5. Signing Process

In this section, step-by-step instructions will be demonstrated towards acquiring a Signed Invoice.

5.1 SHA-256 Hash - Hashing algorithm

The main reason for using SHA-256 is to strengthen the security and protect the data by ensuring it doesn't have any known vulnerabilities that make it insecure, and it has not been "broken" unlike some other popular hashing algorithms.
The output of a hashing algorithm SHA256 will always be the same, consisting of 256 bits (32 bytes), which is displayed as 64 alphanumeric characters.

5.2 Signing steps

Step 1: Generate Invoice Hash

To generate the invoice hash below steps can be followed:

1. Open the invoice XML file.
2. Remove the tags mentioned in the table below using the XPath.
3. Remove the XML version.
4. Canonicalize the Invoice using the C14N11 standard
5. Hash the new invoice body using SHA-256 (output). e.g.:a11b6fe587a50f7daffe3a7fb42dcccf-32b43ee9b37d9f252d04243e54c11a3f
6. Encode the hashed invoice using base64 (output)
   Using HEX-to Base64 Encoder
   e.g.:oRtv5YelD32v/jp/tC3MzzK0PumzfZ8ILQQkPlTBGj8=
   Note: All these values will be used in later steps.

Tags to be removed from invoice	XPath, Use this path to find the target tag
UBLExtension	*[local-name()='Invoice']//*[local-name()='UBLExtensions']
QR	//*[local-name()='AdditionalDocumentReference'] [cbc:ID[normalize-space(text()) = 'QR']]
Signature	*[local-name()='Invoice']//*[local-name()='Signature']

Step 2: Generate Digital Signature

1. Sign the generated invoice hash with ECDSA using the private key (output). (e.g.:MEQCIGvLa-1f3uMCe0AidKUWJ5ghMiDMRcC0qO78ntcTKVOYgAiAKBkX+uuFhbIcye3JznNa45qH1twlLFu/qPzEQ9HMNLw==)
   Note: This value will be used in later steps.

Values to be used
Generated Invoice Hash from Step 1 (not encode)
Private key

Step 3: Generate Certificate Hash

1. Hash the certificate using SHA-256 (output). e.g.:69a95fc237b42714dc4457a33b94cc452fd9f-110504c683c401144d9544894fb
2. Encode the hashed certificate using base64 (output).

e.g.:NjlhOTVmYzIzN2I0MjcxNGRjNDQ1N2EzM2I5NGNjNDUyZmQ5ZjExMDUwNGM2ODNjNDAx-MTQ0ZDk1NDQ4OTRmYg==

Note: All these values will be used in later steps.

Values to be used
Certificate

Step 4: Populate the Signed Properties Output

1. Open the original invoice (not updated in Step 1).
2. Remove the tags UBLExtentions, QR and Signature (refer to their XPath mentioned in Step 1).
3. Replace the removed tags mentioned below with the same tags but without values (to be filled in later steps).
4. Refer to the below table to fill mentioned fields with their corresponding values using the related XPath.

Notes:

- The original invoice XML file is used in this step (not the one updated in the 1st step).
- Populated Signed Properties will be used in the next step.

Fields	Values	XPath
DigestValue	Encoded certificate hashed from Step 3	/Invoice/ext:UBLExtensions/ext:UBLExtension/ext:ExtensionContent/sig:UBLDocumentSignatures/sac:SignatureInformation/ds:Signature/ds:Object/xades:QualifyingProperties/xades:SignedProperties/xades:SignedSignatureProperties/xades:SigningCertificate/xades:Cert/xades:CertDigest/ds:DigestValue
SigningTime	Sign timestamp as current datetime	/Invoice/ext:UBLExtensions/ext:UBLExtension/ext:ExtensionContent/sig:UBLDocumentSignatures/sac:SignatureInformation/ds:Signature/ds:Object/xades:QualifyingProperties/xades:SignedProperties/xades:SignedSignatureProperties/xades:SigningTime
X509IssuerName	Certificate issuer name From the certificate (decoded)	/Invoice/ext:UBLExtensions/ext:UBLExtension/ext:ExtensionContent/sig:UBLDocumentSignatures/sac:SignatureInformation/ds:Signature/ds:Object/xades:QualifyingProperties/xades:SignedProperties/xades:SignedSignatureProperties/xades:SigningCertificate/xades:Cert/xades:IssuerSerial/ds:X509IssuerName
X509SerialNumber	Certificate serial number From the certificate (decoded)	/Invoice/ext:UBLExtensions/ext:UBLExtension/ext:ExtensionContent/sig:UBLDocumentSignatures/sac:SignatureInformation/ds:Signature/ds:Object/xades:QualifyingProperties/xades:SignedProperties//xades:SignedSignatureProperties/xades:SigningCertificate/xades:Cert/xades:IssuerSerial/ds:X509SerialNumber

Step 5: Generate Signed Properties Hash

1. Get the properties tag only using the XPath (don't remove from XML file).
2. Linearize the XML block (properties tag) and remove the spaces
3. Hash the property tag using SHA-256 (output). e.g.:99282555b5d79209be5883cc23eb234cd01b-d33ea7d54d88f491248d33e321f1
4. Encode the hashed property tag using HEX-to Base64 Encoder (output). E.g.:mSglVbXXkgm+WIP-MI+sjTNAb0z6n1U2I9JEkjTPjIfE=

Values to be used	XPath
Populated Signed Properties from Step 4	/Invoice/ext:UBLExtensions/ext:UBLExtension/ext:ExtensionContent/sig:UBLDocumentSignatures/sac:SignatureInformation/ds:Signature/ds:-Object/xades:QualifyingProperties/xades:SignedProperties

Step 6: Populate The UBL Extensions Output

1. Use the invoice XML file acquired from Step 4.
2. Refer to the below table to fill mentioned fields with their corresponding values using the related XPath.

Note: We are using the updated invoice acquired from the Step 4

Fields	Values	XPath
SignatureValue	Digital Signature from Step 2	/Invoice/ext:UBLExtensions/ext:UBLExtension/ext:Extension-Content/sig:UBLDocumentSignatures/sac:SignatureInforma-tion/ds:Signature/ds:SignatureValue
X509Certificate	Certificate	/Invoice/ext:UBLExtensions/ext:UBLExtension/ext:Extension-Content/sig:UBLDocumentSignatures/sac:SignatureInforma-tion/ds:Signature/ds:KeyInfo/ds:X509Data/ds:X509Certificate
DigestValue	Encoded signed Properties hash from Step 5	/Invoice/ext:UBLExtensions/ext:UBLExtension/ext:Extension-Content/sig:UBLDocumentSignatures/sac:SignatureInforma-tion/ds:Signature/ds:SignedInfo/ds:Reference[@URI='#xades-SignedProperties']/ds:DigestValue
DigestValue	Encoded invoice hash from Step 1	/Invoice/ext:UBLExtensions/ext:UBLExtension/ext:Extension-Content/sig:UBLDocumentSignatures/sac:SignatureInforma-tion/ds:Signature/ds:SignedInfo/ds:Reference[@Id='invoice-SignedData']/ds:DigestValue

To finalize the signing, refer to the Section 6 - QR code.

Additional reference: Openssl commands & urls for support

Openssl:

- Hash function: openssl dgst -sha256 <xml_file_name>
- Generate private key: openssl ecparam -name secp256k1 -genkey -noout -out PrivateKey.pem
- Generate public key: openssl ec -in PrivateKey.pem -pubout -conv_form compressed -out PublicKey.pem
- Generate csr: openssl req -new -sha256 -key privateKey.pem -extensions v3_req -config config.cnf -out taxpayer.csr

URLs:

- XML Canonical online tool: http://www.soapclient.com/xmlcanon.html
- XPATHER ONLINE TOOL: http://xpather.com/
- Hashing online tool: https://emn178.github.io/online-tools/sha256.html
- Hex to base 64 online: https://base64.guru/converter/encode/hex
- ENCODER BASE64 online: https://www.base64encode.org/
- ECDSA SIGN online: https://8gwifi.org/ecsignverify.jsp
- CSR and certificate decoder online: https://certlogik.com/decoder/
- TEXT to HEXA online: https://www.online-toolz.com/tools/text-hex-convertor.php
- private key decoder online: http://certificate.fyicenter.com/2145_FYIcenter_Public_Private_Key_Decoder_and_Viewer.html#Result
- TLV QR decoder online: https://emvlab.org/tlvutils/

6. QR code

Structure of the QR code For Electronic Tax Invoices

It is mandatory to generate and print QR code encoded in Base64 format with up to 700 characters that must contain the fields specified in the below table as per Annex (2) of the Controls, Requirements, Technical Specifications and Procedural Rules for Implementing the Provisions of the E-Invoicing Regulation.

The QR code fields shall be encoded in Tag-Length-Value (TLV) format with the tag values specified in the "Tag" column of the adjacent table.

The TLV encoding shall be as follows:

- Tag: The tag value as mentioned above stored in one byte.
- Length: The length of the byte array resulted from the UTF8 encoding of the field value. The length shall be stored in one byte.
- Value: The byte array resulting from the UTF8 encoding of the field value.

Field Definition for the QR Code

Description	Tag	Enforcement date
Seller's name	1	December 4th, 2021
VAT registration number of the seller	2	
Time stamp of the invoice (date and time)	3	
Invoice total (with VAT)	4	
VAT total	5	
Hash of XML invoice	6	Starting Jan 1st, 2023 in waves
ECDSA signature	7	
ECDSA public key	8	
For Simplified Tax Invoices and their associated notes, the ECDSA signature of the cryptographic stamp's public key by ZATCA's technical CA	9	

6.1 TLV - TAG - LENGTH - VALUE construction and file format

- QR code is the base64 encoded TLV (Tag, Length, Value)
- Type/Tag-Length-Value (TLV) is an encoding scheme used in many communication protocols to encode data. A TLV-encoded message has a defined structure which consists of 3 sections/parts, see Figure (1). Those are:
  - Code of the message type (T) - 1Byte
  - Message value length (L) - 1Byte
  - Message value itself. (V) - Variable
- The Tag/Type and Length are of fixed sizes of 1bytes while the value has a variable size.
- As the general idea behind encoding is to transform abstract data into a stream of bits, using TLV, there are different sets of encoding rules that can be used according to the Abstract Syntax Notation Version 1 (ASN.1). We are using a simple version of Basic Encoding Rules (BER).

6.2 Creation of TLV QR code

Description	Tag	Length	Value	Hex
Seller's name	1	23	Ahmed Mohamed AL Ahmady	41 68 6d 65 64 20 4d 6f 68 61 6d 65 64 20 41 4c 20 41 68 6d 61 64 79
VAT registration number of the seller	2	15	301121971500003	33 30 31 31 32 31 39 37 31 35 30 30 30 30 33
Time stamp of the invoice (date and time)	3	20	2022-03-13T14:40:40Z	32 30 32 32 2d 30 33 2d 31 33 54 31 34 3a 34 30 3a 34 30 5a
Invoice total (with VAT)	4	6	1108.90	31 31 30 38 2e 39 30
VAT total	5	45	114.90	31 31 34 2e 39 30
Hash of XML invoice	6	44	QnVEexW4nWv4CaE39a/ 66Jp/ OXO/evHQ8pDI-G7weq/4=	51 6e 56 45 65 78 57 34 6e 57 76 34 43 61 45 33 39 61 2f 36 36 4a 70 2f 4f 58 4f 2f 65 76 48 51 38 70 44 6c 47 37 77 65 71 2f 34 3d 20
ECDSA signature	7	192	4d 45 55 43 49 51 44 35 7a 78 79 58 4f 42 37 4e 76 57 66 36 32 72 56 45 5a 41 59 55 37 31 6a 70 79 39 48 45 45 6e 5a 30 71 39 4f 39 36 77 72 4c 36 51 49 67 51 4a 7a 43 47 48 62 77 36 59 42 48 4c 59 56 64 4f 31 77 6e 55 68 42 67 4b 6d 38 6a 4d 54 79 76 63 6b 39 4d 2b 72 50 39 78 59 59 3d	4d 45 55 43 49 51 44 35 7a 78 79 58 4f 42 37 4e 76 57 66 36 32 72 56 45 5a 41 59 55 37 31 6a 70 79 39 48 45 45 6e 5a 30 71 39 4f 39 36 77 72 4c 36 51 49 67 51 4a 7a 43 47 48 62 77 36 59 42 48 4c 59 56 64 4f 31 77 6e 55 68 42 67 4b 6d 38 6a 4d 54 79 76 63 6b 39 4d 2b 72 50 39 78 59 59 3d
ECDSA public key	8	48	30 56 30 10 06 07 2a ef bf bd 48 ef bf bd 3d 02 01 06 05 2b ef bf bd 04	30 56 30 10 06 07 2a ef bf bd 48 ef bf bd 3d 02 01 06 05 2b ef bf bd 04
For Simplified Tax Invoices and their associated notes, the ECDSA signature of the cryptographic stamp's public key by ZATCA's technical CA	9	144	30 46 02 21 00 ee 61 d3 eb 28 3c e6 3b 50 19 6a 77 33 bb 4f 4f b2 64 db ec ec bd 51 c6 b3 76 d4 e5 9e d8 13 af 02 21 00 fa d1 e6 d0 6a 66 23 62 f7 5e 6e 71 63 35 fc 78 5f 87 68 a7 b2 ec 10 11 42 35 2b 0b 63 42 05 69	30 46 02 21 00 ee 61 d3 eb 28 3c e6 3b 50 19 6a 77 33 bb 4f 4f b2 64 db ec ec bd 51 c6 b3 76 d4 e5 9e d8 13 af 02 21 00 fa d1 e6 d0 6a 66 23 62 f7 5e 6e 71 63 35 fc 78 5f 87 68 a7 b2 ec 10 11 42 35 2b 0b 63 42 05 69

XML Elements for QR code:

Description	Tag	XML element
Tag1	Seller's name	/Invoice/cac:AccountingSupplierParty/ cac:Party/cac:PartyLegalEntity /cbc:RegistrationName
Tag2	VAT registration number of the seller	/Invoice/cac:AccountingSupplierParty /cac:Party/cac:PartyTaxScheme/ cbc:CompanyID
Tag3	Time stamp of the invoice (date and time)	Date Xpath /Invoice/cbc:IssueDate Time xpath /Invoice/cbc:IssueTime Issue date combination between issue date and issue time, expression sample from the invoice yyyy-MM-dd'T'HH:mm:ss'Z'
Tag4	Invoice total (with VAT)	/Invoice/cac:LegalMonetaryTotal /cbc:TaxInclusiveAmount
Tag5	VAT total	/Invoice/cac:TaxTotal/cbc:TaxAmount
Tag6	Hash of XML invoice	Invoice/ext:UBLExtensions /ext:UBLExtension/ext:ExtensionContent /sig:UBLDocumentSignatures /sac:SignatureInformation /ds:Signature/ds:SignedInfo /ds:Reference/ds:DigestValue
Tag7	ECDSA signature	/Invoice/ext:UBLExtensions/ext: UBLExtension/ext:ExtensionContent /sig:UBLDocumentSignatures /sac:SignatureInformation/ds:Signature /ds:SignatureValue
Tag8	ECDSA public key	/Invoice/ext:UBLExtensions/ext:UBLExtension /ext:ExtensionContent /sig:UBLDocumentSignatures /sac:SignatureInformation /ds:Signature/ds:KeyInfo /ds:X509Data/ds:X509Certificate
Tag9	For Simplified Tax Invoices and their associated notes, the ECDSA signature of the cryptographic stamp's public key by ZATCA's technical CA	/Invoice/UBLExtensions/ UBLExtension/ExtensionContent /UBLDocumentSignatures /SignatureInformation/Signature /KeyInfo/X509Data/X509Certificate

The hex representation:

T	L	V

01 17 41 68 6d 65 64 20 4d 6f 68 61 6d 65 64 20 41 4c 20 41 68 6d 61 64 79 02 0f 33 30 31 31 32 31 39 37 31 35 30 30 30 30 33 03 14 32 30 32 32 2d 30 33 2d 31 33 54 31 34 3a 34 30 3a 34 30 5a 04 07 31 31 30 38 2e 39 30 05 05 31 34 34 2e 39 06 2c 51 6e 56 45 65 78 57 34 6e 57 76 34 43 61 45 33 39 61 2f 36 36 4a 70 2f 4f 58 4f 2f 65 76 48 51 38 70 44 6c 47 37 77 65 71 2f 34 3d 07 60 4d 45 55 43 49 51 44 35 7a 78 79 58 4f 42 37 4e 76 57 66 36 32 72 56 45 5a 41 59 55 37 31 6a 70 79 39 48 45 45 6e 5a 30 71 39 4f 39 36 77 72 4c 36 51 49 67 51 4a 7a 43 47 48 62 77 36 59 42 48 4c 59 56 64 4f 31 77 6e 55 68 42 67 4b 6d 38 6a 4d 54 79 76 63 6b 39 4d 2b 72 50 39 78 59 59 3d 08 58 30 56 30 10 06 07 2a 86 48 ce 3d 02 01 06 05 2b 81 04 00 0a 03 42 00 04 61 83 0c a0 e6 85 60 08 4c 3b fb 2d 7a 8b 5f 67 26 af af aa 75 d5 24 a5 c2 c2 bd 6b 39 ac 2d 8e db d5 bf 85 2e 1a 8c 02 b8 41 d9 da 87 29 ba 31 a8 a3 5f be 42 83 78 f8 69 aa 3b a2 e6 17 27 d1 09 48 30 46 02 21 00 ee 61 d3 eb 28 3c e6 3b 50 19 6a 77 33 bb 4f 4f b2 64 db ec ec bd 51 c6 b3 76 d4 e5 9e d8 13 af 02 21 00 fa d1 e6 d0 6a 66 23 62 f7 5e 6e 71 63 35 fc 78 5f 87 68 a7 b2 ec 10 11 42 35 2b 0b 63 42 05 69

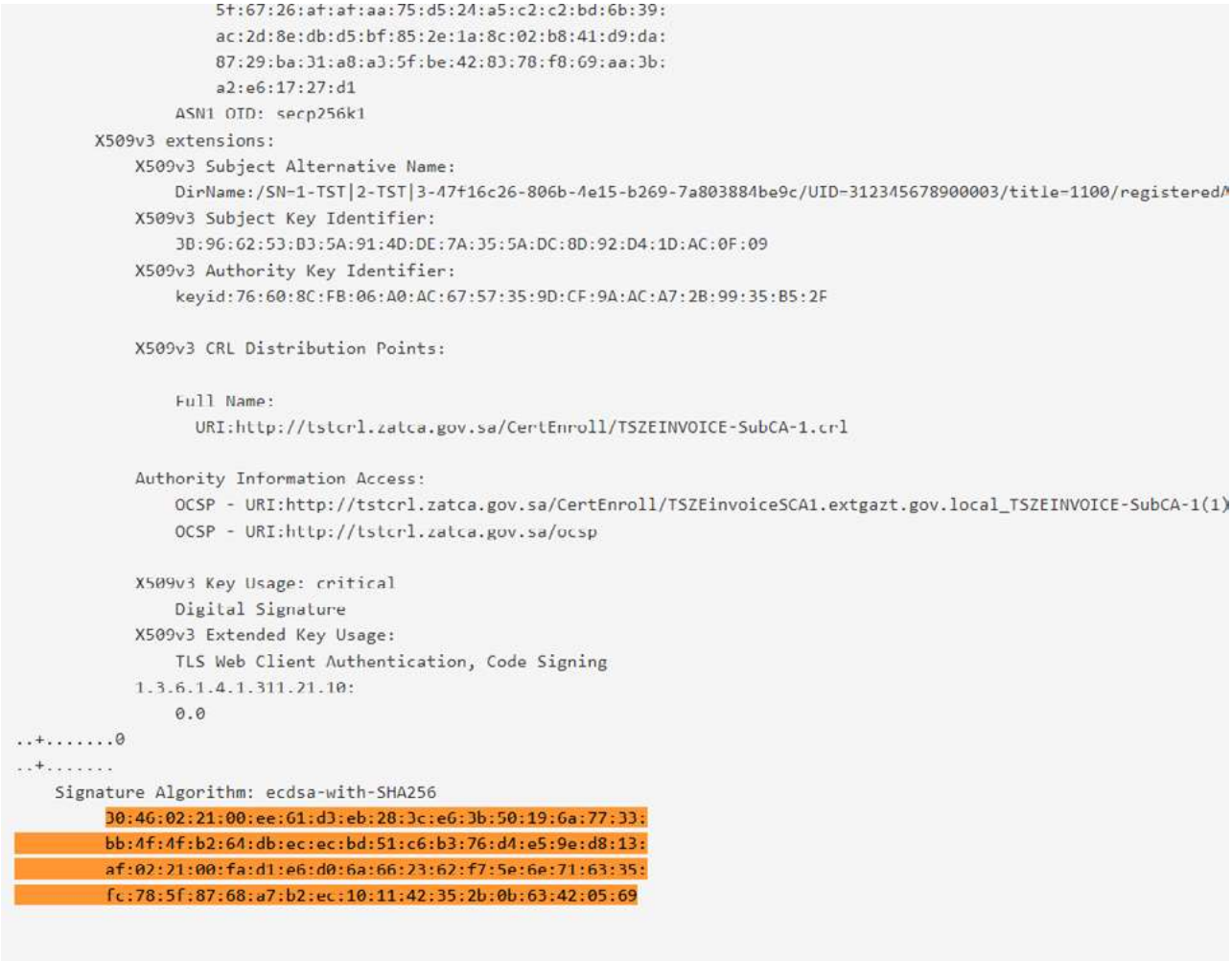
Field Definition for the QR Code

ARdBaG1lZCBNb2hhbWVkIEFMIEFobWFkeQIPMzAxMTIxOTcxNTAwMDAzAxQyMDIyLT-AzLTEzVDE0OjQwOjQwWgQHMTEwOC45MAUFMTQ0LjkGLFFuVkVIeFc0bId2NENhRT-M5YS82NkpwL09YTy9IdkhROHBEbEc3d2VxLzQ9B2BNRVVDSVFENXp4eVhPQj-dOdIdmNjJyVkVaQVIVNzFqcHk5SEVFblowcTIPOTZ3ckw2UUlnUUp6Q0dIYnc2WUJIT-FIWZE8xd25VaEJnS204ak1UeXZjazINK3JQOXhZWT0IWDBWMBAGByqGSM49AgEGB-SuBBAAKA0IABGGDDKDmhWAITDv7LXqLX2cmr6+qddUkpcLCvWs5rC2O29W/hS4ajAK4Qdnahym6MaijX75Cg3j4aao7ouYXJ9EJSDBGAiEA7mHT6yg85jtQGWp3M7tPT7Jk2+zsv-VHGs3bU5Z7YE68CIQD60ebQamYjYvdebnFjNfx4X4dop7LsEBFCNSsLY0IFaQ==

In order to get the value for Tag9 i.e. the Digital Signature of the Certificate please follow the steps below:

1. Get a hold of your device's PCSID (You get this once you have successfully onboarded the device)
2. Decode the PCSID. One available online tool is: https://certlogik.com/decoder/

You should get results that look like the diagram below:



3. Copy the value of "Signature Algorithm: ecdsa-with-SHA256"
4. Populate the value from step three as the value for Tag9 in the QR

6.3 Common mistakes in building the QR code

- Tag and Length are binary values, converted to Hex EXAMPLE: 21 should be represented as 15 in Hex, if the string is converted it becomes 32 31 (1 and 5)
- Value must also be converted to Hex before encoding to Base64
- There should be no padding or separators between the TLV sets in the resulting file
- Not using UTF8 Encoding for Arabic Text

6.4 Manual decoding a TLV QR Code

The QR code can be Extracted and Converted to Hex using publicly available tools

Step 1: Example Base64 Encode QR Code, extracted using QR Code reader (i.e. Mobile Phone):

ARVCb2JzIEJhc2VtZW50IFJIY29yZHMCDzEwMDAyNTkwNjcwMDAwMwMUMjAyMi0wN-C0yNVQxNTozMDowMFoECjIxMDAxMDAuOTkFCTMxNTAxNS4xNQ==

Step 2: Decode this to a hex representation, this can be done at the following site: cryptii

01 15 42 6f 62 73 20 42 61 73 65 6d 65 6e 74 20 52 65 63 6f 72 64 73 02 0f 31 30 30 30 32 35 39 30 36 37 30 30 30 30 33 03 14 32 30 32 32 2d 30 34 2d 32 35 54 31 35 3a 33 30 3a 30 30 5a 04 0a 32 31 30 30 31 30 30 2e 39 39 05 09 33 31 35 30 31 35 2e 31 35

Step 3: Hex Representation can be read by a TLV reader, i.e. : emvlab

Step 4: UTF8 Encoded values can be read using an online tool, i.e. : onlineutf8tools

Using a TLV Decoder to split the record shows the Hex Values, these can then be decoded using a hex to string decoder

Tag	Hex Value	Hex to string
Seller's name	426F627320426173656D656E7420 5265636F726473	Bobs Basement Records
VAT registration number of the seller	31303030323539303 6373030303033	100025906700003
Time stamp of the invoice (date and time)	323032322D30342D32355431353 A33303A30305A	2022-04-25T15:30:00Z
Invoice total (with VAT)	323130303130302E3939	2100100.99
VAT total	33313530313 52E3135	315015.15

6.5 Creation of QR code in JAVA - Javascript - nodeJS

This function takes in 2 args:

- tagNum: Tag Number
- tagValue: Value of Message

Convert the tagNum into byte array

Get the length of the value and convert it into byte array

Convert the value into byte array

Once we have 3 byte arrays, we concat them into 1 byte array representing our TLV Message

```
function getTLVForValue(tagNum, tagValue) {

    var tagBuf = Buffer.from([tagNum], 'utf8');


    tagValueLenBuf = Buffer.from([tagValue.length], 'utf8')


    var tagValueBuf = Buffer.from(tagValue, 'utf8');


    var bufsArray = [tagBuf, tagValueLenBuf, tagValueBuf]


    return Buffer.concat(bufsArray);
}
```

You do the previous steps for each of the Tags you want to add to the QR code. For example, here we have sellerName, VatReg, etc.

Concat those buffs into a single array representing the QR code (see 1, 2)

Afterwards, encode into Base64 (see 3)

```
// 1. Seller Name
var sellerNameBuf = getTLVForValue("1", "salah hospital");

// 2. VAT Registration
var vatRegistrationNameBuf = getTLVForValue("2", "312345678901 23");

// ....
```

```
var tagsBufsArray = [sellerNameBuf,
    vatRegistrationNameBuf,
(1) timeStampBuf, taxTotalNameBuf, vatTotalBuf,
    hashedXmlBuf, keyBuf, signatureBuf];

var qrCodeBuf = Buffer.concat(tagsBufsArray)(2)

var qrCodeB64 =  qrCodeBuf.toString('base64');
                                              (3)
```

Dart - Use the BytesBuilder class to add each segment of each TLV message i.e. 3 per message.
We repeat for each message we want to add to the QR Code

```
BytesBuilder bytesBuilder = BytesBuilder();

// 1. Seller Name
bytesBuilder.addByte(1);
List<int> sellerNameBytes = utf8.encode(sellerName);
bytesBuilder.addByte(sellerNameBytes.length);
bytesBuilder.add(sellerNameBytes);

// 2. VAT Registration
bytesBuilder.addByte(2);
List<int> vat_registrationBytes = utf8.encode(vat_registration);
bytesBuilder.addByte(vat_registrationBytes.length);
bytesBuilder.add(vat_registrationBytes);

// ....
```

```
Uint8List qrCodeAsBytes = bytesBuilder.toBytes();
final Base64Encoder b64Encoder = Base64Encoder();
return b64Encoder.convert(qrCodeAsBytes);
```


Once all messages are added to the builder, convert it into bytes (see 1) which gives you Uint8List (Darts way of byte []), then encode the list into Base64 using an instance of the Base64Encoder class (see 2).

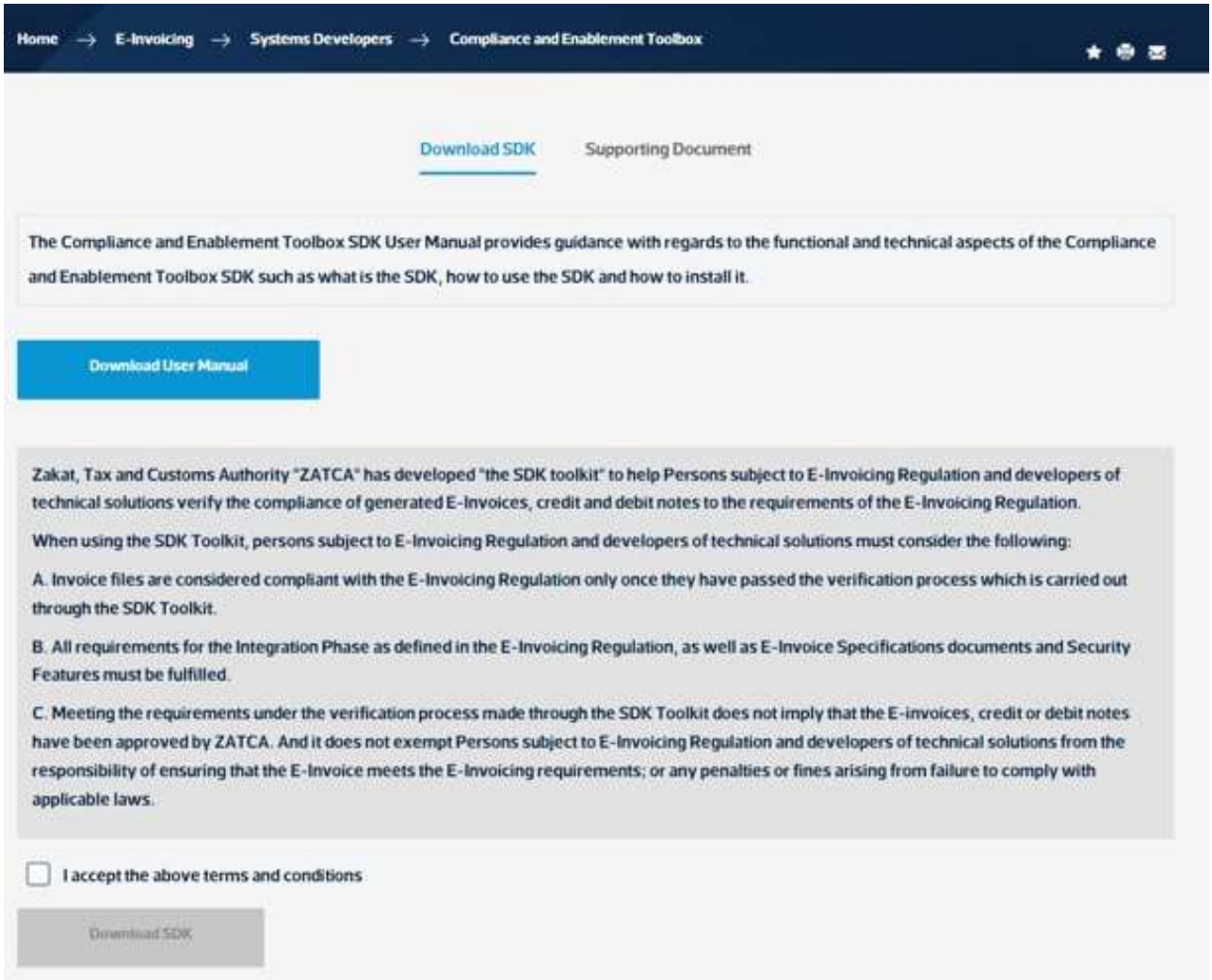
Representation of the QR code Data Examples:

Hyperlink to a Website	Data in Text Format	TLV Base64 string
https://zatca.gov.sa/en/pages/default.aspx	Seller's name Bobs Records VAT registration number 310122393500003 Time stamp 2022-04-25T15:30:00Z VAT total 1000.00 VAT total 150.00	AQxCb2JzI-FJIY29yZHMCDzMx-MDEyMjM5MzU-wMDAwMwMUM-jAyMi0wNC0yN-VQxNTozMDowM-FoEBzEwMDAuM-DAFBjE1MC4wMA==
Empty or unknown numbers	**Hyperlink to the invoice online**	
2000555663314	https://mcusercontent.com/a90cefeb037ed376188308d34/files/2ca406b2-8627-66d9-45a4-94d186a4f3a5/User_Manual_Software_Development_Kit_SDK_.01.pdf	

SDK validation

The QR code can be validated using SDK available on ZATCA's website (https://zatca.gov.sa/en/E-Invoicing/SystemsDevelopers/ComplianceEnablementToolbox/Pages/DownloadSDK.aspx).

Command line	Results
fatoorah - v	To display (Version)
fatoorah - h	Help window
fatoorah validateqr -qr	Validate QR code structure
fatoorah generate -f (Invoicename.xml) -q	Generate compliant QR code



7. Business FAQs

Questions (EN)	Answers (EN)
Is there a limit on the number of decimals for unit price of an invoice?	No, as specified in the table provided under paragraph 7.3 of XML Implementation Standard, there are no restrictions on the decimal places for Unit Price.
Can taxpayer sign and submit invoices from different devices?	Yes, as long as the submitting device submits the document with the CSID of the device that stamped the document. It is recommended that each EGS independently is able submit documents to ZATCA.
How long after VAT registration can Taxpayer submit or receive invoices?	Newly registered VAT Taxpayers are required to wait for 2 business days before they can onboard their EGS and start submitting documents to ZATCA. Similarly documents issued to newly registered VAT Taxpayers will be rejected unless and until 2 business days have passed.
Can Taxpayer use their existing device to submit invoices after joining a VAT Group?	Once a Taxpayer joins a VAT Group, their own VAT Registration is suspended and all CSIDs associated with their VAT Registration Number will be automatically revoked. To continue using the same EGS as part of the VAT Group, the Group Lead (Representative) needs to onboard the EGS under the Group VAT Registration Number and provide the OTP(s) to the Group member. The Member can then obtain a new CSID which is associated with the Group and can then start submitting documents to ZATCA as part of the VAT Group.
How will Taxpayer know when Clearance is disabled?	When Clearance is disabled, any Standard documents submitted to the Clearance API will return a 303 response indicating that Clearance is off and the document needs to be submitted using the Reporting API. When Clearance is disabled, the Reporting API will accept Standard documents. Note that Standard documents submitted using the Reporting API will still undergo validations associated with Standard documents however there will be no stamp or QR code returned by the Core E-invoicing Solution and the document will instead be Reported.

Questions (EN)	Answers (EN)
Can Taxpayer resubmit a rejected documented?	Rejected documents can be resubmitted as a new document however this new document should indicate the Previous Document Hash as the hash of the document that was generated immediately before the resubmitted document and not of the document that was generated before the original rejected document. For example, consider the scenario: Document 1 is submitted and accepted Document 2 is submitted and accepted Document 3 is submitted and accepted Document 4 (Resubmission of Document 2 after addressing errors) In this case the Previous Document Hash of the Resubmitted document should be the hash of Document 3 and not Document 1. Note that the ZATCA's platform will be storing and tracking the hash of rejected documents as well. Accordingly, in the example above Document 3 should have its Previous Invoice Hash as the hash of Document 2 even though it was rejected.
What is the difference between error and warning?	An error is a validation failure associated with a rejected document, while a warning is a validation failure associated with an accepted document. If a submitted document has even one error, it will be rejected entirely.
When re-submitting an invoice, should it be updated with the re-submitted time or left as it was in the original submission ?	The time can be updated based on the re-submitted document generation time; it does not need to capture the time of original submission. Re-submission is similar to submitting a new invoice, it does not need to have any link to the original document that was rejected.
For Standard Tax Invoices, what should be done if the clearance fails before issuing the invoice to the buyer?	In case of Standard Tax Invoices, if clearing fails (Response is 400 Error), then the taxpayer must submit another invoice for clearance after rectifying the errors. Please note that every document shall have its own hash and counter value. Rejected document's hash and counter value should not be changed or updated.

Questions (EN)	Answers (EN)
In case a Standard Tax Invoices (B2B) is submitted and not cleared / rejected from ZATCA, should the invoice be resubmitted with different invoice counter value?	Invoice counter value should not be re-used. Once a document is generated with an Invoice Counter Value (ICV), then that ICV cannot be mentioned on another document. A new document should have its own ICV.
For Simplified Tax Invoices, what should be done if the reporting fails after issuing the invoice to the buyer?	In case of Simplified Tax Invoices, if the reporting fails, then the taxpayer must correct the error from to prevent it from happening on subsequent documents. The error in rejected document can be rectified and a new document can be submitted for Reporting. As the invoice would have already been issued to customer, there is no need to issue another invoice (in most cases customer may not be available to share the invoice again and therefore such a requirement would be impractical). Transaction should be included in monthly or quarterly VAT return submission. Intention is not to stop B2C transactions for technical failures or errors in XML documents.
Can taxpayers report invoices not cleared from ZATCA in their VAT return reports?	Taxpayers must determine VAT liability based on their transactions. Technical failure or error on XML does not invalidate the transaction. VAT becomes due irrespective of whether a valid invoice was issued or not. There are different penalties relating to "Non payment of VAT" and "Not issuing valid Tax Invoices". These two are separate events. Therefore, sometimes there will be scenario where taxpayers may have to include even the rejected invoices in VAT returns as the VAT becomes due for the tax period.
What will happen if we sign some invoices with non-valid certificates?	Invoices signed with invalid certificates will be rejected. Taxpayer must complete the "Onboarding process" to receive valid Cryptographic Stamp Identifier (CSID) Certificate.
Do we have to wait for the previous invoice to be cleared before sending the next invoice?	There is no dependency on ZATCA's clearance for generating new invoice. ZATCA's stamp or QR Code string is not part of the Hash. Taxpayers can continue generating invoices without waiting for Clearance from ZATCA as Clearance does not change the hash of the document.

Questions (EN)	Answers (EN)
What is the Billing Reference ID found in Debit and Credit Notes?	Billing Reference ID is the link to Original Invoices for which Debit and Credit Note is generated. It is a mandatory field. As per Article 54 of KSA VAT Regulations, every Debit and Credit Note should refer to Original Invoice that it relates to. This is not a new requirement, it existed since 1 Jan 2018.
Is it possible to send to ZATCA multiple "Supply dates" and "Supply end dates" for one credit note?	UBL standard does not allow multiple "Supply dates". Taxpayers may select a supply date range which covers all original invoices to which the particular credit note relates.
Is there an option to report simplified invoices by bulk?	No - there is currently no option to do bulk reporting.
What is the FATOORA Portal?	The FATOORA Portal is provided by ZATCA to Taxpayers in order to allow them to onboard their E-invoicing Generation Solution Unit(s) . It is considered as the starting point for Taxpayers who want to onboard a EGS Unit and receive a Cryptographic Stamp Identifier (CSID) for the first-time, renew an existing CSID or revoke an existing CSID. Taxpayers can also use the Portal in order to view a summary list of all their onboarded EGS Unit(s) along with specific EGS Unit information that is provided as a part of the Certificate Signing Request (CSR).
How does the log-in to FATOORA Portal work?	The FATOORA Portal uses Single Sign On (SSO) based on the Taxpayers credentials for the Taxation Portal (ERAD).
Who is authorized to use the FATOORA Portal?	The FATOORA Portal can be accessed and all off its functionalities can be used by all Taxpayers who are registered on the Taxation Portal (ERAD) for VAT and who have a VAT Registration (TRN) status of "Active" or "Reactive". Taxpayers whose VAT registration status used to be "Active" or "Reactive" but changes to "Deregistered" or "Suspended" would be able to access the FATOORA Portal for a period of 90 days but can only view a list of their previously onboarded EGS Units and cannot use any other onboarding functionalities such as generating an OTP. Once the buffer period of 90 days is over, these Taxpayers will no longer be able to access the FATOORA Portal.

Questions (EN)	Answers (EN)
What is the difference between the Onboarding and Renewal process?	Both Onboarding and Renewal follow the same process and steps from a Taxpayer point of view. From a technical point of view, the request type submitted by the E-invoicing Generation Solution Unit (EGS Unit) is different and the renewal process includes revoking the existing Cryptographic Stamp Identifier (CSID) of the EGS Unit and issuing a new one.
What is the validity period of the OTPs?	OTPs are valid for 1 hour from the date of their generation.
Are there any differences between the onboarding process for VAT groups and individual Taxpayers?	Both individual Taxpayers and VAT groups follow a uniform process for onboarding in terms of the steps required to complete an onboarding, renewal or revocation. However, there are certain aspects that are specific to VAT groups, namely: In case of VAT groups, the Organization Unit Name that is a field in the Certificate Signing Request (CSR), should contain the 10-digit TIN number of the individual group member whose EGS Unit is being onboarded, in case the EGS Unit is to be used by a particular group member. The access rights for using the FATOORA Portal differ, whereby only the group lead would be able to meet the authorization criteria to access and use the functionalities provided by the Portal.
How will I know that my E-invoicing Generation Solution Unit (EGS Unit) has been successfully onboarded, renewed or revoked?	Email notifications will be sent to the Taxpayer once the status of an EGS Unit changes. In addition, Taxpayers can view the status of their EGS Unit(s) through the summary list of onboarded EGS Unit(s) that is a part of the FATOORA Portal. Once an EGS Unit has been onboarded, renewed or revoked, the respective fields of the list are updated to reflect the status accordingly.

Questions (EN)	Answers (EN)
What is a Certificate Signing Request (CSR)?	A certificate signing request (CSR) is one of the first steps towards getting the EGS's own Cryptographic Stamp Identifier (Certificate). It includes the following: Common Name: Name or Asset Tracking Number for the Solution Unit EGS Serial Number: Manufacturer or Solution Provider Name, Model or Version and Serial Number Organization Identifier: VAT or Group VAT Registration Number Organization Unit Name: Organization Unit Organization Name: Taxpayer Name Country Name Invoice Type: Functionality Map Location: Location of Branch or Device or Solution Unit Industry: Industry or location Please refer to Section 3.3.3 of the Taxpayer User Manual for more details on the CSR fields and the inputs required.
What are possible CSR failure situations?	Possible CSR failure situations including inserting the wrong algorithm, providing invalid values, missing information, inputting the wrong format or including expired/invalid OTP (note that the OTP is provided in the API header).
What is a Cryptographic Stamp Identifier (CSID)?	A CSID is technically a cryptographic certificate, which is a credential that allows for authenticated signing and encryption of communication. The certificate is also known as a public key certificate or an identity certificate. It is an electronic document used as proof of ownership of a public key. A CSID is used to uniquely identify an Invoice Generation Solution Unit in possession of a Taxpayer for the purpose of stamping (technically cryptographically signing) Simplified Invoices (B2C) and for accessing the Reporting and Clearance APIs.

Questions (EN)	Answers (EN)
What is a Compliance CSID?	A Compliance CSID is a CSID that is used by the EGS to call the compliance APIs and perform compliance checks, specifically the Compliance CSID is added as a request header when calling those APIs. Moreover, the Compliance CSID is generated by the e-invoicing platform itself rather than ZATCA CA since it is used only to ensure the compliance of the EGS with ZATCA specifications.
What is a Production CSID?	A Production CSID is a CSID that is used by the EGS to call the core e-invoicing production APIs such as reporting, clearance, etc. This CSID is generated by ZATCA CA and returned to the EGS which it can use to invoke the aforementioned APIs. Additionally, the Production CSID is added as a request header and also is used to authenticate and authorize the EGS.
What is the OTP generation process	The process of OTP generation is provided on the FATOORA portal for security and authentication reasons. Note that EGS providers can enhance their solutions to obtain the OTP automatically from the header after successful login. Currently, OTPs are valid for 1 hour providing sufficient time for Taxpayers with a large number of EGS units to be onboarded. OTP generation is managed by the FATOORA portal and must be taken from the portal itself, no need for any API. The OTP step is mandatory for the Onboarding / Renewal processes.

Questions (EN)	Answers (EN)
Is there a way to get an OTP through an API?	There is no API for OTP. You can only get OTPs thought the portal.
What is the maximum number of OTP we can request at once?	Each request can have till up to 100 OTPs.
What should we do with our UUID and ICV if an invoice gets rejected?	After an invoice gets rejected UUID and ICV should not be re-used. System should assign a new ICV when the document is submitted after fixing errors.
What are possible compliance checks failure situations?	Possible compliance checks failure situations include invalid documents/inputs or missing/invalid/expired Compliance CSID.
In case we sent an Invoice with errors and got the response from Reporting API «status: NOT_REPORTED», what should we do next?	In the case that an invoice is not accepted, the error should be checked, the invoice should be cancelled via a credit note and a new invoice generated. Once "Not Reported" the invoice is deemed invalid.
How does Single Sign On (SSO) work?	The SSO enables the Taxpayer to sign in only once to access different portals and websites. It allows the Taxpayer to access our platform of different components as if they were a single portal.
In case of reporting rejection from ZATCA, what shall the taxpayer do?	In case of reporting rejection from ZATCA, the taxpayer can fix the error and re-submit the Simplified Tax Invoice for reporting. The new invoice should include its own new unique hash, UUID, invoice counter value and timestamp. The date on invoice should remain as when transaction took place. Previous Invoice Hash will be based on immediately preceding document (not necessarily linked to the rejected invoice).
In case of any incidents, technical errors, or emergency matters in E-Invoicing solution which hinder the generation of Electronic Invoices or Electronic Notes, what shall the taxpayer do?	In case of any incidents, technical errors, or emergency matters in your E-Invoicing solution which hinder the generation of Electronic Invoices or Electronic Notes, taxpayers are required to notify ZATCA using the service https://zatca.gov.sa/en/E-Invoicing/FailureNotifications/Pages/VerifyTaxpayer.aspx Once the system failure is resolved, all the transactions that took place during the system downtime should be cleared/reported in compliant XML format.

Questions (EN)	Answers (EN)
In case of clearance rejection from ZATCA, what shall the taxpayer do?	In case of clearance rejection from ZATCA, the taxpayer can fix the error and re-submit the Standard Tax Invoice for clearance. The new invoice should include its own new unique hash, UUID, invoice counter value and timestamp. The date on invoice should be updated as non-cleared invoice is not considered as a valid Tax Invoice from ZATCA's perspective. Previous Invoice Hash will be based on immediately preceding document (not necessarily linked to the rejected invoice).
In case ZATCA disables the clearance, what shall the taxpayer do?	If clearance is disabled by ZATCA, the taxpayer will receive a 303 response and a message saying that the Clearance is disabled and that the Standard Tax Invoices or CN / DN should be submitted using the Reporting API.
Can a taxpayer submit the same invoice twice?	There are no restrictions on submitting same document more than once, system will not reject a document for the reason that it is already submitted earlier. However, taxpayers must investigate the reason and resolve it to avoid any potential duplication issues. ZATCA will consider the Invoice only once based on unique identifiers such as UUID and hash of the invoice.
Can anyone access sandbox and FATOORA portal?	No, Sandbox can be accessed by anyone, but FATOORA production system can be accessed only by taxpayers using Taxpayer portal credentials (ERAD credentials).
What is the sequence of sharing invoices with ZATCA and invoice counter?	There are no limitations on the sequence of uploading or submitting invoices to ZATCA, it can be uploaded in any order so long as they are generated in a sequence. Sequence will be validated only in backend (and not real-time validation at the time of receiving invoices). For example, if invoices 1, 2 and 3 were generated in sequence they can be submitted as 3, 1 and 2, while keeping in mind Clearance requirements for standard invoices. Any gaps identified in the sequence during backend validation will be subject to further investigation by ZATCA and may result in penalties if found to be non-compliant.

Questions (EN)	Answers (EN)
Can FATOORA portal and Sandbox be accessed from anywhere or only from KSA?	Yes, both FATOORA and Sandbox can be accessed from anywhere globally, not only from KSA
Should invoice be in English or Arabic?	All the data fields that are visible on human readable form of the invoice must be in Arabic. VAT Regulations require the Tax Invoices and Simplified Tax Invoices (along with corresponding credit notes or debit notes) to be in Arabic mandatorily. Invoice can be bilingual and include English as well.
If reported invoice had warnings, do I need to resubmit it?	No, you should not resubmit an invoice for which the response was "202 - Accepted with warnings" as it has been accepted by ZATCA. However, the warnings should be investigated and resolved at the earliest. Repeated non-compliance (that result in warnings) will be investigated by ZATCA and may be subject to applicable penalties.
Should CSID be generated for each invoice or only once during onboarding?	Only once during onboarding. Once a CSID is generated, it remains valid for multiple years.
In case of B2B invoice, if the VAT number is not available, should it be left empty?	Article 53(1) of VAT Regulations require the seller to issue a Standard Tax Invoice for B2B transactions. VAT Registration Number is required if the buyer is VAT registered. If the Buyer is not VAT Registered, then the field can be left blank. However, Additional Buyer ID is mandatory in case VAT Registration Number is left blank. Additional Buyer ID can be any ID in the following order of preference based on availability: (1) Tax Identification Number (TIN), (2) CR, (3) MOMRA License, (4) MLSD License, (5) 700 Number (6) SAGIA License, (7) National ID, (8) GCC ID, (9) Iqama Number or (10) Passport ID of the buyer (B2B).

8. Appendix

Glossary

ZATCA	ZAKAT, Tax and Customs Authority
QR Code	Quick Response Code
PKI	Public Key Infrastructure
EGS Unit	E-invoice Generation Solution Unit
API	Answers (EN) Application Programming Interface
CA	Certificate Authority
OTP	One time Password
CSR	Certificate Signing Request
CSID	Cryptographic Stamp Identifier
SSO	Single Sign On
TRN	Tax Registration Number
CN	Credit Note
DN	Debit Note

External Document

This guide has been prepared for educational and awareness purposes only, its content may be modified at any time. It is not considered in any way binding to ZATCA and is not considered in any way a legal consultation. It cannot be relied upon as a legal reference in and of itself, It is always necessary to refer to the applicable regulations in this regard. Every person subject to zakat, tax and customs laws must check his duties and obligations, he is solely responsible for compliance with these regulations. ZATCA shall not be responsible in any way for any damage or loss The taxpayer is exposed to that results from non-compliance with the applicable regulations.