment
  - NAIS Root Certification Authority (Self signed certificate)
    - NAIS Certification Authority
      - NAIS Class 3 Certification Authority
- Production environment
  - NAIS Root Certification Authority (Self signed certificate)
    - NAIS Certification Authority
      - NAIS Class 1 Certification Authority

4.2 TRANSPORT SECURITY

To ensure data security and integrity of the communication between the issuer and the service, service is using One-way TLS, specifically protocol version TLS V1.2. Service presents a certificate to client issued by the public CA different from NAIS CA, and with common name *.tatime.gov.al.

4.3 MESSAGE SECURITY

To ensure unambiguous identification of the taxpayer and to provide unchanged content, each request data message and IIC data element is digitally signed with a private key that is unique pair with the valid taxpayer certificate. Response data messages form the CIS are digitally signed with a private key that is unique pair with the valid CIS certificate.

In most cases, the private key used to digitally sign request data message and IIC data is the same. An exception under this rule is possible if the certificate used at the time of initial creation of the request is no longer valid at the time of resending the request. In that case, a private key from valid corresponding certificate must be used to digitally sign request data message, but not the IIC data element, which remains the same.

Request and response data messages are digitally signed according to the XML Signature Syntax and Processing standard edition 1.1 available at https://www.w3.org/TR/xmldsig-core/. Additional description is available in the chapter 4.3.1.

IIC data element is created and digitally signed according to the custom cryptographic algorithm described in the chapter 4.3.2.

4.3.1 REQUEST AND RESPONSE DATA MESSASGE SIGNING

Every request and response data message described in the chapter 3, must contain signature XML element. That element is generated according to XML Signature Syntax and Processing standard edition 1.1 available at https://www.w3.org/TR/xmldsig-core/.

Element to signed is a first and only element inside soap envelope body XML element, with Id equals to Request or Response, depending on the message direction.

XML digital signature element is created with following options:

- Signature type: Enveloped, http://www.w3.org/2000/09/xmldsig#enveloped-signature
- Canonicalization method: C14 Exclusive, http://www.w3.org/2001/10/xml-exc-c14n#
- Digest method: SHA256, http://www.w3.org/2001/04/xmlenc#sha256
- Signing method: RSA SHA256, http://www.w3.org/2001/04/xmldsig-more#rsa-sha256

4.3.2 IIC DATA ELEMENT

IIC, Issuer's Invoice Code, is an alphanumeric security code generated by the issuer which uniquely matches issued invoice with an issuer. It is generated by concatenating specific parameters of the invoice and signed with a private key of the issuer.

IIC has two purposes:

1. To protect issuer from malicious third party because only the issuer that generated IIC can regenerate it by supplying the algorithm with the same parameters and using the same private key.
2. To verify that issued invoice is registered in the CIS.

At the tax administration's request, the taxpayer, based on the same input parameters, must create an IIC equal to that of the invoice.

IIC is generated using the following algorithm steps:

1. Concatenate parameters
2. Calculate digital signature with SHA256, RSA and RSASSA-PKCS-v1_5 padding
3. Calculate digest

4.3.2.1 Concatenate parameters

IIC is generated by concatenating following parameters of the invoice:

- Issuer NUIS (Chapter 3.7.1.51)
- Date and time created (Chapter 3.7.1.8)
- Invoice number (Chapter 3.7.1.10)
- Business unit code (Chapter 3.7.1.21)
- TCR code (Chapter 3.7.1.22)
- Software code (Chapter 3.7.1.22)

- Total price (Chapter 3.7.1.19)

Before concatenation, all parameters must be converted into UTF-8 encoding. Parameters are concatenated with pipe character UTF-8 with decimal code 124.

For example, for parameters:

- Issuer NUIS: I12345678I
- Date and time created: 2019-06-12T17:05:43+02:00
- Invoice number: 9952
- Business unit code: bb123bb123
- TCR code: cc123cc123
- Software code: ss123ss123
- Total price: 99.01

Resulted concatenated value is:
I12345678I|2019-06-12T17:05:43+02:00|9952|bb123bb123|cc123cc123|ss123ss123|99.01

4.3.2.2 Calculate digital signature

After the concatenation, resulting value is hashed with SHA256 algorithm and then signed with RSA algorithm and issuer's private key.

For example, for values:

- Concatenated value:
  I12345678I|2019-06-12T17:05:43+02:00|9952|bb123bb123|cc123cc123|ss123ss123|99.01
- PEM encoded private key:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEA6zOR5ItNYHJNVMxljZtd/KQUyGIozbnIJ8IWqcEesktRV5FF
HviQZsx2DpyeVQTu/Kel9Xh+Z6OZ6t5sADzfYnkwCrsb0FhT+01m2PIHaIUZhVtc
ppn0gxNWfgzW4sTvTyrYk6O1Kxymsx/rck/WRQB1mp68au8mgGMzGukHfL7Wk4jO
U5VD3HlStBx1MjVW+soN5GUL/rWGaYun6Zsn9aYYEujbOhKvKDy8nOtNIS69dqdd
piZAkvdh9sYdF1ElgXZhdmZsGURMm6OcePUPZO/HFKq7RlK6vIxXVI6l9O6tWt+G
uhul8e0x2VTwbTdpwG4FpdfUTqUDK6cswHOhTQIDAQABAoIBAQCqBWJuUqDBmn76
ULMMlYZwjfAUFpkmdikRTIVzew4EltubMIFF7Sr9lMm2sFLoZKOZ8lrOwqalpqcq
GFT8KwTUO4SWDUIC7wbuf7pcE0F1tdmIBE5KhLozUnRQtFlWHkRb9z4OI+Zf3ttG
W0mpHbtnr/hTqHHN30j2wD7+MfvemPbcAvu9JLCYUzUZ06qxUwAjyFgsW7YyLa0a
qFB0QOYc6RsLvoSFXW0M5ghdtgoZvl+ayt4fgz1L3FjAMuXoLEX/778VA92/NZ0Q
mzQdKTT6B4Pm5s8XrY9OhLlsYqKuyR/aoSHC/anSLw0yJ/5Gis2gmCwo3a7+PEYy
LUN7C0yFAoGBAPhgYufTkdod5PqG/SCEE2i6pjk0ZnuIUu9f2cmhxnvyChlig2wk
oDWUSGuXwItNF+X7j3XoZz8FNJcriK7KP2UPDOWP0ZvxZgZEcmwut27x1vVjzjCG
sl0w5fFO363hhtX35Jq2lVZGbN1LpIoEZgCeS/nBs+9DcRjDoXliKWfHAoGBAPJr
qSWLVO3gIG1wikXBWCYZUTSzsO6NWfxcWPHKTnKVrOifBTK23zuZ6ggluNqLz/Ae
64ZwssMoIVIyXE01XMPP8io4QidyVEd2n70pjrVcUVYyr9IwKmchmNBfKFMof05f
NV29P1Am1Jqv2EQi5jE/BbBu9kLifs2YyGBAn/ZLAoGAVsLsqciZAVVCAFWZJHue
gA37NK5eQja7qcyUuj9dozxIVNe5ytP8dtrmdVccNkzm1TqLwYc+UaBS35+gblZN
0NJyEdqsQMoRdo0AX1PuVb369ds4UnEq6yzClgmUTxwhyqp+W6D+B5YwPxlGT8P7
kam6JnOIlEK9xgXIaStmBU8CgYB6RwXVszcOmYuhyC9mygSNix2j6LNpUJFAMtCG
fZYeRBMobvWvRADLznH21Bgu3HDxXJdOg9AXkklkbZSTOURmXKB43VG5Ffke5t3i
C3E5V6yLPxvieHsa9B5hlG4BrB6yyGFhvBCQfFWnBOWgUL4tvu0+tmmvCRIO4G7J
5i8JiwKBgQCQHTfRrGaEsq1BG7zPOQSqo9q5cxL8WzYd0sTs3FDcwCtHqxBEQ3rr
O/l+HvRa+y6ZEH6q4pREewTIymfv9tmGxVe3f8zrKGR5litvN6OnZuWJdq57Y1lN
J1sdpMxTtxQQmexsADif+QByCvdeFKE5C3veMLdgS5I6HTMN9k5laA==
-----END RSA PRIVATE KEY-----
```

Resulting signature value is:

404ADDB017B2DE49B0A51340A991130E670F08BC2BE854EEAAE9C3F41A2C98E1D70545690F0EFBD13511A38DB1E3
6E086DC253C3519E7DAF896A418BFAFCCE9836B0759B2E84713B25C39C040E35608AC85141A65D623454BAF4D0E04
D69A8D77505879C1DB9552542309A110B8CB2B9885C2236C3C6D65E695DFA4CA7D6258BD9EB0749A9EE09DA237C
4E1B8EE39C3CAD3E32A21F807DA0908192DADA3F9D55C4FEB3C100F97D5AA81CFE157E1A90059111E6DCD2F2AD3D
B9AAA202D084144E60ADED38988C384012967EF47B548135804EF2F4542DD0971E11AA392F048836D1C7DF9014F50
7B79258FA9B43AA14E32196D6127FD8154C24CE0CB374677D20

4.3.2.3　　　Calculate digest

After the signing, resulting value is hashed with a MD5 algorithm.

For example, for a value:

- Signature value:
  404ADDB017B2DE49B0A51340A991130E670F08BC2BE854EEAAE9C3F41A2C98E1D70545690F0EFBD13511A
  38DB1E36E086DC253C3519E7DAF896A418BFAFCCE9836B0759B2E84713B25C39C040E35608AC85141A65D6
  23454BAF4D0E04D69A8D77505879C1DB9552542309A110B8CB2B9885C2236C3C6D65E695DFA4CA7D6258B
  D9EB0749A9EE09DA237C4E1B8EE39C3CAD3E32A21F807DA0908192DADA3F9D55C4FEB3C100F97D5AA81CF
  E157E1A90059111E6DCD2F2AD3DB9AAA202D084144E60ADED38988C384012967EF47B548135804EF2F4542
  DD0971E11AA392F048836D1C7DF9014F507B79258FA9B43AA14E32196D6127FD8154C24CE0CB374677D20

MD5 digest value is: D04C13B4063D63A13B5D822A90178A7C

5. ANNEX - Code examples

This chapter covers the code examples for specific actions.

5.1   IIC GENERATION CODE

5.1.1   JAVA EXAMPLE

This is the example for the generation of the IIC in Java language. Variables are hardcoded as this is just an example.

```java
import java.io.FileInputStream;
import java.security.*;

import javax.xml.bind.DatatypeConverter;

public class SampleGenerateIIC {

    private static final String KEYSTORE_LOCATION = "***.p12";
    private static final String KEYSTORE_TYPE = "PKCS12";
    private static final String KEYSTORE_PASS = "***";
    private static final String KEYSTORE_KEY_ALIAS = "***";

    public static void main(String[] args) {

        String iicInput = "";

        // issuerNuis
        iicInput += "I12345678I";
        // dateTimeCreated
        iicInput += "|2019-06-12T17:05:43+02:00";
        // invoiceNumber
        iicInput += "|9952";
        // busiUnitCode
        iicInput += "|bb123bb123";
        // tcrCode
        iicInput += "|cc123cc123";
        // softCode
        iicInput += "|ss123ss123";
        // totalPrice
        iicInput += "|99.01";

        try (FileInputStream fileInputStream = new FileInputStream(KEYSTORE_LOCATION)) {
            // Load a private from a key store
            KeyStore keyStore = KeyStore.getInstance(KEYSTORE_TYPE);
            keyStore.load(fileInputStream, KEYSTORE_PASS.toCharArray());
            Key privateKey = keyStore.getKey(KEYSTORE_KEY_ALIAS, KEYSTORE_PASS.toCharArray());

            // Create IIC signature according to RSASSA-PKCS-v1_5
            Signature signature = Signature.getInstance("SHA256withRSA");
            signature.initSign((PrivateKey)privateKey);
            signature.update(iicInput.getBytes());
            byte[] iicSignature = signature.sign();
            String iicSignatureString = DatatypeConverter.printHexBinary(iicSignature).toUpperCase();
            System.out.println ("The IIC signature is: " + iicSignatureString);

            // Hash IIC signature with MD5 to create IIC
            MessageDigest md = MessageDigest.getInstance("MD5");
            byte[] iic = md.digest(iicSignature);
            String iicString = DatatypeConverter.printHexBinary(iic).toUpperCase();
            System.out.println ("The IIC is: " + iicString);
        } catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```

5.1.2 C# EXAMPLE

This is the example for the generation of the IIC in C# language. Variables are hardcoded as this is just an example.

```csharp
using System;
using System.Security.Cryptography;
using System.Security.Cryptography.X509Certificates;
using System.Text;

namespace FiscalizationSigningUtilityDotNet
{
    class SampleGenerateIIC
    {
        private const String KEYSTORE_LOCATION = "***.p12";
        private const String KEYSTORE_PASS = "***";

        public static void Main(string[] args)
        {
            String iicInput = "";

            // issuerNuis
            iicInput += "I12345678I";
            // dateTimeCreated
            iicInput += "|2019-06-12T17:05:43+02:00";
            // invoiceNumber
            iicInput += "|9952";
            // busiUnitCode
            iicInput += "|bb123bb123";
            // tcrCode
            iicInput += "|cc123cc123";
            // softCode
            iicInput += "|ss123ss123";
            // totalPrice
            iicInput += "|99.01";

            using (X509Certificate2 keyStore = new X509Certificate2(KEYSTORE_LOCATION, KEYSTORE_PASS))
            {
                try
                {
                    // Load a private from a key store
                    RSA privateKey = keyStore.GetRSAPrivateKey();

                    // Create IIC signature according to RSASSA-PKCS-v1_5
                    byte[] iicSignature = privateKey.SignData(Encoding.ASCII.GetBytes(iicInput), HashAlgorithmName.SHA256,
RSASignaturePadding.Pkcs1);
                    string iicSignatureString = BitConverter.ToString(iicSignature).Replace("-", string.Empty);
                    Console.WriteLine("The IIC signature is: " + iicSignatureString);

                    // Hash IIC signature with MD5 to create IIC
                    byte[] iic = ((HashAlgorithm)CryptoConfig.CreateFromName("MD5")).ComputeHash(iicSignature);
                    string iicString = BitConverter.ToString(iic).Replace("-", string.Empty);
                    Console.WriteLine("The IIC is: " + iicString);
                }
                catch (Exception ex)
                {
                    Console.WriteLine(ex.Message);
                }
            }
        }
    }
}
```

5.2 SIGNATURE GENERATION CODE

5.2.1 JAVA EXAMPLE

This is the example for the generation of the signature in Java language. Variables are hardcoded as this is just an example.

```java
import java.io.*;
import java.security.*;
import java.security.cert.X509Certificate;
import java.util.*;

import javax.xml.crypto.dsig.*;
import javax.xml.crypto.dsig.keyinfo.*;
import javax.xml.crypto.dsig.spec.*;
import javax.xml.crypto.dsig.dom.DOMSignContext;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.*;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

import org.w3c.dom.*;
import org.xml.sax.InputSource;

public class SampleGenerateSignature {

    private static final XMLSignatureFactory xmlSigFactory = XMLSignatureFactory.getInstance("DOM");
```

```java
    public static final String XML_SCHEMA_NS = "https://eFiskalizimi.tatime.gov.al/FiscalizationService/schema";
    public static final String XML_REQUEST_ELEMENT = "RegisterInvoiceRequest";
    public static final String XML_REQUEST_ID = "Request";
    public static final String XML_SIG_METHOD = "http://www.w3.org/2001/04/xmldsig-more#rsa-sha256";

    private static final String REQUEST_TO_SIGN =
        "<RegisterInvoiceRequest " +
        "    xmlns=\"https://eFiskalizimi.tatime.gov.al/FiscalizationService/schema\" " +
        "    xmlns:ns2=\"http://www.w3.org/2000/09/xmldsig#\" " +
        "    Id=\"Request\" " +
        "    Version=\"2\">\r\n" +
        "  <Header>...</Header>\r\n" +
        "  <Invoice>...</Invoice>\r\n" +
        "</RegisterInvoiceRequest>";
    private static final String KEYSTORE_LOCATION = "***.p12";
    private static final String KEYSTORE_TYPE = "PKCS12";
    private static final String KEYSTORE_PASS = "***";
    private static final String KEYSTORE_KEY_ALIAS = "***";

    public static void main(String[] args) {

        try (FileInputStream fileInputStream = new FileInputStream(KEYSTORE_LOCATION)) {
            // Load a private from a key store
            KeyStore keyStore = KeyStore.getInstance(KEYSTORE_TYPE);
            keyStore.load(fileInputStream, KEYSTORE_PASS.toCharArray());
            Key privateKey = keyStore.getKey(KEYSTORE_KEY_ALIAS, KEYSTORE_PASS.toCharArray());
            X509Certificate certificate = (X509Certificate)keyStore.getCertificate(KEYSTORE_KEY_ALIAS);

            // Load XML to DOC
            DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
            docFactory.setNamespaceAware(true);
            DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
            Document doc = docBuilder.parse(new InputSource(new StringReader(REQUEST_TO_SIGN)));

            // Find root request element
            NodeList nodeToSignList = doc.getElementsByTagNameNS(XML_SCHEMA_NS, XML_REQUEST_ELEMENT);
            if (nodeToSignList.getLength() == 0) {
                    throw new Exception(String.format("XML element %s not found", XML_REQUEST_ELEMENT));
            }
            Node nodeToSign = nodeToSignList.item(0);

            // Create transform list
            List<Transform> transformList = new ArrayList<>();
            transformList.add(xmlSigFactory.newTransform(Transform.ENVELOPED, (TransformParameterSpec) null));
            transformList.add(xmlSigFactory.newTransform(CanonicalizationMethod.EXCLUSIVE, (C14NMethodParameterSpec) null));

            // Create digest reference element
            Reference ref = xmlSigFactory.newReference(
                    "#" + XML_REQUEST_ID,
                    xmlSigFactory.newDigestMethod(DigestMethod.SHA256, null),
                    transformList,
                    null,
                    null);

            // Create signature method
            SignatureMethod signatureMethod = xmlSigFactory.newSignatureMethod(XML_SIG_METHOD, (SignatureMethodParameterSpec) null);

            // Create signed info element
            SignedInfo signedInfo = xmlSigFactory.newSignedInfo(
                    xmlSigFactory.newCanonicalizationMethod(CanonicalizationMethod.EXCLUSIVE, (C14NMethodParameterSpec) null),
                    signatureMethod,
                    Collections.singletonList(ref));

            // Add certificate
            List<X509Certificate> certificateList = new ArrayList<>();
            certificateList.add(certificate);

            // Create key info element
            KeyInfoFactory keyInfoFactory = xmlSigFactory.getKeyInfoFactory();
            X509Data x509Data = keyInfoFactory.newX509Data(certificateList);
            KeyInfo keyInfo = keyInfoFactory.newKeyInfo(Collections.singletonList(x509Data));

            // Create context for signing
            DOMSignContext dsc = new DOMSignContext(privateKey, nodeToSign);
            dsc.setIdAttributeNS((Element)nodeToSign, null, "Id");

            // Sign document
            XMLSignature signature = xmlSigFactory.newXMLSignature(signedInfo, keyInfo);
            signature.sign(dsc);

            // Output to string
            TransformerFactory transformFactory = TransformerFactory.newInstance();
            Transformer transformer = transformFactory.newTransformer();
            transformer.setOutputProperty(OutputKeys.OMIT_XML_DECLARATION, "yes");
            StringWriter sw = new StringWriter();
            StreamResult streamRes = new StreamResult(sw);
            transformer.transform(new DOMSource(doc), streamRes);
            System.out.println ("Signed document is: " + sw.toString());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

5.2.2 C# EXAMPLE

This is the example for the generation of the signature in .NET C# language. Variables are hardcoded as this is just an example.

```csharp
using System;
using System.IO;
using System.Security.Cryptography;
using System.Security.Cryptography.X509Certificates;
using System.Security.Cryptography.Xml;
using System.Xml;

namespace FiscalizationSigningUtilityDotNet
{
    class SampleGenerateSignature
    {
        public const String XML_SCHEMA_NS = "https://eFiskalizimi.tatime.gov.al/FiscalizationService/schema";
        public const String XML_REQUEST_ID = "Request";
        public const String XML_SIG_METHOD = "http://www.w3.org/2001/04/xmldsig-more#rsa-sha256";
        public const String XML_DIG_METHOD = "http://www.w3.org/2001/04/xmlenc#sha256";

        private const String REQUEST_TO_SIGN =
            "<RegisterInvoiceRequest " +
            "       xmlns=\"https://eFiskalizimi.tatime.gov.al/FiscalizationService/schema\" " +
            "       xmlns:ns2=\"http://www.w3.org/2000/09/xmldsig#\" " +
            "       Id=\"Request\" " +
            "       Version=\"2\">\r\n" +
            "   <Header>...</Header>\r\n" +
            "   <Invoice>...</Invoice>\r\n" +
            "</RegisterInvoiceRequest>";
        private const String KEYSTORE_LOCATION = "***.p12";
        private const String KEYSTORE_PASS = "***";

        public static void Main(string[] args)
        {
            using (X509Certificate2 keyStore = new X509Certificate2(KEYSTORE_LOCATION, KEYSTORE_PASS))
            {
                try
                {
                    // Load a private from a key store
                    RSA privateKey = keyStore.GetRSAPrivateKey();

                    // Convert string XML to object
                    XmlDocument request = new XmlDocument();
                    request.LoadXml(REQUEST_TO_SIGN);

                    // Create key info element
                    KeyInfo keyInfo = new KeyInfo();
                    KeyInfoX509Data keyInfoData = new KeyInfoX509Data();
                    keyInfoData.AddCertificate(keyStore);
                    keyInfo.AddClause(keyInfoData);

                    // Create signature reference
                    Reference reference = new Reference("");
                    reference.AddTransform(new XmlDsigEnvelopedSignatureTransform(false));
                    reference.AddTransform(new XmlDsigExcC14NTransform(false));
                    reference.DigestMethod = XML_DIG_METHOD;
                    reference.Uri = "#" + XML_REQUEST_ID;

                    // Create signature
                    SignedXml xml = new SignedXml(request);
                    xml.SigningKey = privateKey;
                    xml.SignedInfo.CanonicalizationMethod = SignedXml.XmlDsigExcC14NTransformUrl;
                    xml.SignedInfo.SignatureMethod = XML_SIG_METHOD;
                    xml.KeyInfo = keyInfo;
                    xml.AddReference(reference);
                    xml.ComputeSignature();

                    // Add signature element to the request
                    XmlElement signature = xml.GetXml();
                    request.DocumentElement.AppendChild(signature);

                    // Convert signed request to string and print
                    StringWriter sw = new StringWriter();
                    XmlTextWriter xw = new XmlTextWriter(sw);
                    request.WriteTo(xw);
                    Console.WriteLine("Signed document is: " + sw.ToString());
                }
                catch (Exception ex)
                {
                    Console.WriteLine(ex.Message);
                }
            }
        }
    }
}
```

6. ANNEX - WSDL version 2

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
    name="FiscalizationService"
    targetNamespace="https://eFiskalizimi.tatime.gov.al/FiscalizationService"
    xmlns:al="https://eFiskalizimi.tatime.gov.al/FiscalizationService"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:als="https://eFiskalizimi.tatime.gov.al/FiscalizationService/schema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <wsdl:types>
        <xsd:schema>
            <xsd:import namespace="https://eFiskalizimi.tatime.gov.al/FiscalizationService/schema" schemaLocation="alimc-fiscalization-service.xsd"/>
        </xsd:schema>
    </wsdl:types>

    <wsdl:message name="RegisterInvoiceRequest">
        <wsdl:documentation>Element representing register sale message.</wsdl:documentation>
        <wsdl:part element="als:RegisterInvoiceRequest" name="request" />
    </wsdl:message>

    <wsdl:message name="RegisterInvoiceResponse">
        <wsdl:documentation>Element representing register invoice response message.</wsdl:documentation>
        <wsdl:part element="als:RegisterInvoiceResponse" name="response" />
    </wsdl:message>

    <wsdl:message name="RegisterTCRRequest">
        <wsdl:documentation>Element representing request of the TCR registration message.</wsdl:documentation>
        <wsdl:part element="als:RegisterTCRRequest" name="request" />
    </wsdl:message>

    <wsdl:message name="RegisterTCRResponse">
        <wsdl:documentation>Element representing response of the TCR registration message.</wsdl:documentation>
        <wsdl:part element="als:RegisterTCRResponse" name="response" />
    </wsdl:message>

    <wsdl:message name="RegisterCashDepositRequest">
        <wsdl:documentation>Element representing request of the cash deposit request message.</wsdl:documentation>
        <wsdl:part element="als:RegisterCashDepositRequest" name="request" />
    </wsdl:message>

    <wsdl:message name="RegisterCashDepositResponse">
        <wsdl:documentation>Element representing response of the cash deposit response message.</wsdl:documentation>
        <wsdl:part element="als:RegisterCashDepositResponse" name="response" />
    </wsdl:message>

    <wsdl:portType name="FiscalizationServicePortType">
        <wsdl:operation name="registerInvoice">
            <wsdl:input message="al:RegisterInvoiceRequest"/>
            <wsdl:output message="al:RegisterInvoiceResponse"/>
        </wsdl:operation>
        <wsdl:operation name="registerTCR">
            <wsdl:input message="al:RegisterTCRRequest"/>
            <wsdl:output message="al:RegisterTCRResponse"/>
        </wsdl:operation>
        <wsdl:operation name="registerCashDeposit">
            <wsdl:input message="al:RegisterCashDepositRequest"/>
            <wsdl:output message="al:RegisterCashDepositResponse"/>
        </wsdl:operation>
    </wsdl:portType>

    <wsdl:binding name="FiscalizationServiceSoap" type="al:FiscalizationServicePortType">
        <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="registerInvoice">
            <soap:operation soapAction="https://eFiskalizimi.tatime.gov.al/FiscalizationService/RegisterInvoice"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="registerTCR">
            <soap:operation soapAction="https://eFiskalizimi.tatime.gov.al/FiscalizationService/RegisterTCR"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="registerCashDeposit">
            <soap:operation soapAction="https://eFiskalizimi.tatime.gov.al/FiscalizationService/RegisterCashDeposit"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
```

```
    <wsdl:service name="FiscalizationService">
        <wsdl:port name="FiscalizationServicePort" binding="al:FiscalizationServiceSoap">
            <soap:address location="https://eFiskalizimi.tatime.gov.al/FiscalizationService"/>
        </wsdl:port>
    </wsdl:service>

</wsdl:definitions>
```

7. ANNEX - XSD version 2

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="https://eFiskalizimi.tatime.gov.al/FiscalizationService/schema"
    xmlns:al="https://eFiskalizimi.tatime.gov.al/FiscalizationService/schema"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:vc="http://www.w3.org/2007/XMLSchema-versioning"
    elementFormDefault="qualified"
    vc:minVersion="1.1">

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

    <element name="RegisterInvoiceRequest">
        <annotation>
            <documentation>Root XML element representing register invoice message.</documentation>
        </annotation>
        <complexType>
            <all minOccurs="1" maxOccurs="1">
                <element name="Header" type="al:RegisterInvoiceRequestHeaderType" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <documentation>XML element representing header of the invoice containing data about the message (request)
sent.</documentation>
                    </annotation>
                </element>
                <element name="Invoice" type="al:InvoiceType" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <documentation>XML element representing a single invoice.</documentation>
                    </annotation>
                </element>
                <element ref="ds:Signature" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <documentation>XML element representing signature for the invoice.</documentation>
                    </annotation>
                </element>
            </all>
            <attribute name="Id" type="string" use="required" fixed="Request">
                <annotation>
                    <documentation>Attribute used for signature creation and verification.</documentation>
                </annotation>
            </attribute>
            <attribute name="Version" type="al:PositiveIntSType" use="required" fixed="2">
                <annotation>
                    <documentation>Attribute used to specify compliance with XSD schema.</documentation>
                </annotation>
            </attribute>
        </complexType>
    </element>
    <element name="RegisterInvoiceResponse">
        <annotation>
            <documentation>Root XML element representing register invoice response message.</documentation>
```

```xml
                </annotation>
        <complexType>
            <all>
                <element name="Header" type="al:RegisterInvoiceResponseHeaderType" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <documentation>Element representing the header of the message.</documentation>
                    </annotation>
                </element>
                <element name="FIC" type="al:UUIDSType" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <documentation>CIS generated verification code that can be used to uniquely identify registered
invoice.</documentation>
                    </annotation>
                </element>
                <element ref="ds:Signature" minOccurs="1" maxOccurs="1"/>
            </all>
            <attribute name="Id" type="string" use="required" fixed="Response">
                <annotation>
                    <documentation>Identification of the response, used to reference a signature.</documentation>
                </annotation>
            </attribute>
            <attribute name="Version" type="al:PositiveIntSType" use="required" fixed="2">
                <annotation>
                    <documentation>Identification of the schema version.</documentation>
                </annotation>
            </attribute>
        </complexType>
    </element>
    <element name="RegisterCashDepositRequest">
        <annotation>
            <documentation>Root XML element representing cash deposit request message.</documentation>
        </annotation>
        <complexType>
            <all minOccurs="1" maxOccurs="1">
                <element name="Header" type="al:RegisterCashDepositRequestHeaderType" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <documentation>Element representing the header of the message.</documentation>
                    </annotation>
                </element>
                <element name="CashDeposit" type="al:CashDepositType" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <documentation>Element representing a single cash deposit request.</documentation>
                    </annotation>
                </element>
                <element ref="ds:Signature" minOccurs="1" maxOccurs="1"/>
            </all>
            <attribute name="Id" type="string" use="required" fixed="Request">
                <annotation>
                    <documentation>Identification of the request, used to reference a signature.</documentation>
                </annotation>
            </attribute>
            <attribute name="Version" type="al:PositiveIntSType" use="required" fixed="2">
                <annotation>
```

```xml
                    <documentation>Identification of the schema version.</documentation>
                </annotation>
            </attribute>
        </complexType>
    </element>
    <element name="RegisterCashDepositResponse">
        <annotation>
            <documentation>Root XML element representing cash deposit response message.</documentation>
        </annotation>
        <complexType>
            <all minOccurs="1" maxOccurs="1">
                <element name="Header" type="al:RegisterCashDepositResponseHeaderType" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <documentation>Element representing the header of the message.</documentation>
                    </annotation>
                </element>
                <element name="FCDC" type="al:UUIDSType" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <documentation>Fiscalization cash deposit code generated by the CIS.</documentation>
                    </annotation>
                </element>
                <element ref="ds:Signature" minOccurs="1" maxOccurs="1"/>
            </all>
            <attribute name="Id" type="string" use="required" fixed="Response">
                <annotation>
                    <documentation>Identification of the response, used to reference a signature.</documentation>
                </annotation>
            </attribute>
            <attribute name="Version" type="al:PositiveIntSType" use="required" fixed="2">
                <annotation>
                    <documentation>Identification of the schema version.</documentation>
                </annotation>
            </attribute>
        </complexType>
    </element>
    <element name="RegisterTCRRequest">
        <annotation>
            <documentation>Root XML element representing TCR request message.</documentation>
        </annotation>
        <complexType>
            <all minOccurs="1" maxOccurs="1">
                <element name="Header" type="al:RegisterTCRRequestHeaderType" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <documentation>Element representing the header of the message.</documentation>
                    </annotation>
                </element>
                <element name="TCR" type="al:TCRType" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <documentation>XML element representing a single TCR registration message.</documentation>
                    </annotation>
                </element>
                <element ref="ds:Signature" minOccurs="1" maxOccurs="1"/>
            </all>
```

```xml
                    <attribute name="Id" type="string" use="required" fixed="Request">
                        <annotation>
                            <documentation>Identification of the request, used to reference a signature.</documentation>
                        </annotation>
                    </attribute>
                    <attribute name="Version" type="al:PositiveIntSType" use="required" fixed="2">
                        <annotation>
                            <documentation>Identification of the schema version.</documentation>
                        </annotation>
                    </attribute>
                </complexType>
            </element>
            <element name="RegisterTCRResponse">
                <annotation>
                    <documentation>Root XML element representing TCR response message.</documentation>
                </annotation>
                <complexType>
                    <all minOccurs="1" maxOccurs="1">
                        <element name="Header" type="al:RegisterTCRResponseHeaderType" minOccurs="1" maxOccurs="1">
                            <annotation>
                                <documentation>Element representing the header of the message.</documentation>
                            </annotation>
                        </element>
                        <element name="TCRCode" type="al:RegistrationCodeSType" minOccurs="1" maxOccurs="1">
                            <annotation>
                                <documentation>TCR code generated by the service.</documentation>
                            </annotation>
                        </element>
                        <element ref="ds:Signature" minOccurs="1" maxOccurs="1"/>
                    </all>
                    <attribute name="Id" type="string" use="required" fixed="Response">
                        <annotation>
                            <documentation>Identification of the response, used to reference a signature.</documentation>
                        </annotation>
                    </attribute>
                    <attribute name="Version" type="al:PositiveIntSType" use="required" fixed="2">
                        <annotation>
                            <documentation>Identification of the schema version.</documentation>
                        </annotation>
                    </attribute>
                </complexType>
            </element>

            <complexType name="RegisterInvoiceRequestHeaderType">
                <attribute name="UUID" type="al:UUIDSType" use="required">
                    <annotation>
                        <documentation>UUID generated by a TCR for every register sale data message send to the CIS.</documentation>
                    </annotation>
                </attribute>
                <attribute name="SendDateTime" use="required" type="al:UTCSType">
                    <annotation>
                        <documentation>Date and time of sending the register invoice data message from a TCR to the CIS.</documentation>
                    </annotation>
                </attribute>
```

```xml
                        </attribute>
                    <attribute name="IsSubseqDeliv" type="boolean" use="required" >
                        <annotation>
                            <documentation>Element that says if the request message is delivered subsequently.</documentation>
                        </annotation>
                    </attribute>
                </complexType>
            <complexType name="RegisterInvoiceResponseHeaderType">
                    <attribute name="UUID" type="al:UUIDSType" use="required">
                        <annotation>
                            <documentation>Element generated by the CIS for every message sent to the TCR. It uniquely identifies the message sent to the
TCR.</documentation>
                        </annotation>
                    </attribute>
                    <attribute name="RequestUUID" type="al:UUIDSType" use="required">
                        <annotation>
                            <documentation>Element generated by the TCR and referenced by the CIS. It uniquely identifies the request message for which
response message was sent to the TCR.</documentation>
                        </annotation>
                    </attribute>
                    <attribute name="SendDateTime" type="al:UTCSType" use="required">
                        <annotation>
                            <documentation>Element represents date and time of sending the response message to the TCR. </documentation>
                        </annotation>
                    </attribute>
                </complexType>
                <complexType name="RegisterCashDepositRequestHeaderType">
                    <attribute name="UUID" type="al:UUIDSType" use="required" >
                        <annotation>
                            <documentation>Element generated by the TCR. It uniquely identifies the request message sent from TCR to CIS.</documentation>
                        </annotation>
                    </attribute>
                    <attribute name="SendDateTime" type="al:UTCSType" use="required" >
                        <annotation>
                            <documentation>Element represents date and time of sending the request message to the CIS.</documentation>
                        </annotation>
                    </attribute>
                    <attribute name="IsSubseqDeliv" type="boolean" use="required" >
                        <annotation>
                            <documentation>Element that says if the request message is delivered subsequently.</documentation>
                        </annotation>
                    </attribute>
                </complexType>
            <complexType name="RegisterCashDepositResponseHeaderType">
                    <attribute name="UUID" type="al:UUIDSType" use="required">
                        <annotation>
                            <documentation>Element generated by the CIS. It uniquely identifies the response message sent from CIS to TCR.</documentation>
                        </annotation>
                    </attribute>
                    <attribute name="RequestUUID" type="al:UUIDSType" use="required">
                        <annotation>
                            <documentation>Element generated by the TCR and referenced by the CIS. It uniquely identifies the request message for which
response message was sent to the TCR.</documentation>
                        </annotation>
```

```xml
            </attribute>
            <attribute name="SendDateTime" type="al:UTCSType" use="required">
                <annotation>
                    <documentation>Element represents date and time of sending the response message to the TCR. </documentation>
                </annotation>
            </attribute>
        </complexType>
        <complexType name="RegisterTCRRequestHeaderType">
            <attribute name="UUID" type="al:UUIDSType" use="required">
                <annotation>
                    <documentation>Element generated by the TCR. It uniquely identifies the request message sent from TCR to CIS.</documentation>
                </annotation>
            </attribute>
            <attribute name="SendDateTime" type="al:UTCSType" use="required">
                <annotation>
                    <documentation>Element represents date and time of sending the request message to the CIS.</documentation>
                </annotation>
            </attribute>
        </complexType>
        <complexType name="RegisterTCRResponseHeaderType">
            <attribute name="UUID" type="al:UUIDSType" use="required">
                <annotation>
                    <documentation>Element generated by the CIS. It uniquely identifies the response message sent from CIS to TCR.
</documentation>
                </annotation>
            </attribute>
            <attribute name="RequestUUID" type="al:UUIDSType" use="required">
                <annotation>
                    <documentation>Element generated by the TCR and referenced by the CIS. It uniquely identifies the request message for which
the response message was sent to the TCR.</documentation>
                </annotation>
            </attribute>
            <attribute name="SendDateTime" type="al:UTCSType" use="required">
                <annotation>
                    <documentation>Element represents date and time of sending the response message to the TCR. </documentation>
                </annotation>
            </attribute>
        </complexType>


        <complexType name="InvoiceType">
            <all>
                <element name="SupplyDateOrPeriod" type="al:SupplyDateOrPeriodType" minOccurs="0" maxOccurs="1">
                    <annotation>
                        <documentation>XML element representing supply date or period is supply is different from the date when the invoice was
issued.</documentation>
                    </annotation>
                </element>
                <element name="CorrectiveInv" type="al:CorrectiveInvType" minOccurs="0" maxOccurs="1">
                    <annotation>
                        <documentation>XML element groups data for a corrective invoice.</documentation>
                    </annotation>
                </element>
                <element name="PayMethods" type="al:PayMethodsType" minOccurs="1" maxOccurs="1">
                    <annotation>
```

```xml
                    <documentation>XML element representing list of payment methods.</documentation>
                </annotation>
                <unique name="UniquePayMethodType">
                    <selector xpath=".//al:PayMethod"/>
                    <field xpath="@Type"/>
                </unique>
            </element>
            <element name="Currency" type="al:CurrencyType" minOccurs="0" maxOccurs="1">
                <annotation>
                    <documentation>XML element representing currency in which the amount on the invoice is expressed.</documentation>
                </annotation>
            </element>
            <element name="Seller" type="al:SellerType" minOccurs="1" maxOccurs="1" >
                <annotation>
                    <documentation>XML element representing an seller of the goods.</documentation>
                </annotation>
            </element>
            <element name="Buyer" type="al:BuyerType" minOccurs="0" maxOccurs="1" >
                <annotation>
                    <documentation>XML element representing a buyer of the goods.</documentation>
                </annotation>
            </element>
            <element name="Items" type="al:InvoiceItemsType" minOccurs="1" maxOccurs="1" >
                <annotation>
                    <documentation>XML element representing list of invoice items (goods or services).</documentation>
                </annotation>
            </element>
            <element name="SameTaxes" type="al:SameTaxesType" minOccurs="0" maxOccurs="1" >
                <annotation>
                    <documentation>XML element representing list of items of the same VAT rate.</documentation>
                </annotation>
            </element>
            <element name="ConsTaxes" type="al:ConsTaxesType" minOccurs="0" maxOccurs="1" >
                <annotation>
                    <documentation>XML element representing list of invoice items (goods or services) that are under consumption
tax.</documentation>
                </annotation>
            </element>
            <element name="Fees" type="al:FeesType" minOccurs="0" maxOccurs="1" >
                <annotation>
                    <documentation>XML element representing list of fees.</documentation>
                </annotation>
            </element>
            <element name="SumInvIICRefs" type="al:SumInvIICRefsType" minOccurs="0" maxOccurs="1" >
                <annotation>
                    <documentation>XML element representing list of invoices IIC referenced by a summary invoice.</documentation>
                </annotation>
            </element>
        </all>
        <attribute name="TypeOfInv" type="al:InvoiceSType" use="required" >
            <annotation>
                <documentation>Type of the item represents the type of invoice item, e.g. regular sale or a returned item.</documentation>
            </annotation>
```

```xml
        </attribute>
        <attribute name="IsSimplifiedInv" type="boolean" use="required" >
            <annotation>
                <documentation>If the invoice is simplified or not.</documentation>
            </annotation>
        </attribute>
        <attribute name="TypeOfSelfIss" type="al:SelfIssSType" use="optional" >
            <annotation>
                <documentation>Type of self-issuing.</documentation>
            </annotation>
        </attribute>
        <attribute name="IssueDateTime" type="al:UTCSType" use="required" >
            <annotation>
                <documentation>Date and time of invoice created and issued at TCR.</documentation>
            </annotation>
        </attribute>
        <attribute name="InvNum" type="al:InvNumSType" use="required" >
            <annotation>
                <documentation>Invoice number composed of invoice ordinal number, year of invoice issuance and code of TCR that issued
invoice.</documentation>
            </annotation>
        </attribute>
        <attribute name="InvOrdNum" type="al:PositiveIntSType" use="required" >
            <annotation>
                <documentation>Invoice ordinal number.</documentation>
            </annotation>
        </attribute>
        <attribute name="TCRCode" type="al:RegistrationCodeSType" use="optional" >
            <annotation>
                <documentation>Code of the device that issued the invoice.</documentation>
            </annotation>
        </attribute>
        <attribute name="IsIssuerInVAT" type="boolean" use="required" >
            <annotation>
                <documentation>Is the taxpayer in the VAT system.</documentation>
            </annotation>
        </attribute>
        <attribute name="TaxFreeAmt" type="al:DecimalNegSType" use="optional" >
            <annotation>
                <documentation>Amount on items that are tax free.</documentation>
            </annotation>
        </attribute>
        <attribute name="MarkUpAmt" type="al:DecimalNegSType" use="optional" >
            <annotation>
                <documentation>Amount of the mark-up on the invoice.</documentation>
            </annotation>
        </attribute>
        <attribute name="GoodsExAmt" type="al:DecimalNegSType" use="optional" >
            <annotation>
                <documentation>Total price of delivery of exported goods. There is no VAT on the invoice.</documentation>
            </annotation>
        </attribute>
        <attribute name="TotPriceWoVAT" type="al:DecimalNegSType" use="required" >
```

```xml
            <annotation>
                <documentation>Total price of the invoice without VAT.</documentation>
            </annotation>
        </attribute>
        <attribute name="TotVATAmt" type="al:DecimalNegSType" use="optional" >
            <annotation>
                <documentation>Total VAT amount.</documentation>
            </annotation>
        </attribute>
        <attribute name="TotPrice" type="al:DecimalNegSType" use="required" >
            <annotation>
                <documentation>Total price of the invoice including VAT.</documentation>
            </annotation>
        </attribute>
        <attribute name="OperatorCode" type="al:RegistrationCodeSType" use="required" >
            <annotation>
                <documentation>Reference to the operator who is operating on TCR. Value represents code of the operator.</documentation>
            </annotation>
        </attribute>
        <attribute name="BusinUnitCode" type="al:RegistrationCodeSType" use="required" >
            <annotation>
                <documentation>Code (ID) of the business unit in which the invoice is issued.</documentation>
            </annotation>
        </attribute>
        <attribute name="SoftCode" type="al:RegistrationCodeSType" use="required" >
            <annotation>
                <documentation>Number of the software used for invoice issuing.</documentation>
            </annotation>
        </attribute>
        <attribute name="IIC" type="al:Hex32SType" use="required" >
            <annotation>
                <documentation>Issuers's invoice code which is generated by the cash register of the issuer of the invoice. This is a unique code for every invoice. </documentation>
            </annotation>
        </attribute>
        <attribute name="IICSignature" type="al:Hex512SType" use="required" >
            <annotation>
                <documentation>Signed issuer's invoice code concatenated parameters.</documentation>
            </annotation>
        </attribute>


        <attribute name="IsReverseCharge" type="boolean" use="required" >
            <annotation>
                <documentation>Buyer is obliged to pay taxes by himself rather than issuer does it for him.</documentation>
            </annotation>
        </attribute>
        <attribute name="IsBadDebt" type="boolean" use="required" >
            <annotation>
                <documentation>If the invoice is marked as unpayable, it gets "bad debt" note.</documentation>
            </annotation>
        </attribute>
        <attribute name="PayDeadline" type="al:DateType" use="optional" >
            <annotation>
```

```xml
                    <documentation>Last day for a payment.</documentation>
                </annotation>
            </attribute>
    </complexType>
    <complexType name="InvoiceItemType">
        <attribute name="N" type="al:String50SType" use="required">
            <annotation>
                <documentation>Name of the item.</documentation>
            </annotation>
        </attribute>
        <attribute name="C" type="al:String50SType" use="optional">
            <annotation>
                <documentation>Code of the item.</documentation>
            </annotation>
        </attribute>
        <attribute name="U" type="al:String50SType" use="required">
            <annotation>
                <documentation>
                    Unit of measure for specific item – piece, weight,
                    length...
                </documentation>
            </annotation>
        </attribute>
        <attribute name="Q" type="al:NonNegDoubleForQuantitySType"
            use="required">
            <annotation>
                <documentation>
                    Amount or number (quantity) of items.
                </documentation>
            </annotation>
        </attribute>
        <attribute name="UPB" type="al:DecimalNegSType" use="required">
            <annotation>
                <documentation>
                    Price of one item before VAT is applied (unit price without VAT).
                </documentation>
            </annotation>
        </attribute>
        <attribute name="UPA" type="al:DecimalNegSType" use="optional">
            <annotation>
                <documentation>
                    Price of one item after VAT is applied (unit price with VAT).
                </documentation>
            </annotation>
        </attribute>
        <attribute name="R" type="al:DecimalSType" use="optional">
            <annotation>
                <documentation>Rebate percentage.</documentation>
            </annotation>
        </attribute>
        <attribute name="RR" type="boolean" use="optional">
            <annotation>
                <documentation>
```

```xml
                    Is rebate reducing base price?
                </documentation>
            </annotation>
        </attribute>
        <attribute name="PB" type="al:DecimalNegSType" use="required">
            <annotation>
                <documentation>
                    Price before VAT for the items in this group of
                    items. This is not the unit price of the item. It is
                    the unit price multiplied by the quantity of items.
                </documentation>
            </annotation>
        </attribute>
        <attribute name="VR" type="al:DecimalSType" use="optional">
            <annotation>
                <documentation>
                    Rate of value added tax expressed as percentage.
                </documentation>
            </annotation>
        </attribute>
        <attribute name="VA" type="al:DecimalNegSType" use="optional">
            <annotation>
                <documentation>
                    Amount of value added tax for the items in this
                    group of items.
                </documentation>
            </annotation>
        </attribute>
        <attribute name="PA" type="al:DecimalNegSType" use="required">
            <annotation>
                <documentation>
                    Price after applying VAT for the items in this group
                    of items.
                </documentation>
            </annotation>
        </attribute>
        <attribute name="EX" type="al:ExemptFromVATSType"
            use="optional">
            <annotation>
                <documentation>Exempt from VAT.</documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="InvoiceItemsType">
        <sequence>
            <element name="I" type="al:InvoiceItemType" minOccurs="1" maxOccurs="1000">
                <annotation>
                    <documentation>Element representing a single item on the invoice.</documentation>
                </annotation>
            </element>
        </sequence>
    </complexType>
    <complexType name="SupplyDateOrPeriodType">
```

```xml
            <attribute name="Start" type="al:DateType" use="required">
                <annotation>
                    <documentation>Issuer's NUIS.</documentation>
                </annotation>
            </attribute>
            <attribute name="End" type="al:DateType" use="required">
                <annotation>
                    <documentation>Issuer's name.</documentation>
                </annotation>
            </attribute>
        </complexType>
    <complexType name="CorrectiveInvType">
        <attribute name="IICRef" type="al:Hex32SType" use="required">
            <annotation>
                <documentation>IIC reference on the original invoice.</documentation>
            </annotation>
        </attribute>
        <attribute name="IssueDateTime" type="al:UTCSType" use="required">
            <annotation>
                <documentation>Date and time the original invoice is created and issued at TCR</documentation>
            </annotation>
        </attribute>
        <attribute name="Type" type="al:CorrectiveInvTypeSType" use="required">
            <annotation>
                <documentation>Type of the corrective invoice.</documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="CurrencyType">
        <attribute name="Code" type="al:CurrencyCodeSType" use="required">
            <annotation>
                <documentation>Currency code from the ISO 4217 standard.</documentation>
            </annotation>
        </attribute>
        <attribute name="ExRate" type="al:NonNegDoubleSType" use="required">
            <annotation>
                <documentation>Exchange rate applied to calculate the equivalent amount of the total amount expressed in currency different from ALL.</documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="SellerType">
        <attribute name="IDType" type="al:IDTypeSType" use="required">
            <annotation>
                <documentation>Seller's identification number type.</documentation>
            </annotation>
        </attribute>
        <attribute name="IDNum" type="al:String20SType" use="required">
            <annotation>
                <documentation>Seller's identification number.</documentation>
            </annotation>
        </attribute>
        <attribute name="Name" type="al:String100SType" use="required">
```

```
                <annotation>
                    <documentation>Seller's name.</documentation>
                </annotation>
            </attribute>
            <attribute name="Address" type="al:String100SType" use="optional">
                <annotation>
                    <documentation>Seller's address.</documentation>
                </annotation>
            </attribute>
            <attribute name="Town" type="al:String100SType" use="optional">
                <annotation>
                    <documentation>Seller's town.</documentation>
                </annotation>
            </attribute>
            <attribute name="Country" type="al:CountryCodeSType" use="optional">
                <annotation>
                    <documentation>Seller's country.</documentation>
                </annotation>
            </attribute>
        </complexType>
        <complexType name="BuyerType">
            <attribute name="IDType" type="al:IDTypeSType" use="optional">
                <annotation>
                    <documentation>Buyer's identification number type.</documentation>
                </annotation>
            </attribute>
            <attribute name="IDNum" type="al:String20SType" use="optional">
                <annotation>
                    <documentation>Buyer's identification number.</documentation>
                </annotation>
            </attribute>
            <attribute name="Name" type="al:String100SType" use="optional">
                <annotation>
                    <documentation>Buyer's name.</documentation>
                </annotation>
            </attribute>
            <attribute name="Address" type="al:String100SType" use="optional">
                <annotation>
                    <documentation>Buyer's address.</documentation>
                </annotation>
            </attribute>
            <attribute name="Town" type="al:String100SType" use="optional">
                <annotation>
                    <documentation>Buyer's town.</documentation>
                </annotation>
            </attribute>
            <attribute name="Country" type="al:CountryCodeSType" use="optional">
                <annotation>
                    <documentation>Buyer's country.</documentation>
                </annotation>
            </attribute>
        </complexType>
        <complexType name="PayMethodsType">
```

```xml
<sequence>
    <element name="PayMethod" type="al:PayMethodType" minOccurs="1" maxOccurs="10">
        <annotation>
            <documentation>XML element representing one payment method.</documentation>
        </annotation>
    </element>
</sequence>
</complexType>
<complexType name="SameTaxesType">
    <sequence>
        <element name="SameTax" type="al:SameTaxType" minOccurs="1" maxOccurs="20">
            <annotation>
                <documentation>Element representing an item of list of same tax items.</documentation>
            </annotation>
        </element>
    </sequence>
</complexType>
<complexType name="ConsTaxesType">
    <sequence>
        <element name="ConsTax" type="al:ConsTaxType" minOccurs="1" maxOccurs="20">
            <annotation>
                <documentation>Element representing an item on list of items under consumption tax.</documentation>
            </annotation>
        </element>
    </sequence>
</complexType>
<complexType name="FeesType">
    <sequence>
        <element name="Fee" type="al:FeeType" minOccurs="1" maxOccurs="20">
            <annotation>
                <documentation>Element representing an item of list of fee items.</documentation>
            </annotation>
        </element>
    </sequence>
</complexType>
<complexType name="SumInvIICRefsType">
    <sequence>
        <element name="SumInvIICRef" type="al:SumInvIICRefType" minOccurs="1" maxOccurs="1000">
            <annotation>
                <documentation>XML element representing a single IIC reference on the list of SumInvIICRefs.</documentation>
            </annotation>
        </element>
    </sequence>
</complexType>
<complexType name="PayMethodType">
    <sequence>
        <element name="Vouchers" type="al:VouchersType" minOccurs="0" maxOccurs="1">
            <annotation>
                <documentation>XML element representing list of voucher numbers if the payment method is voucher.</documentation>
            </annotation>
        </element>
    </sequence>
    <attribute name="Type" type="al:PaymentMethodTypeSType" use="required">
```

```xml
                    <annotation>
                        <documentation>Type of the payment method.</documentation>
                    </annotation>
                </attribute>
                <attribute name="Amt" type="al:DecimalNegSType" use="optional">
                    <annotation>
                        <documentation>Amount payed by payment method in the country currency.</documentation>
                    </annotation>
                </attribute>
                <attribute name="CompCard" type="al:String50SType" use="optional">
                    <annotation>
                        <documentation>Company card number if the payment method is company cards.</documentation>
                    </annotation>
                </attribute>
            </complexType>
            <complexType name="VouchersType">
                <sequence>
                    <element name="Voucher" type="al:VoucherType" minOccurs="1" maxOccurs="20">
                        <annotation>
                            <documentation>XML element representing a single voucher number.</documentation>
                        </annotation>
                    </element>
                </sequence>
            </complexType>
            <complexType name="VoucherType">
                <attribute name="Num" type="al:VoucherNumSType" use="required">
                    <annotation>
                        <documentation>Voucher number.</documentation>
                    </annotation>
                </attribute>
            </complexType>
            <complexType name="SameTaxType">
                <attribute name="NumOfItems" type="al:PositiveIntSType" use="required">
                    <annotation>
                        <documentation>Number of items.</documentation>
                    </annotation>
                </attribute>
                <attribute name="PriceBefVAT" type="al:DecimalNegSType" use="required">
                    <annotation>
                        <documentation>Price before VAT.</documentation>
                    </annotation>
                </attribute>
                <attribute name="VATRate" type="al:DecimalNegSType" use="optional">
                    <annotation>
                        <documentation>VAT rate.</documentation>
                    </annotation>
                </attribute>
                <attribute name="ExemptFromVAT" type="al:ExemptFromVATSType" use="optional">
                    <annotation>
                        <documentation>Exempt from VAT.</documentation>
                    </annotation>
                </attribute>
                <attribute name="VATAmt" type="al:DecimalNegSType" use="optional">
```

```xml
            <annotation>
                <documentation>VAT amount.</documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="ConsTaxType">
        <attribute name="NumOfItems" type="al:PositiveIntSType" use="required">
            <annotation>
                <documentation>Number of items.</documentation>
            </annotation>
        </attribute>
        <attribute name="PriceBefConsTax" type="al:DecimalNegSType" use="required">
            <annotation>
                <documentation>Price before consumption tax.</documentation>
            </annotation>
        </attribute>
        <attribute name="ConsTaxRate" type="al:DecimalNegSType" use="required">
            <annotation>
                <documentation>Consumption tax rate.</documentation>
            </annotation>
        </attribute>
        <attribute name="ConsTaxAmt" type="al:DecimalNegSType" use="required">
            <annotation>
                <documentation>Consumption tax amount.</documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="FeeType">
        <attribute name="Type" type="al:FeeTypeSType" use="required">
            <annotation>
                <documentation>Type of the fee.</documentation>
            </annotation>
        </attribute>
        <attribute name="Amt" type="al:DecimalNegSType" use="required">
            <annotation>
                <documentation>The decimal amount of the fee in the country currency.</documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="SumInvIICRefType">
        <attribute name="IIC" type="al:Hex32SType" use="required">
            <annotation>
                <documentation>IIC of the invoice that is referenced by the summary invoice.</documentation>
            </annotation>
        </attribute>
        <attribute name="IssueDateTime" type="al:UTCSType" use="required">
            <annotation>
                <documentation>Date and time the invoice referenced by the summary invoice is created and issued at TCR.</documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="CashDepositType">
        <attribute name="ChangeDateTime" type="al:UTCSType" use="required" >
```

```xml
        <annotation>
            <documentation>Element representing date and time when the deposit of cash was checked in the cash register.</documentation>
        </annotation>
    </attribute>
    <attribute name="Operation" type="al:CashDepositOperationSType" use="required">
        <annotation>
            <documentation>This is an XML element that represents the operation made at the cash register.</documentation>
        </annotation>
    </attribute>
    <attribute name="CashAmt" type="al:DecimalNegSType" use="required" >
        <annotation>
            <documentation>Element representing the amount of cash found in the cash register after the operation.</documentation>
        </annotation>
    </attribute>
    <attribute name="TCRCode" type="al:RegistrationCodeSType" use="required" >
        <annotation>
            <documentation>Element representing the unique code of the TCR in question.</documentation>
        </annotation>
    </attribute>
    <attribute name="IssuerNUIS" type="al:NUISSType" use="required">
        <annotation>
            <documentation>Element representing issuer's NUIS (tax number).</documentation>
        </annotation>
    </attribute>
</complexType>
<complexType name="TCRType">
    <simpleContent>
        <extension base="string">
            <attribute name="IssuerNUIS" type="al:NUISSType" use="required">
                <annotation>
                    <documentation>Issuer's NUIS.</documentation>
                </annotation>
            </attribute>
            <attribute name="BusinUnitCode" type="al:RegistrationCodeSType" use="required">
                <annotation>
                    <documentation>Code of the business unit in which the invoice is issued.</documentation>
                </annotation>
            </attribute>
            <attribute name="TCRIntID" type="al:String50SType" use="required">
                <annotation>
                    <documentation>Element representing the internal identification of the TCR.</documentation>
                </annotation>
            </attribute>
            <attribute name="SoftCode" type="al:RegistrationCodeSType" use="optional">
                <annotation>
                    <documentation>Code of the software used for invoice issuing.</documentation>
                </annotation>
            </attribute>
            <attribute name="MaintainerCode" type="al:RegistrationCodeSType" use="optional">
                <annotation>
                    <documentation>Code of the maintainer of the software used for invoice issuing.</documentation>
                </annotation>
            </attribute>
```

```xml
                    <attribute name="ValidFrom" type="al:DateType" use="optional">
                        <annotation>
                            <documentation>Date from which the TCR will be used.</documentation>
                        </annotation>
                    </attribute>
                    <attribute name="ValidTo" type="al:DateType" use="optional">
                        <annotation>
                            <documentation>Date until the TCR will be used.</documentation>
                        </annotation>
                    </attribute>
                </extension>
            </simpleContent>
        </complexType>
        <simpleType name="InvoiceSType">
            <annotation>
                <documentation>Invoice type.</documentation>
            </annotation>
            <restriction base="string">
                <enumeration value="CASH">
                    <annotation>
                        <documentation>Cash invoice.</documentation>
                    </annotation>
                </enumeration>
                <enumeration value="NONCASH">
                    <annotation>
                        <documentation>Non-cash invoice.</documentation>
                    </annotation>
                </enumeration>
            </restriction>
        </simpleType>
        <simpleType name="DecimalSType">
            <annotation>
                <documentation>Decimal number with two numbers after decimal point.</documentation>
            </annotation>
            <restriction base="decimal">
                <pattern value="([1-9][0-9]*|0)\.[0-9]{2}|0"/>
            </restriction>
        </simpleType>
        <simpleType name="SelfIssSType">
            <annotation>
                <documentation>Self issuing type.</documentation>
            </annotation>
            <restriction base="string">
                <enumeration value="AGREEMENT">
                    <annotation>
                        <documentation>The previous agreement between the parties.</documentation>
                    </annotation>
                </enumeration>
                <enumeration value="DOMESTIC">
                    <annotation>
                        <documentation>Buying from domestic farmers.</documentation>
                    </annotation>
                </enumeration>
```

```xml
            <enumeration value="ABROAD">
                <annotation>
                    <documentation>Buying services from abroad.</documentation>
                </annotation>
            </enumeration>
            <enumeration value="OTHER">
                <annotation>
                    <documentation>Other</documentation>
                </annotation>
            </enumeration>
        </restriction>
    </simpleType>
    <simpleType name="PaymentMethodTypeSType">
        <annotation>
            <documentation>Payment method type.</documentation>
        </annotation>
        <restriction base="string">
            <enumeration value="BANKNOTE">
                <annotation>
                    <documentation>Banknotes and coins</documentation>
                </annotation>
            </enumeration>
            <enumeration value="CARD">
                <annotation>
                    <documentation>Credit and debit card</documentation>
                </annotation>
            </enumeration>
            <enumeration value="CHECK">
                <annotation>
                    <documentation>Bank check</documentation>
                </annotation>
            </enumeration>
            <enumeration value="SVOUCHER">
                <annotation>
                    <documentation>Single-purpose voucher</documentation>
                </annotation>
            </enumeration>
            <enumeration value="MVOUCHER">
                <annotation>
                    <documentation>Multi-purpose voucher</documentation>
                </annotation>
            </enumeration>
            <enumeration value="COMPANY">
                <annotation>
                    <documentation>Seller's company cards and similar</documentation>
                </annotation>
            </enumeration>
            <enumeration value="ORDER">
                <annotation>
                    <documentation>Invoice not yet paid. It will be paid by summary invoice.</documentation>
                </annotation>
            </enumeration>
            <enumeration value="ACCOUNT">
```

```xml
                <annotation>
                    <documentation>Transaction account</documentation>
                </annotation>
            </enumeration>
            <enumeration value="FACTORING">
                <annotation>
                    <documentation>Factoring</documentation>
                </annotation>
            </enumeration>
            <enumeration value="COMPENSATION">
                <annotation>
                    <documentation>Compensation</documentation>
                </annotation>
            </enumeration>
            <enumeration value="TRANSFER">
                <annotation>
                    <documentation>Transfer of rights or debts</documentation>
                </annotation>
            </enumeration>
            <enumeration value="WAIVER">
                <annotation>
                    <documentation>Waiver of debts</documentation>
                </annotation>
            </enumeration>
            <enumeration value="KIND">
                <annotation>
                    <documentation>Payment in kind (clearing)</documentation>
                </annotation>
            </enumeration>
            <enumeration value="OTHER">
                <annotation>
                    <documentation>Other cashless payments</documentation>
                </annotation>
            </enumeration>
        </restriction>
</simpleType>
<simpleType name="UTCSType">
    <annotation>
        <documentation>Date and time represented as UTC time with ISO 8601 format.</documentation>
    </annotation>
    <restriction base="dateTime">
        <pattern value="[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}[+-][0-9]{2}:[0-9]{2}"/>
    </restriction>
</simpleType>
<simpleType name="DateType">
    <annotation>
        <documentation>Date expressed as year-month-day.</documentation>
    </annotation>
    <restriction base="date">
        <pattern value="[0-9]{4}-[0-9]{2}-[0-9]{2}"/>
    </restriction>
</simpleType>
<simpleType name="UUIDSType">
```

```xml
        <annotation>

            <documentation>UUID constructed according to the RFC4122 (https://tools.ietf.org/html/rfc4122).</documentation>

        </annotation>

        <restriction base="string">

            <pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[1-5][0-9a-fA-F]{3}-[89abAB][0-9a-fA-F]{3}-[0-9a-fA-F]{12}"/>

        </restriction>

    </simpleType>

    <simpleType name="VoucherNumSType">

        <annotation>

            <documentation>Voucher number.</documentation>

        </annotation>

        <restriction base="string">

            <pattern value="[1-9]{1}[0-9]{0,7}-[0-9]{4}-[a-zA-Z]{1}[0-9]{8}[a-zA-Z]{1}"/>

        </restriction>

    </simpleType>

    <simpleType name="NUISSType">

        <annotation>

            <documentation>NUIS constructed in one letter - eight numbers - one letter pattern, unique.</documentation>

        </annotation>

        <restriction base="string">

            <pattern value="[a-zA-Z]{1}[0-9]{8}[a-zA-Z]{1}"/>

        </restriction>

    </simpleType>

    <simpleType name="InvNumSType">

        <annotation>

            <documentation>Invoice number composed of Invoice ordinal number, year of invoice issuing and TCR code.</documentation>

        </annotation>

        <restriction base="string">

            <pattern value="[0-9][1-9]{0,14}\/[0-9]{4}(\/[a-z]{2}[0-9]{3}[a-z]{2}[0-9]{3})?"/>

        </restriction>

    </simpleType>

    <simpleType name="String100SType">

        <annotation>

            <documentation>String of up to 100 characters.</documentation>

        </annotation>

        <restriction base="string">

            <maxLength value="100"/>

        </restriction>

    </simpleType>

    <simpleType name="String50SType">

        <annotation>

            <documentation>String of up to 50 characters.</documentation>

        </annotation>

        <restriction base="string">

            <maxLength value="50"/>

        </restriction>

    </simpleType>

    <simpleType name="String20SType">

        <annotation>

            <documentation>String of up to 20 characters.</documentation>

        </annotation>

        <restriction base="string">

            <maxLength value="20"/>
```

```xml
            </restriction>
        </simpleType>
        <simpleType name="Hex32SType">
            <annotation>
                <documentation>Hexadecimal code made of 32 characters.</documentation>
            </annotation>
            <restriction base="string">
                <pattern value="[0-9a-fA-F]{32}"/>
            </restriction>
        </simpleType>
        <simpleType name="Hex512SType">
            <annotation>
                <documentation>Hexadecimal code made of 512 characters.</documentation>
            </annotation>
            <restriction base="string">
                <pattern value="[0-9a-fA-F]{512}"/>
            </restriction>
        </simpleType>
        <simpleType name="DecimalNegSType">
            <annotation>
                <documentation>Decimal number that can be written as negative number.</documentation>
            </annotation>
            <restriction base="decimal">
                <pattern value="(-?[1-9][0-9]*|0)\.[0-9]{2}|0"/>
            </restriction>
        </simpleType>
        <simpleType name="RegistrationCodeSType">
            <annotation>
                <documentation>Registration code in two letters - three numbers - two letters - three number format.</documentation>
            </annotation>
            <restriction base="string">
                <pattern value="[a-z]{2}[0-9]{3}[a-z]{2}[0-9]{3}"/>
            </restriction>
        </simpleType>
        <simpleType name="NonNegDoubleForQuantitySType">
            <annotation>
                <documentation>Double number that can not be negative.</documentation>
            </annotation>
            <restriction base="double">
                <minInclusive value="0.001"/>
            </restriction>
        </simpleType>
        <simpleType name="NonNegDoubleSType">
            <annotation>
                <documentation>
                    Double number that can not be negative.
                </documentation>
            </annotation>
            <restriction base="double">
                <minExclusive value="0"></minExclusive>
            </restriction>
        </simpleType>
        <simpleType name="PositiveIntSType">
```

```xml
                <annotation>
                    <documentation>Positive integer.</documentation>
                </annotation>
                <restriction base="int">
                    <minInclusive value="0"/>
                </restriction>
        </simpleType>
        <simpleType name="CashDepositOperationSType">
                <annotation>
                    <documentation>Cash deposit operation types.</documentation>
                </annotation>
                <restriction base="string">
                    <enumeration value="INITIAL">
                        <annotation>
                            <documentation>Initial deposit in the TCR.</documentation>
                        </annotation>
                    </enumeration>
                    <enumeration value="INOUT">
                        <annotation>
                            <documentation>Amount of cash deposited or withdrawn from the TCR.</documentation>
                        </annotation>
                    </enumeration>
                </restriction>
        </simpleType>
        <simpleType name="IDTypeSType">
                <annotation>
                    <documentation>Identification number types.</documentation>
                </annotation>
                <restriction base="string">
                    <enumeration value="NUIS">
                        <annotation>
                            <documentation>NUIS number.</documentation>
                        </annotation>
                    </enumeration>
                    <enumeration value="ID">
                        <annotation>
                            <documentation>Personal ID number.</documentation>
                        </annotation>
                    </enumeration>
                    <enumeration value="PASS">
                        <annotation>
                            <documentation>Passport number.</documentation>
                        </annotation>
                    </enumeration>
                    <enumeration value="VAT">
                        <annotation>
                            <documentation>VAT number.</documentation>
                        </annotation>
                    </enumeration>
                    <enumeration value="TAX">
                        <annotation>
                            <documentation>TAX number.</documentation>
                        </annotation>
```

```xml
                    </enumeration>
                    <enumeration value="SOC">
                        <annotation>
                            <documentation>Social security number.</documentation>
                        </annotation>
                    </enumeration>
                </restriction>
        </simpleType>
        <simpleType name="FeeTypeSType">
            <annotation>
                <documentation>Fee types.</documentation>
            </annotation>
            <restriction base="string">
                <enumeration value="PACK">
                    <annotation>
                        <documentation>Packaging fee.</documentation>
                    </annotation>
                </enumeration>
                <enumeration value="BOTTLE">
                    <annotation>
                        <documentation>Fee for the return of glass bottles.</documentation>
                    </annotation>
                </enumeration>
                <enumeration value="COMMISSION">
                    <annotation>
                        <documentation>Commission for currency exchange activities.</documentation>
                    </annotation>
                </enumeration>
                <enumeration value="OTHER">
                    <annotation>
                        <documentation>Other fees that are not listed here.</documentation>
                    </annotation>
                </enumeration>
            </restriction>
        </simpleType>
        <simpleType name="CorrectiveInvTypeSType">
            <annotation>
                <documentation>Types of the corrective invoice.</documentation>
            </annotation>
            <restriction base="string">
                <enumeration value="CORRECTIVE">
                    <annotation>
                        <documentation>Corrective invoice.</documentation>
                    </annotation>
                </enumeration>
                <enumeration value="DEBIT">
                    <annotation>
                        <documentation>Debit note.</documentation>
                    </annotation>
                </enumeration>
                <enumeration value="CREDIT">
                    <annotation>
                        <documentation>Credit note.</documentation>
```

```xml
            </annotation>
        </enumeration>
    </restriction>
</simpleType>
<simpleType name="ExemptFromVATSType">
    <annotation>
        <documentation>Exempt from VAT types.</documentation>
    </annotation>
    <restriction base="string">
        <enumeration value="TYPE_1">
            <annotation>
                <documentation>Exempt type 1.</documentation>
            </annotation>
        </enumeration>
        <enumeration value="TYPE_2">
            <annotation>
                <documentation>Exempt type 2.</documentation>
            </annotation>
        </enumeration>
    </restriction>
</simpleType>
<simpleType name="CountryCodeSType">
    <annotation>
        <documentation>Country codes from ISO 3166-1 alpha3 standard</documentation>
    </annotation>
    <restriction base="string">
        <enumeration value="ABW"><annotation><documentation>Aruba</documentation></annotation></enumeration>
        <enumeration value="AFG"><annotation><documentation>Afghanistan</documentation></annotation></enumeration>
        <enumeration value="AGO"><annotation><documentation>Angola</documentation></annotation></enumeration>
        <enumeration value="AIA"><annotation><documentation>Anguilla</documentation></annotation></enumeration>
        <enumeration value="ALA"><annotation><documentation>Aland Islands</documentation></annotation></enumeration>
        <enumeration value="ALB"><annotation><documentation>Albania</documentation></annotation></enumeration>
        <enumeration value="AND"><annotation><documentation>Andorra</documentation></annotation></enumeration>
        <enumeration value="ARE"><annotation><documentation>United Arab Emirates</documentation></annotation></enumeration>
        <enumeration value="ARG"><annotation><documentation>Argentina</documentation></annotation></enumeration>
        <enumeration value="ARM"><annotation><documentation>Armenia</documentation></annotation></enumeration>
        <enumeration value="ASM"><annotation><documentation>American Samoa</documentation></annotation></enumeration>
        <enumeration value="ATA"><annotation><documentation>Antarctica</documentation></annotation></enumeration>
        <enumeration value="ATF"><annotation><documentation>French Southern Territories</documentation></annotation></enumeration>
        <enumeration value="ATG"><annotation><documentation>Antigua and Barbuda</documentation></annotation></enumeration>
        <enumeration value="AUS"><annotation><documentation>Australia</documentation></annotation></enumeration>
        <enumeration value="AUT"><annotation><documentation>Austria</documentation></annotation></enumeration>
        <enumeration value="AZE"><annotation><documentation>Azerbaijan</documentation></annotation></enumeration>
        <enumeration value="BDI"><annotation><documentation>Burundi</documentation></annotation></enumeration>
        <enumeration value="BEL"><annotation><documentation>Belgium</documentation></annotation></enumeration>
        <enumeration value="BEN"><annotation><documentation>Benin</documentation></annotation></enumeration>
        <enumeration value="BES"><annotation><documentation>Bonaire, Sint Eustatius and Saba</documentation></annotation></enumeration>
        <enumeration value="BFA"><annotation><documentation>Burkina Faso</documentation></annotation></enumeration>
        <enumeration value="BGD"><annotation><documentation>Bangladesh</documentation></annotation></enumeration>
        <enumeration value="BGR"><annotation><documentation>Bulgaria</documentation></annotation></enumeration>
        <enumeration value="BHR"><annotation><documentation>Bahrain</documentation></annotation></enumeration>
        <enumeration value="BHS"><annotation><documentation>Bahamas</documentation></annotation></enumeration>
        <enumeration value="BIH"><annotation><documentation>Bosnia and Herzegovina</documentation></annotation></enumeration>
```

```xml
<enumeration value="BLM"><annotation><documentation>Saint Barthelemy</documentation></annotation></enumeration>
<enumeration value="BLR"><annotation><documentation>Belarus</documentation></annotation></enumeration>
<enumeration value="BLZ"><annotation><documentation>Belize</documentation></annotation></enumeration>
<enumeration value="BMU"><annotation><documentation>Bermuda</documentation></annotation></enumeration>
<enumeration value="BOL"><annotation><documentation>Bolivia (Plurinational State of)</documentation></annotation></enumeration>
<enumeration value="BRA"><annotation><documentation>Brazil</documentation></annotation></enumeration>
<enumeration value="BRB"><annotation><documentation>Barbados</documentation></annotation></enumeration>
<enumeration value="BRN"><annotation><documentation>Brunei Darussalam</documentation></annotation></enumeration>
<enumeration value="BTN"><annotation><documentation>Bhutan</documentation></annotation></enumeration>
<enumeration value="BVT"><annotation><documentation>Bouvet Island</documentation></annotation></enumeration>
<enumeration value="BWA"><annotation><documentation>Botswana</documentation></annotation></enumeration>
<enumeration value="CAF"><annotation><documentation>Central African Republic</documentation></annotation></enumeration>
<enumeration value="CAN"><annotation><documentation>Canada</documentation></annotation></enumeration>
<enumeration value="CCK"><annotation><documentation>Cocos (Keeling) Islands</documentation></annotation></enumeration>
<enumeration value="CHE"><annotation><documentation>Switzerland</documentation></annotation></enumeration>
<enumeration value="CHL"><annotation><documentation>Chile</documentation></annotation></enumeration>
<enumeration value="CHN"><annotation><documentation>China</documentation></annotation></enumeration>
<enumeration value="CIV"><annotation><documentation>Cote d'Ivoire</documentation></annotation></enumeration>
<enumeration value="CMR"><annotation><documentation>Cameroon</documentation></annotation></enumeration>
<enumeration value="COD"><annotation><documentation>Congo, Democratic Republic of the</documentation></annotation></enumeration>
<enumeration value="COG"><annotation><documentation>Congo</documentation></annotation></enumeration>
<enumeration value="COK"><annotation><documentation>Cook Islands</documentation></annotation></enumeration>
<enumeration value="COL"><annotation><documentation>Colombia</documentation></annotation></enumeration>
<enumeration value="COM"><annotation><documentation>Comoros</documentation></annotation></enumeration>
<enumeration value="CPV"><annotation><documentation>Cabo Verde</documentation></annotation></enumeration>
<enumeration value="CRI"><annotation><documentation>Costa Rica</documentation></annotation></enumeration>
<enumeration value="CUB"><annotation><documentation>Cuba</documentation></annotation></enumeration>
<enumeration value="CUW"><annotation><documentation>Curacao</documentation></annotation></enumeration>
<enumeration value="CXR"><annotation><documentation>Christmas Island</documentation></annotation></enumeration>
<enumeration value="CYM"><annotation><documentation>Cayman Islands</documentation></annotation></enumeration>
<enumeration value="CYP"><annotation><documentation>Cyprus</documentation></annotation></enumeration>
<enumeration value="CZE"><annotation><documentation>Czechia</documentation></annotation></enumeration>
<enumeration value="DEU"><annotation><documentation>Germany</documentation></annotation></enumeration>
<enumeration value="DJI"><annotation><documentation>Djibouti</documentation></annotation></enumeration>
<enumeration value="DMA"><annotation><documentation>Dominica</documentation></annotation></enumeration>
<enumeration value="DNK"><annotation><documentation>Denmark</documentation></annotation></enumeration>
<enumeration value="DOM"><annotation><documentation>Dominican Republic</documentation></annotation></enumeration>
<enumeration value="DZA"><annotation><documentation>Algeria</documentation></annotation></enumeration>
<enumeration value="ECU"><annotation><documentation>Ecuador</documentation></annotation></enumeration>
<enumeration value="EGY"><annotation><documentation>Egypt</documentation></annotation></enumeration>
<enumeration value="ERI"><annotation><documentation>Eritrea</documentation></annotation></enumeration>
<enumeration value="ESH"><annotation><documentation>Western Sahara</documentation></annotation></enumeration>
<enumeration value="ESP"><annotation><documentation>Spain</documentation></annotation></enumeration>
<enumeration value="EST"><annotation><documentation>Estonia</documentation></annotation></enumeration>
<enumeration value="ETH"><annotation><documentation>Ethiopia</documentation></annotation></enumeration>
<enumeration value="FIN"><annotation><documentation>Finland</documentation></annotation></enumeration>
<enumeration value="FJI"><annotation><documentation>Fiji</documentation></annotation></enumeration>
<enumeration value="FLK"><annotation><documentation>Falkland Islands (Malvinas)</documentation></annotation></enumeration>
<enumeration value="FRA"><annotation><documentation>France</documentation></annotation></enumeration>
<enumeration value="FRO"><annotation><documentation>Faroe Islands</documentation></annotation></enumeration>
<enumeration value="FSM"><annotation><documentation>Micronesia (Federated States of)</documentation></annotation></enumeration>
<enumeration value="GAB"><annotation><documentation>Gabon</documentation></annotation></enumeration>
<enumeration value="GBR"><annotation><documentation>United Kingdom of Great Britain and Northern
```

```xml
Ireland</documentation></annotation></enumeration>
          <enumeration value="GEO"><annotation><documentation>Georgia</documentation></annotation></enumeration>
          <enumeration value="GGY"><annotation><documentation>Guernsey</documentation></annotation></enumeration>
          <enumeration value="GHA"><annotation><documentation>Ghana</documentation></annotation></enumeration>
          <enumeration value="GIB"><annotation><documentation>Gibraltar</documentation></annotation></enumeration>
          <enumeration value="GIN"><annotation><documentation>Guinea</documentation></annotation></enumeration>
          <enumeration value="GLP"><annotation><documentation>Guadeloupe</documentation></annotation></enumeration>
          <enumeration value="GMB"><annotation><documentation>Gambia</documentation></annotation></enumeration>
          <enumeration value="GNB"><annotation><documentation>Guinea-Bissau</documentation></annotation></enumeration>
          <enumeration value="GNQ"><annotation><documentation>Equatorial Guinea</documentation></annotation></enumeration>
          <enumeration value="GRC"><annotation><documentation>Greece</documentation></annotation></enumeration>
          <enumeration value="GRD"><annotation><documentation>Grenada</documentation></annotation></enumeration>
          <enumeration value="GRL"><annotation><documentation>Greenland</documentation></annotation></enumeration>
          <enumeration value="GTM"><annotation><documentation>Guatemala</documentation></annotation></enumeration>
          <enumeration value="GUF"><annotation><documentation>French Guiana</documentation></annotation></enumeration>
          <enumeration value="GUM"><annotation><documentation>Guam</documentation></annotation></enumeration>
          <enumeration value="GUY"><annotation><documentation>Guyana</documentation></annotation></enumeration>
          <enumeration value="HKG"><annotation><documentation>Hong Kong</documentation></annotation></enumeration>
          <enumeration value="HMD"><annotation><documentation>Heard Island and McDonald Islands</documentation></annotation></enumeration>
          <enumeration value="HND"><annotation><documentation>Honduras</documentation></annotation></enumeration>
          <enumeration value="HRV"><annotation><documentation>Croatia</documentation></annotation></enumeration>
          <enumeration value="HTI"><annotation><documentation>Haiti</documentation></annotation></enumeration>
          <enumeration value="HUN"><annotation><documentation>Hungary</documentation></annotation></enumeration>
          <enumeration value="IDN"><annotation><documentation>Indonesia</documentation></annotation></enumeration>
          <enumeration value="IMN"><annotation><documentation>Isle of Man</documentation></annotation></enumeration>
          <enumeration value="IND"><annotation><documentation>India</documentation></annotation></enumeration>
          <enumeration value="IOT"><annotation><documentation>British Indian Ocean Territory</documentation></annotation></enumeration>
          <enumeration value="IRL"><annotation><documentation>Ireland</documentation></annotation></enumeration>
          <enumeration value="IRN"><annotation><documentation>Iran (Islamic Republic of)</documentation></annotation></enumeration>
          <enumeration value="IRQ"><annotation><documentation>Iraq</documentation></annotation></enumeration>
          <enumeration value="ISL"><annotation><documentation>Iceland</documentation></annotation></enumeration>
          <enumeration value="ISR"><annotation><documentation>Israel</documentation></annotation></enumeration>
          <enumeration value="ITA"><annotation><documentation>Italy</documentation></annotation></enumeration>
          <enumeration value="JAM"><annotation><documentation>Jamaica</documentation></annotation></enumeration>
          <enumeration value="JEY"><annotation><documentation>Jersey</documentation></annotation></enumeration>
          <enumeration value="JOR"><annotation><documentation>Jordan</documentation></annotation></enumeration>
          <enumeration value="JPN"><annotation><documentation>Japan</documentation></annotation></enumeration>
          <enumeration value="KAZ"><annotation><documentation>Kazakhstan</documentation></annotation></enumeration>
          <enumeration value="KEN"><annotation><documentation>Kenya</documentation></annotation></enumeration>
          <enumeration value="KGZ"><annotation><documentation>Kyrgyzstan</documentation></annotation></enumeration>
          <enumeration value="KHM"><annotation><documentation>Cambodia</documentation></annotation></enumeration>
          <enumeration value="KIR"><annotation><documentation>Kiribati</documentation></annotation></enumeration>
          <enumeration value="KNA"><annotation><documentation>Saint Kitts and Nevis</documentation></annotation></enumeration>
          <enumeration value="KOR"><annotation><documentation>Korea, Republic of</documentation></annotation></enumeration>
          <enumeration value="KWT"><annotation><documentation>Kuwait</documentation></annotation></enumeration>
          <enumeration value="LAO"><annotation><documentation>Lao People's Democratic Republic</documentation></annotation></enumeration>
          <enumeration value="LBN"><annotation><documentation>Lebanon</documentation></annotation></enumeration>
          <enumeration value="LBR"><annotation><documentation>Liberia</documentation></annotation></enumeration>
          <enumeration value="LBY"><annotation><documentation>Libya</documentation></annotation></enumeration>
          <enumeration value="LCA"><annotation><documentation>Saint Lucia</documentation></annotation></enumeration>
          <enumeration value="LIE"><annotation><documentation>Liechtenstein</documentation></annotation></enumeration>
          <enumeration value="LKA"><annotation><documentation>Sri Lanka</documentation></annotation></enumeration>
          <enumeration value="LSO"><annotation><documentation>Lesotho</documentation></annotation></enumeration>
```

```xml
<enumeration value="LTU"><annotation><documentation>Lithuania</documentation></annotation></enumeration>
<enumeration value="LUX"><annotation><documentation>Luxembourg</documentation></annotation></enumeration>
<enumeration value="LVA"><annotation><documentation>Latvia</documentation></annotation></enumeration>
<enumeration value="MAC"><annotation><documentation>Macao</documentation></annotation></enumeration>
<enumeration value="MAF"><annotation><documentation>Saint Martin (French part)</documentation></annotation></enumeration>
<enumeration value="MAR"><annotation><documentation>Morocco</documentation></annotation></enumeration>
<enumeration value="MCO"><annotation><documentation>Monaco</documentation></annotation></enumeration>
<enumeration value="MDA"><annotation><documentation>Moldova, Republic of</documentation></annotation></enumeration>
<enumeration value="MDG"><annotation><documentation>Madagascar</documentation></annotation></enumeration>
<enumeration value="MDV"><annotation><documentation>Maldives</documentation></annotation></enumeration>
<enumeration value="MEX"><annotation><documentation>Mexico</documentation></annotation></enumeration>
<enumeration value="MHL"><annotation><documentation>Marshall Islands</documentation></annotation></enumeration>
<enumeration value="MKD"><annotation><documentation>North Macedonia</documentation></annotation></enumeration>
<enumeration value="MLI"><annotation><documentation>Mali</documentation></annotation></enumeration>
<enumeration value="MLT"><annotation><documentation>Malta</documentation></annotation></enumeration>
<enumeration value="MMR"><annotation><documentation>Myanmar</documentation></annotation></enumeration>
<enumeration value="MNE"><annotation><documentation>Montenegro</documentation></annotation></enumeration>
<enumeration value="MNG"><annotation><documentation>Mongolia</documentation></annotation></enumeration>
<enumeration value="MNP"><annotation><documentation>Northern Mariana Islands</documentation></annotation></enumeration>
<enumeration value="MOZ"><annotation><documentation>Mozambique</documentation></annotation></enumeration>
<enumeration value="MRT"><annotation><documentation>Mauritania</documentation></annotation></enumeration>
<enumeration value="MSR"><annotation><documentation>Montserrat</documentation></annotation></enumeration>
<enumeration value="MTQ"><annotation><documentation>Martinique</documentation></annotation></enumeration>
<enumeration value="MUS"><annotation><documentation>Mauritius</documentation></annotation></enumeration>
<enumeration value="MWI"><annotation><documentation>Malawi</documentation></annotation></enumeration>
<enumeration value="MYS"><annotation><documentation>Malaysia</documentation></annotation></enumeration>
<enumeration value="MYT"><annotation><documentation>Mayotte</documentation></annotation></enumeration>
<enumeration value="NAM"><annotation><documentation>Namibia</documentation></annotation></enumeration>
<enumeration value="NCL"><annotation><documentation>New Caledonia</documentation></annotation></enumeration>
<enumeration value="NER"><annotation><documentation>Niger</documentation></annotation></enumeration>
<enumeration value="NFK"><annotation><documentation>Norfolk Island</documentation></annotation></enumeration>
<enumeration value="NGA"><annotation><documentation>Nigeria</documentation></annotation></enumeration>
<enumeration value="NIC"><annotation><documentation>Nicaragua</documentation></annotation></enumeration>
<enumeration value="NIU"><annotation><documentation>Niue</documentation></annotation></enumeration>
<enumeration value="NLD"><annotation><documentation>Netherlands</documentation></annotation></enumeration>
<enumeration value="NOR"><annotation><documentation>Norway</documentation></annotation></enumeration>
<enumeration value="NPL"><annotation><documentation>Nepal</documentation></annotation></enumeration>
<enumeration value="NRU"><annotation><documentation>Nauru</documentation></annotation></enumeration>
<enumeration value="NZL"><annotation><documentation>New Zealand</documentation></annotation></enumeration>
<enumeration value="OMN"><annotation><documentation>Oman</documentation></annotation></enumeration>
<enumeration value="PAK"><annotation><documentation>Pakistan</documentation></annotation></enumeration>
<enumeration value="PAN"><annotation><documentation>Panama</documentation></annotation></enumeration>
<enumeration value="PCN"><annotation><documentation>Pitcairn</documentation></annotation></enumeration>
<enumeration value="PER"><annotation><documentation>Peru</documentation></annotation></enumeration>
<enumeration value="PHL"><annotation><documentation>Philippines</documentation></annotation></enumeration>
<enumeration value="PLW"><annotation><documentation>Palau</documentation></annotation></enumeration>
<enumeration value="PNG"><annotation><documentation>Papua New Guinea</documentation></annotation></enumeration>
<enumeration value="POL"><annotation><documentation>Poland</documentation></annotation></enumeration>
<enumeration value="PRI"><annotation><documentation>Puerto Rico</documentation></annotation></enumeration>
<enumeration value="PRK"><annotation><documentation>Korea (Democratic People's Republic
of)</documentation></annotation></enumeration>
<enumeration value="PRT"><annotation><documentation>Portugal</documentation></annotation></enumeration>
<enumeration value="PRY"><annotation><documentation>Paraguay</documentation></annotation></enumeration>
```

```xml
            <enumeration value="PSE"><annotation><documentation>Palestine, State of</documentation></annotation></enumeration>

            <enumeration value="PYF"><annotation><documentation>French Polynesia</documentation></annotation></enumeration>

            <enumeration value="QAT"><annotation><documentation>Qatar</documentation></annotation></enumeration>

            <enumeration value="REU"><annotation><documentation>Reunion</documentation></annotation></enumeration>

            <enumeration value="ROU"><annotation><documentation>Romania</documentation></annotation></enumeration>

            <enumeration value="RUS"><annotation><documentation>Russian Federation</documentation></annotation></enumeration>

            <enumeration value="RWA"><annotation><documentation>Rwanda</documentation></annotation></enumeration>

            <enumeration value="SAU"><annotation><documentation>Saudi Arabia</documentation></annotation></enumeration>

            <enumeration value="SDN"><annotation><documentation>Sudan</documentation></annotation></enumeration>

            <enumeration value="SEN"><annotation><documentation>Senegal</documentation></annotation></enumeration>

            <enumeration value="SGP"><annotation><documentation>Singapore</documentation></annotation></enumeration>

            <enumeration value="SGS"><annotation><documentation>South Georgia and the South Sandwich
Islands</documentation></annotation></enumeration>

            <enumeration value="SHN"><annotation><documentation>Saint Helena, Ascension and Tristan da
Cunha</documentation></annotation></enumeration>

            <enumeration value="SJM"><annotation><documentation>Svalbard and Jan Mayen</documentation></annotation></enumeration>

            <enumeration value="SLB"><annotation><documentation>Solomon Islands</documentation></annotation></enumeration>

            <enumeration value="SLE"><annotation><documentation>Sierra Leone</documentation></annotation></enumeration>

            <enumeration value="SLV"><annotation><documentation>El Salvador</documentation></annotation></enumeration>

            <enumeration value="SMR"><annotation><documentation>San Marino</documentation></annotation></enumeration>

            <enumeration value="SOM"><annotation><documentation>Somalia</documentation></annotation></enumeration>

            <enumeration value="SPM"><annotation><documentation>Saint Pierre and Miquelon</documentation></annotation></enumeration>

            <enumeration value="SRB"><annotation><documentation>Serbia</documentation></annotation></enumeration>

            <enumeration value="SSD"><annotation><documentation>South Sudan</documentation></annotation></enumeration>

            <enumeration value="STP"><annotation><documentation>Sao Tome and Principe</documentation></annotation></enumeration>

            <enumeration value="SUR"><annotation><documentation>Suriname</documentation></annotation></enumeration>

            <enumeration value="SVK"><annotation><documentation>Slovakia</documentation></annotation></enumeration>

            <enumeration value="SVN"><annotation><documentation>Slovenia</documentation></annotation></enumeration>

            <enumeration value="SWE"><annotation><documentation>Sweden</documentation></annotation></enumeration>

            <enumeration value="SWZ"><annotation><documentation>Eswatini</documentation></annotation></enumeration>

            <enumeration value="SXM"><annotation><documentation>Sint Maarten (Dutch part)</documentation></annotation></enumeration>

            <enumeration value="SYC"><annotation><documentation>Seychelles</documentation></annotation></enumeration>

            <enumeration value="SYR"><annotation><documentation>Syrian Arab Republic</documentation></annotation></enumeration>

            <enumeration value="TCA"><annotation><documentation>Turks and Caicos Islands</documentation></annotation></enumeration>

            <enumeration value="TCD"><annotation><documentation>Chad</documentation></annotation></enumeration>

            <enumeration value="TGO"><annotation><documentation>Togo</documentation></annotation></enumeration>

            <enumeration value="THA"><annotation><documentation>Thailand</documentation></annotation></enumeration>

            <enumeration value="TJK"><annotation><documentation>Tajikistan</documentation></annotation></enumeration>

            <enumeration value="TKL"><annotation><documentation>Tokelau</documentation></annotation></enumeration>

            <enumeration value="TKM"><annotation><documentation>Turkmenistan</documentation></annotation></enumeration>

            <enumeration value="TLS"><annotation><documentation>Timor-Leste</documentation></annotation></enumeration>

            <enumeration value="TON"><annotation><documentation>Tonga</documentation></annotation></enumeration>

            <enumeration value="TTO"><annotation><documentation>Trinidad and Tobago</documentation></annotation></enumeration>

            <enumeration value="TUN"><annotation><documentation>Tunisia</documentation></annotation></enumeration>

            <enumeration value="TUR"><annotation><documentation>Turkey</documentation></annotation></enumeration>

            <enumeration value="TUV"><annotation><documentation>Tuvalu</documentation></annotation></enumeration>

            <enumeration value="TWN"><annotation><documentation>Taiwan, Province of China</documentation></annotation></enumeration>

            <enumeration value="TZA"><annotation><documentation>Tanzania, United Republic of</documentation></annotation></enumeration>

            <enumeration value="UGA"><annotation><documentation>Uganda</documentation></annotation></enumeration>

            <enumeration value="UKR"><annotation><documentation>Ukraine</documentation></annotation></enumeration>

            <enumeration value="UMI"><annotation><documentation>United States Minor Outlying
Islands</documentation></annotation></enumeration>

            <enumeration value="UNK"><annotation><documentation>Kosovo</documentation></annotation></enumeration>

            <enumeration value="URY"><annotation><documentation>Uruguay</documentation></annotation></enumeration>
```

```xml
            <enumeration value="USA"><annotation><documentation>United States of America</documentation></annotation></enumeration>

            <enumeration value="UZB"><annotation><documentation>Uzbekistan</documentation></annotation></enumeration>

            <enumeration value="VAT"><annotation><documentation>Holy See</documentation></annotation></enumeration>

            <enumeration value="VCT"><annotation><documentation>Saint Vincent and the Grenadines</documentation></annotation></enumeration>

            <enumeration value="VEN"><annotation><documentation>Venezuela (Bolivarian Republic of)</documentation></annotation></enumeration>

            <enumeration value="VGB"><annotation><documentation>Virgin Islands (British)</documentation></annotation></enumeration>

            <enumeration value="VIR"><annotation><documentation>Virgin Islands (U.S.)</documentation></annotation></enumeration>

            <enumeration value="VNM"><annotation><documentation>Viet Nam</documentation></annotation></enumeration>

            <enumeration value="VUT"><annotation><documentation>Vanuatu</documentation></annotation></enumeration>

            <enumeration value="WLF"><annotation><documentation>Wallis and Futuna</documentation></annotation></enumeration>

            <enumeration value="WSM"><annotation><documentation>Samoa</documentation></annotation></enumeration>

            <enumeration value="YEM"><annotation><documentation>Yemen</documentation></annotation></enumeration>

            <enumeration value="ZAF"><annotation><documentation>South Africa</documentation></annotation></enumeration>

            <enumeration value="ZMB"><annotation><documentation>Zambia</documentation></annotation></enumeration>

            <enumeration value="ZWE"><annotation><documentation>Zimbabwe</documentation></annotation></enumeration>

        </restriction>

    </simpleType>

    <simpleType name="CurrencyCodeSType">

        <annotation>

            <documentation>Currency codes from ISO 4217 standard</documentation>

        </annotation>

        <restriction base="string">

            <enumeration value="AED"><annotation><documentation>United Arab Emirates Dirham</documentation></annotation></enumeration>

            <enumeration value="AFN"><annotation><documentation>Afghanistan Afghani</documentation></annotation></enumeration>

            <enumeration value="AMD"><annotation><documentation>Armenia Dram</documentation></annotation></enumeration>

            <enumeration value="ANG"><annotation><documentation>Netherlands Antilles Guilder</documentation></annotation></enumeration>

            <enumeration value="AOA"><annotation><documentation>Angola Kwanza</documentation></annotation></enumeration>

            <enumeration value="ARS"><annotation><documentation>Argentina Peso</documentation></annotation></enumeration>

            <enumeration value="AUD"><annotation><documentation>Australia Dollar</documentation></annotation></enumeration>

            <enumeration value="AWG"><annotation><documentation>Aruba Guilder</documentation></annotation></enumeration>

            <enumeration value="AZN"><annotation><documentation>Azerbaijan Manat</documentation></annotation></enumeration>

            <enumeration value="BAM"><annotation><documentation>Bosnia and Herzegovina Convertible
Mark</documentation></annotation></enumeration>

            <enumeration value="BBD"><annotation><documentation>Barbados Dollar</documentation></annotation></enumeration>

            <enumeration value="BDT"><annotation><documentation>Bangladesh Taka</documentation></annotation></enumeration>

            <enumeration value="BGN"><annotation><documentation>Bulgaria Lev</documentation></annotation></enumeration>

            <enumeration value="BHD"><annotation><documentation>Bahrain Dinar</documentation></annotation></enumeration>

            <enumeration value="BIF"><annotation><documentation>Burundi Franc</documentation></annotation></enumeration>

            <enumeration value="BMD"><annotation><documentation>Bermuda Dollar</documentation></annotation></enumeration>

            <enumeration value="BND"><annotation><documentation>Brunei Darussalam Dollar</documentation></annotation></enumeration>

            <enumeration value="BOB"><annotation><documentation>Bolivia Boliviano</documentation></annotation></enumeration>

            <enumeration value="BRL"><annotation><documentation>Brazil Real</documentation></annotation></enumeration>

            <enumeration value="BSD"><annotation><documentation>Bahamas Dollar</documentation></annotation></enumeration>

            <enumeration value="BTN"><annotation><documentation>Bhutan Ngultrum</documentation></annotation></enumeration>

            <enumeration value="BWP"><annotation><documentation>Botswana Pula</documentation></annotation></enumeration>

            <enumeration value="BYN"><annotation><documentation>Belarus Ruble</documentation></annotation></enumeration>

            <enumeration value="BZD"><annotation><documentation>Belize Dollar</documentation></annotation></enumeration>

            <enumeration value="CAD"><annotation><documentation>Canada Dollar</documentation></annotation></enumeration>

            <enumeration value="CDF"><annotation><documentation>Congo/Kinshasa Franc</documentation></annotation></enumeration>

            <enumeration value="CHF"><annotation><documentation>Switzerland Franc</documentation></annotation></enumeration>

            <enumeration value="CLP"><annotation><documentation>Chile Peso</documentation></annotation></enumeration>

            <enumeration value="CNY"><annotation><documentation>China Yuan Renminbi</documentation></annotation></enumeration>

            <enumeration value="COP"><annotation><documentation>Colombia Peso</documentation></annotation></enumeration>
```

```xml
<enumeration value="CRC"><annotation><documentation>Costa Rica Colon</documentation></annotation></enumeration>
<enumeration value="CUC"><annotation><documentation>Cuba Convertible Peso</documentation></annotation></enumeration>
<enumeration value="CUP"><annotation><documentation>Cuba Peso</documentation></annotation></enumeration>
<enumeration value="CVE"><annotation><documentation>Cape Verde Escudo</documentation></annotation></enumeration>
<enumeration value="CZK"><annotation><documentation>Czech Republic Koruna</documentation></annotation></enumeration>
<enumeration value="DJF"><annotation><documentation>Djibouti Franc</documentation></annotation></enumeration>
<enumeration value="DKK"><annotation><documentation>Denmark Krone</documentation></annotation></enumeration>
<enumeration value="DOP"><annotation><documentation>Dominican Republic Peso</documentation></annotation></enumeration>
<enumeration value="DZD"><annotation><documentation>Algeria Dinar</documentation></annotation></enumeration>
<enumeration value="EGP"><annotation><documentation>Egypt Pound</documentation></annotation></enumeration>
<enumeration value="ERN"><annotation><documentation>Eritrea Nakfa</documentation></annotation></enumeration>
<enumeration value="ETB"><annotation><documentation>Ethiopia Birr</documentation></annotation></enumeration>
<enumeration value="EUR"><annotation><documentation>Euro Member Countries</documentation></annotation></enumeration>
<enumeration value="FJD"><annotation><documentation>Fiji Dollar</documentation></annotation></enumeration>
<enumeration value="FKP"><annotation><documentation>Falkland Islands (Malvinas) Pound</documentation></annotation></enumeration>
<enumeration value="GBP"><annotation><documentation>United Kingdom Pound</documentation></annotation></enumeration>
<enumeration value="GEL"><annotation><documentation>Georgia Lari</documentation></annotation></enumeration>
<enumeration value="GGP"><annotation><documentation>Guernsey Pound</documentation></annotation></enumeration>
<enumeration value="GHS"><annotation><documentation>Ghana Cedi</documentation></annotation></enumeration>
<enumeration value="GIP"><annotation><documentation>Gibraltar Pound</documentation></annotation></enumeration>
<enumeration value="GMD"><annotation><documentation>Gambia Dalasi</documentation></annotation></enumeration>
<enumeration value="GNF"><annotation><documentation>Guinea Franc</documentation></annotation></enumeration>
<enumeration value="GTQ"><annotation><documentation>Guatemala Quetzal</documentation></annotation></enumeration>
<enumeration value="GYD"><annotation><documentation>Guyana Dollar</documentation></annotation></enumeration>
<enumeration value="HKD"><annotation><documentation>Hong Kong Dollar</documentation></annotation></enumeration>
<enumeration value="HNL"><annotation><documentation>Honduras Lempira</documentation></annotation></enumeration>
<enumeration value="HRK"><annotation><documentation>Croatia Kuna</documentation></annotation></enumeration>
<enumeration value="HTG"><annotation><documentation>Haiti Gourde</documentation></annotation></enumeration>
<enumeration value="HUF"><annotation><documentation>Hungary Forint</documentation></annotation></enumeration>
<enumeration value="IDR"><annotation><documentation>Indonesia Rupiah</documentation></annotation></enumeration>
<enumeration value="ILS"><annotation><documentation>Israel Shekel</documentation></annotation></enumeration>
<enumeration value="IMP"><annotation><documentation>Isle of Man Pound</documentation></annotation></enumeration>
<enumeration value="INR"><annotation><documentation>India Rupee</documentation></annotation></enumeration>
<enumeration value="IQD"><annotation><documentation>Iraq Dinar</documentation></annotation></enumeration>
<enumeration value="IRR"><annotation><documentation>Iran Rial</documentation></annotation></enumeration>
<enumeration value="ISK"><annotation><documentation>Iceland Krona</documentation></annotation></enumeration>
<enumeration value="JEP"><annotation><documentation>Jersey Pound</documentation></annotation></enumeration>
<enumeration value="JMD"><annotation><documentation>Jamaica Dollar</documentation></annotation></enumeration>
<enumeration value="JOD"><annotation><documentation>Jordan Dinar</documentation></annotation></enumeration>
<enumeration value="JPY"><annotation><documentation>Japan Yen</documentation></annotation></enumeration>
<enumeration value="KES"><annotation><documentation>Kenya Shilling</documentation></annotation></enumeration>
<enumeration value="KGS"><annotation><documentation>Kyrgyzstan Som</documentation></annotation></enumeration>
<enumeration value="KHR"><annotation><documentation>Cambodia Riel</documentation></annotation></enumeration>
<enumeration value="KMF"><annotation><documentation>Comorian Franc</documentation></annotation></enumeration>
<enumeration value="KPW"><annotation><documentation>Korea (North) Won</documentation></annotation></enumeration>
<enumeration value="KRW"><annotation><documentation>Korea (South) Won</documentation></annotation></enumeration>
<enumeration value="KWD"><annotation><documentation>Kuwait Dinar</documentation></annotation></enumeration>
<enumeration value="KYD"><annotation><documentation>Cayman Islands Dollar</documentation></annotation></enumeration>
<enumeration value="KZT"><annotation><documentation>Kazakhstan Tenge</documentation></annotation></enumeration>
<enumeration value="LAK"><annotation><documentation>Laos Kip</documentation></annotation></enumeration>
<enumeration value="LBP"><annotation><documentation>Lebanon Pound</documentation></annotation></enumeration>
<enumeration value="LKR"><annotation><documentation>Sri Lanka Rupee</documentation></annotation></enumeration>
<enumeration value="LRD"><annotation><documentation>Liberia Dollar</documentation></annotation></enumeration>
```

```xml
<enumeration value="LSL"><annotation><documentation>Lesotho Loti</documentation></annotation></enumeration>
<enumeration value="LYD"><annotation><documentation>Libya Dinar</documentation></annotation></enumeration>
<enumeration value="MAD"><annotation><documentation>Morocco Dirham</documentation></annotation></enumeration>
<enumeration value="MDL"><annotation><documentation>Moldova Leu</documentation></annotation></enumeration>
<enumeration value="MGA"><annotation><documentation>Madagascar Ariary</documentation></annotation></enumeration>
<enumeration value="MKD"><annotation><documentation>Macedonia Denar</documentation></annotation></enumeration>
<enumeration value="MMK"><annotation><documentation>Myanmar (Burma) Kyat</documentation></annotation></enumeration>
<enumeration value="MNT"><annotation><documentation>Mongolia Tughrik</documentation></annotation></enumeration>
<enumeration value="MOP"><annotation><documentation>Macau Pataca</documentation></annotation></enumeration>
<enumeration value="MRU"><annotation><documentation>Mauritania Ouguiya</documentation></annotation></enumeration>
<enumeration value="MUR"><annotation><documentation>Mauritius Rupee</documentation></annotation></enumeration>
<enumeration value="MVR"><annotation><documentation>Maldives (Maldive Islands) Rufiyaa</documentation></annotation></enumeration>
<enumeration value="MWK"><annotation><documentation>Malawi Kwacha</documentation></annotation></enumeration>
<enumeration value="MXN"><annotation><documentation>Mexico Peso</documentation></annotation></enumeration>
<enumeration value="MYR"><annotation><documentation>Malaysia Ringgit</documentation></annotation></enumeration>
<enumeration value="MZN"><annotation><documentation>Mozambique Metical</documentation></annotation></enumeration>
<enumeration value="NAD"><annotation><documentation>Namibia Dollar</documentation></annotation></enumeration>
<enumeration value="NGN"><annotation><documentation>Nigeria Naira</documentation></annotation></enumeration>
<enumeration value="NIO"><annotation><documentation>Nicaragua Cordoba</documentation></annotation></enumeration>
<enumeration value="NOK"><annotation><documentation>Norway Krone</documentation></annotation></enumeration>
<enumeration value="NPR"><annotation><documentation>Nepal Rupee</documentation></annotation></enumeration>
<enumeration value="NZD"><annotation><documentation>New Zealand Dollar</documentation></annotation></enumeration>
<enumeration value="OMR"><annotation><documentation>Oman Rial</documentation></annotation></enumeration>
<enumeration value="PAB"><annotation><documentation>Panama Balboa</documentation></annotation></enumeration>
<enumeration value="PEN"><annotation><documentation>Peru Sol</documentation></annotation></enumeration>
<enumeration value="PGK"><annotation><documentation>Papua New Guinea Kina</documentation></annotation></enumeration>
<enumeration value="PHP"><annotation><documentation>Philippines Peso</documentation></annotation></enumeration>
<enumeration value="PKR"><annotation><documentation>Pakistan Rupee</documentation></annotation></enumeration>
<enumeration value="PLN"><annotation><documentation>Poland Zloty</documentation></annotation></enumeration>
<enumeration value="PYG"><annotation><documentation>Paraguay Guarani</documentation></annotation></enumeration>
<enumeration value="QAR"><annotation><documentation>Qatar Riyal</documentation></annotation></enumeration>
<enumeration value="RON"><annotation><documentation>Romania Leu</documentation></annotation></enumeration>
<enumeration value="RSD"><annotation><documentation>Serbia Dinar</documentation></annotation></enumeration>
<enumeration value="RUB"><annotation><documentation>Russia Ruble</documentation></annotation></enumeration>
<enumeration value="RWF"><annotation><documentation>Rwanda Franc</documentation></annotation></enumeration>
<enumeration value="SAR"><annotation><documentation>Saudi Arabia Riyal</documentation></annotation></enumeration>
<enumeration value="SBD"><annotation><documentation>Solomon Islands Dollar</documentation></annotation></enumeration>
<enumeration value="SCR"><annotation><documentation>Seychelles Rupee</documentation></annotation></enumeration>
<enumeration value="SDG"><annotation><documentation>Sudan Pound</documentation></annotation></enumeration>
<enumeration value="SEK"><annotation><documentation>Sweden Krona</documentation></annotation></enumeration>
<enumeration value="SGD"><annotation><documentation>Singapore Dollar</documentation></annotation></enumeration>
<enumeration value="SHP"><annotation><documentation>Saint Helena Pound</documentation></annotation></enumeration>
<enumeration value="SLL"><annotation><documentation>Sierra Leone Leone</documentation></annotation></enumeration>
<enumeration value="SOS"><annotation><documentation>Somalia Shilling</documentation></annotation></enumeration>
<enumeration value="SPL"><annotation><documentation>Seborga Luigino</documentation></annotation></enumeration>
<enumeration value="SRD"><annotation><documentation>Suriname Dollar</documentation></annotation></enumeration>
<enumeration value="STN"><annotation><documentation>Sao Tome and Principe Dobra</documentation></annotation></enumeration>
<enumeration value="SVC"><annotation><documentation>El Salvador Colon</documentation></annotation></enumeration>
<enumeration value="SYP"><annotation><documentation>Syria Pound</documentation></annotation></enumeration>
<enumeration value="SZL"><annotation><documentation>eSwatini Lilangeni</documentation></annotation></enumeration>
<enumeration value="THB"><annotation><documentation>Thailand Baht</documentation></annotation></enumeration>
<enumeration value="TJS"><annotation><documentation>Tajikistan Somoni</documentation></annotation></enumeration>
<enumeration value="TMT"><annotation><documentation>Turkmenistan Manat</documentation></annotation></enumeration>
```

```xml
            <enumeration value="TND"><annotation><documentation>Tunisia Dinar</documentation></annotation></enumeration>

            <enumeration value="TOP"><annotation><documentation>Tonga Pa'anga</documentation></annotation></enumeration>

            <enumeration value="TRY"><annotation><documentation>Turkey Lira</documentation></annotation></enumeration>

            <enumeration value="TTD"><annotation><documentation>Trinidad and Tobago Dollar</documentation></annotation></enumeration>

            <enumeration value="TVD"><annotation><documentation>Tuvalu Dollar</documentation></annotation></enumeration>

            <enumeration value="TWD"><annotation><documentation>Taiwan New Dollar</documentation></annotation></enumeration>

            <enumeration value="TZS"><annotation><documentation>Tanzania Shilling</documentation></annotation></enumeration>

            <enumeration value="UAH"><annotation><documentation>Ukraine Hryvnia</documentation></annotation></enumeration>

            <enumeration value="UGX"><annotation><documentation>Uganda Shilling</documentation></annotation></enumeration>

            <enumeration value="USD"><annotation><documentation>United States Dollar</documentation></annotation></enumeration>

            <enumeration value="UYU"><annotation><documentation>Uruguay Peso</documentation></annotation></enumeration>

            <enumeration value="UZS"><annotation><documentation>Uzbekistan Som</documentation></annotation></enumeration>

            <enumeration value="VEF"><annotation><documentation>Venezuela Bolivar</documentation></annotation></enumeration>

            <enumeration value="VND"><annotation><documentation>Viet Nam Dong</documentation></annotation></enumeration>

            <enumeration value="VUV"><annotation><documentation>Vanuatu Vatu</documentation></annotation></enumeration>

            <enumeration value="WST"><annotation><documentation>Samoa Tala</documentation></annotation></enumeration>

            <enumeration value="XAF"><annotation><documentation>Communaute Financiere Africaine (BEAC) CFA Franc
BEAC</documentation></annotation></enumeration>

            <enumeration value="XCD"><annotation><documentation>East Caribbean Dollar</documentation></annotation></enumeration>

            <enumeration value="XDR"><annotation><documentation>International Monetary Fund (IMF) Special Drawing
Rights</documentation></annotation></enumeration>

            <enumeration value="XOF"><annotation><documentation>Communaute Financiere Africaine (BCEAO)
Franc</documentation></annotation></enumeration>

            <enumeration value="XPF"><annotation><documentation>Comptoirs Francais du Pacifique (CFP)
Franc</documentation></annotation></enumeration>

            <enumeration value="YER"><annotation><documentation>Yemen Rial</documentation></annotation></enumeration>

            <enumeration value="ZAR"><annotation><documentation>South Africa Rand</documentation></annotation></enumeration>

            <enumeration value="ZMW"><annotation><documentation>Zambia Kwacha</documentation></annotation></enumeration>

            <enumeration value="ZWD"><annotation><documentation>Zimbabwe Dollar</documentation></annotation></enumeration>

        </restriction>

    </simpleType>

</schema>
```