For ESDC developers

E-SDCs are software applications or hardware devices whose main function is to:

- fiscalize invoices,
- enable issuing invoices with or without the internet (by safeguarding invoices in its memory)
- deliver fiscalized audit packages to the tax authority
- enable the audit of fiscalized invoices

From the technical point of view, E-SDC is a middleware component that connects an accredited invoicing system (POS) to a secure element and enables standardized communication with TaxCore.API.

This article offers useful initial information about technical requirements for accrediting E-SDC solutions.

NOTE:
Before reading the rest of this article, make sure you have read Getting Started With Accreditation
Also, before you start developing your solution, please read General Information for all vendors.

This article offers E-SDC vendors who are interested in accrediting an E-SDC solution the following insight:

- quick guide for E-SDC accreditation
- desired/needed professional experience for developing an E-SDC product
- how an E-SDC solution fits in with other EFD components?
- how to navigate the rest of the technical instructions in order to initialize development?

Quick step-by-step guide for E-SDC accreditation

1. Register as a vendor for the Sandbox environment
2. Receive a Developer Certificate and use it to access the Developer Portal
3. Use the Developer Portal to request additional certificates (smart cards) for testing purposes
4. Consult all the sections in these technical instructions to see understand all the requirements and how they should be implemented
5. Use the testing application SDC Analyzer on the Developer Portal to test your E-SDC's operation
6. Compile user documentation for your POS
7. Use the My Accreditations section on Developer Portal to apply for accrediting your POS

Which professional experience do I need for developing an E-SDC?

Be mindful that developing E-SDCs is much more challenging than developing POS solutions.

For that reason, we have compiled a list of desired professional experience in order to prepare developers for the challenges that an E-SDC development puts forwards:

- clear understanding of the multi-environment development concept - E-SDCs are developed in the development Sandbox environment and used by taxpayers in the official Production environment

- using APDU commands for communication with secure elements
- using Client Certificate (certificate store, pkcs12, pkcs11) authentication via HTTP protocol
- calling Web API
- using RSA and AES algorithms for encryption
- parsing X.509 certificates and using the obtained data

Integration with other EFD components

E-SDC is one of three components of any EFD setup.

NOTE:
For an overview of all EFD components and how they communicate with each other, see Electronic Fiscal Device.
For an overview of how EFD components interact to create a fiscal invoice, see Fiscal Invoice.

The graph below shows the communication pattern that an E-SDC establishes with other EFD components.

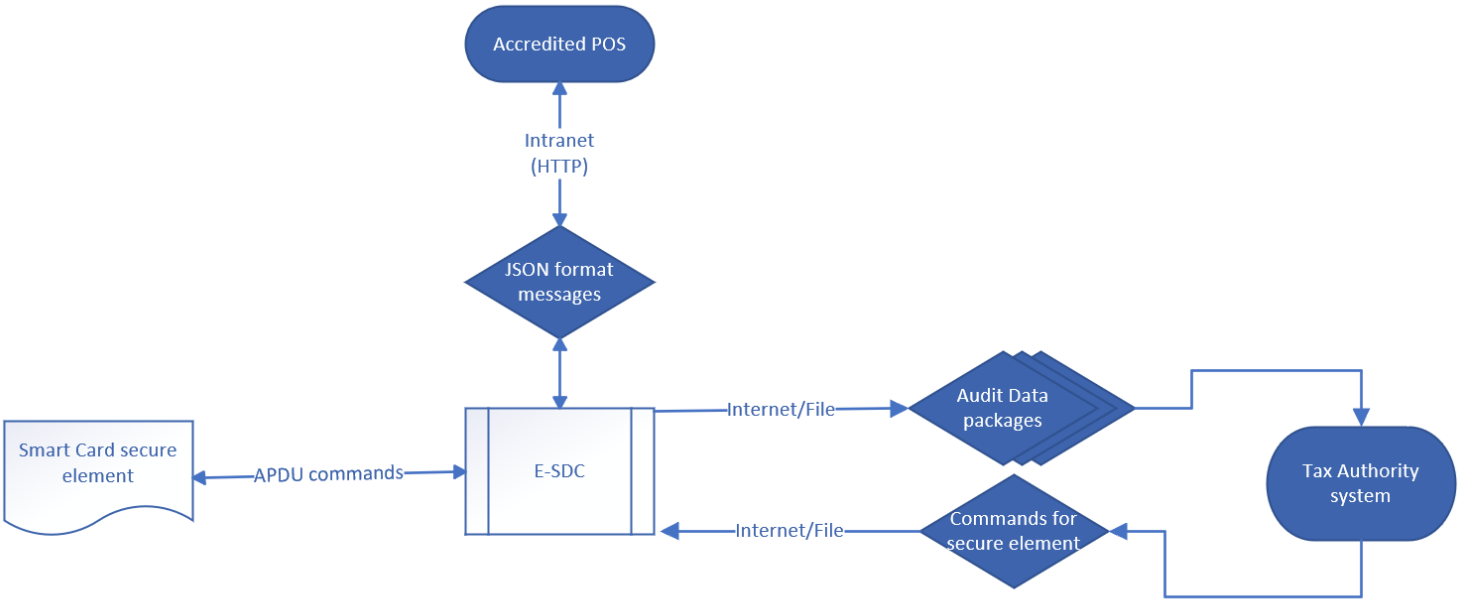


Image of the E-SDC communication pattern

What is the typical process flow of E-SDC operations?

All E-SDC solutions must follow the basic steps in the process of configuration and creating fiscal invoices (although some might have additional, manufacturer-specific, steps).

For a detailed step-by-step explanation of all standard E-SDC operations, see Processes.

NOTE:
Be mindful that E-SDC must be initialized every time when a new smart card is used. For more details, see E-SDC Initialization.

Communication with an accredited invoicing system (POS)

E-SDC communicates with an accredited invoicing system (POS) to receive invoice requests and return invoice responses - as part of fiscalizing each invoice.

Communication between a POS solution and E-SDC is established through JSON formatted messages using the POS to SDC Protocol.

Communication with a secure element

E-SDC communicates with a secure element (issued to taxpayers a smart card) to:

- perform fiscalization of data submitted as an invoice request from POS
- enable the audit of invoices fiscalized by that secure element
- enable transmission of other messages exchanged between the secure element and the tax authority (e.g. transferring new tax rates to the secure element)

Communication between an E-SDC and a secure element is established using APDU Commands.

For authentication purposes, an E-SDC also must be able to communicate with the PKI Applet installed on each smart card secure element. For more information, see PKI Applet.

Communication with the TaxCore.API (tax authority system)

E-SDC communicates with TaxCore.API to:

- submit fiscalized audit packages to the TaxCore database
- receive commands from TaxCore (which it then forwards to the secure element)

Communication between E-SDC and tax authority is established via TaxCore.API.

File-based communication

In case of prologued internet failure, E-SDC must allow file-based communication with the tax authority's system for the purpose of configuration and performing local audits.

All sections of the technical instructions

Technical instructions specific for E-SDC vendors/developers consist of the following sections:

1. High Level Requirements
   This section describes high-level requirements to consider when

2. Architecture
   This figure shows the components of a fiscalization system and their mutual relationship.

3. Data Structures
   This section contains descriptions of the main data structures used during the fiscalization and audit processes.

4. Processes
   This section describes processes performed by an E-SDC.

5. Protocols
   This section describes Application Programming Interfaces (API) and protocols exposed by an E-SDC or used by an E-SDC to communicate with the other components (TaxCore.API, Secure element Applet, PKI Applet or SD Card/USB Flash Drive) required to fulfill its primary role – to safeguard a transaction and to transfer the audit packages to the Tax Service's system.

6. Manuals
   E-SDC must have a user manual that explains the following topics in detail:

Related articles

- EFD Vendors General Information

High-Level Requirements

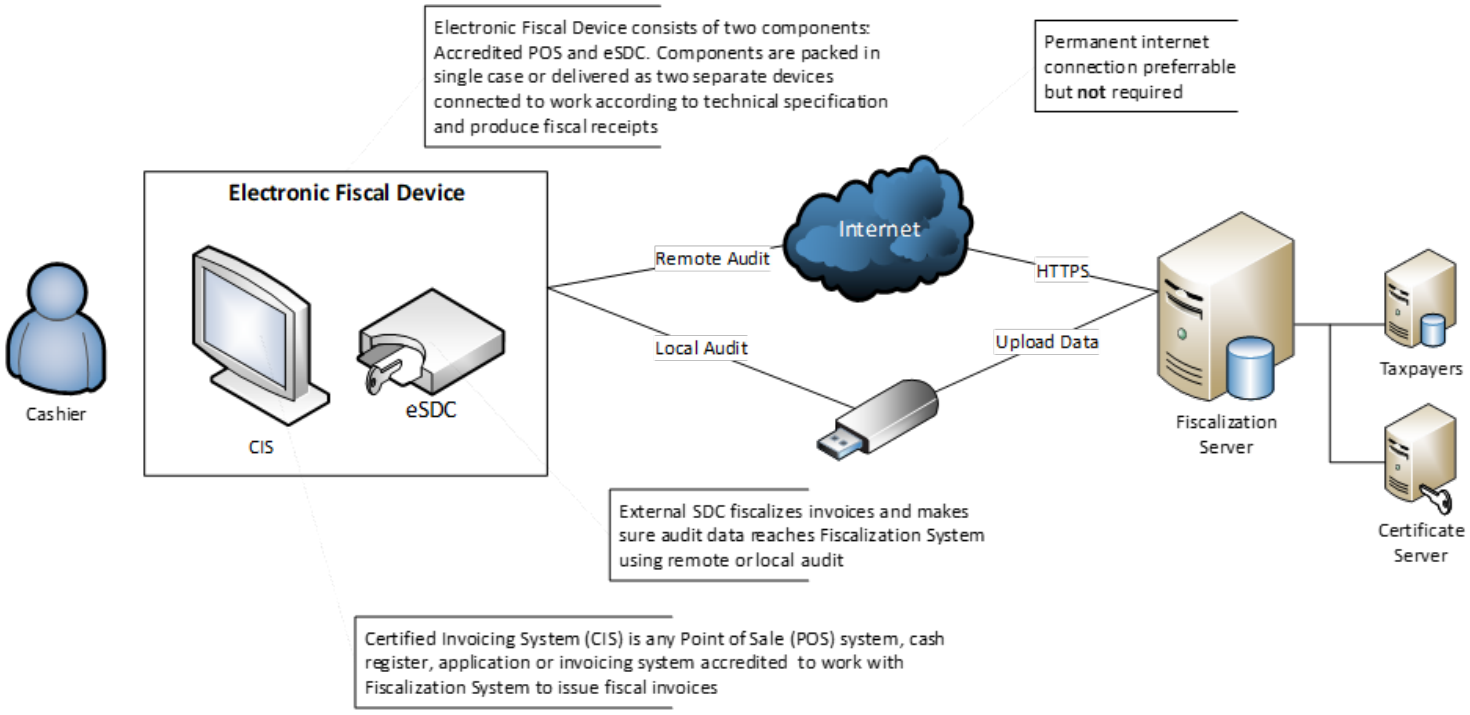
This section describes high-level requirements to consider when designing an E-SDC.

1. E-SDC is able to generate an invoice without an Internet connection.
2. E-SDC relies on a Secure Element (delivered as a smart card) to safeguard an invoice.
3. E-SDC must be designed so that it can work in multiple environments (development and production), depending on the certificate on the secure element it is using.
4. E-SDC calculates tax liabilities based on received Invoice items and defined tax rates.
5. Initial E-SDC configuration can be performed via a file (e.g. on USB disk) or the Internet connection.
6. E-SDC exposes a standardized protocol to a POS - JSON via HTTP.
7. E-SDC processes all commands received from the Tax Service's system in consecutive order. These commands can include time synchronization, an update of tax rates, etc.
8.

E-SDC encrypts audit data and stores it locally in an encrypted form.

9. E-SDC periodically sends stored audit data to a Tax Service and this process is called an audit.
10. E-SDC does not have to keep audit data submitted and successfully stored on the Tax Service's system.
11. E-SDC submits a proof of audit (PoA) generated by the Tax Service's system to the Secure Element as soon as the E-SDC receives it.
12. E-SDC does not store the Secure Element's PIN code except in the working memory. Once the E-SDC is restarted, the cashier is required to enter the PIN code again.
13. E-SDC stores Audit Packages in its non-volatile memory for the length of the period prescribed by the governing tax authority.
14. E-SDC keeps a log of all required error events with the exact local date and time

This figure shows the high-level architecture of Fiscalization System, involving an Accredited POS, E-SDC and Tax Service's system (TaxCore).



Fiscalization System high-level overview

Connectivity And Modes of Operation

Introduction

External Sales Data Controller (E-SDC) device exposes HTTP protocol for communication with an Accredited POS via a UTP cable, Wi-Fi and similar.

E-SDC uses a Secure Element to digitally sign invoices received from the Accredited POS and to produce audit data. The audit data is stored on the E-SDC's internal non-volatile memory, which enables a local and remote audit.

Multiple POSs can be connected to a single E-SDC. However, this should be avoided as multiple devices could send the data simultaneously, and since every smart card has its own limitations (resources, processing speed), that could slow down the overall process at sale point.

Use an online mode whenever possible

Taxpayers are encouraged to use an online mode whenever it's possible - V-SDC service will be widely available and accessible for a variety of Accredited POS devices and software solutions. But, in order to rollout, a fiscalization system needs to have the ability to close any possible gap in the fiscal discipline due to poor network coverage or the Internet unavailability.

Semi-Connected mode

E-SDC must be able to work in the Semi-offline mode.

In the semi-offline mode, the E-SDC can work both online (if the Internet is available) and offline (if the Internet is not available).

In the semi-offline mode, the Secure Element signs an invoice and the E-SDC device immediately tries to contact the Tax Service's system and perform a remote audit by invoking Submit Audit Package web method.

If the Tax Service's system is not accessible (for example, the internet connection is interrupted), the E-SDC automatically switches to the offline operation. In the offline operation, the Secure Element signs an invoice and the E-SDC device stores an audit package locally in a secure manner.

When the connection with the Tax Service system is re-established, the E-SDC will send the locally stored audit packages.

If the connection is interrupted for a prolonged period, the audit packages will be retrieved through a local audit process.

Architecture

Content

1. E SDC Implementation
   E-SDC can be implemented as a hardware device or a software service depending on the E-SDC manufacturer's decision and clients' infrastructure. E-SDC can also be implemented as an integral part of a POS

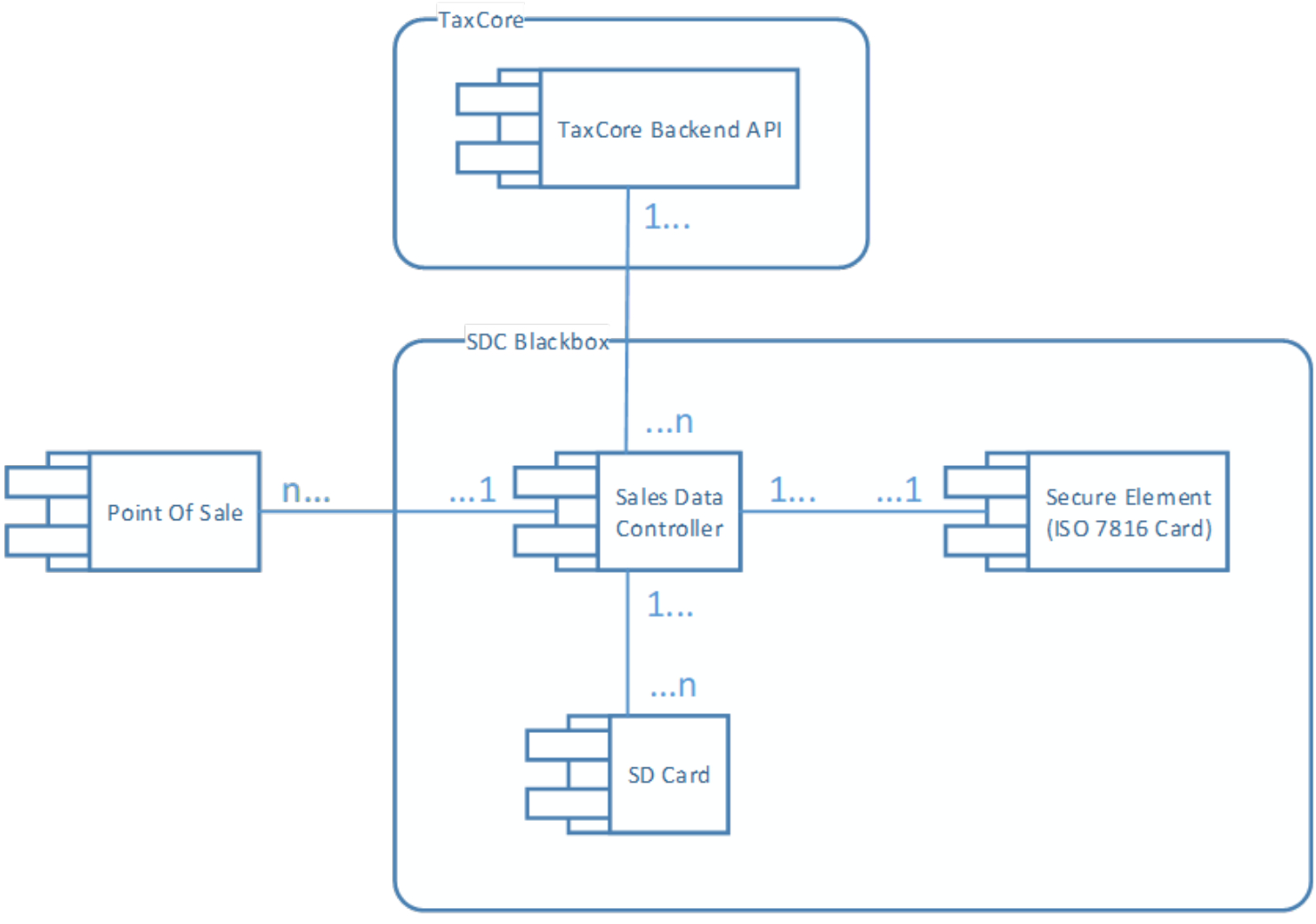
2. [Smart Cards]
   For standard operations, each E-SDC requires a Smart card issued by a Tax Service, which consists of two applets:

3. [E SDC States]
   The diagram below shows the basic E-SDC states and transitions.

4. [Authentication]
   Authentication against the Tax Service's system is performed using the taxpayer's digital certificate.

5. [TaxCore.API]
   TaxCore.API is a REST API exposed by a Tax Service's system to E-SDC devices. It provides services used by the E-SDCs to submit Audit Packages, to notify TaxCore if the online status has been changed and to receive configuration commands.

This figure shows the components of a fiscalization system and their mutual relationship.



ESDC Architecture – Image showing the components of a fiscalization system and their mutual relationship

E-SDC Implementation

E-SDC can be implemented as a hardware device or a software service depending on the E-SDC manufacturer's decision and clients' infrastructure. E-SDC can also be implemented as an integral part of a POS.

In any of those cases, the E-SDC component has to pass the same accreditation process and prove that E-SDC is implemented according to the instructions described in this document.

1. Standards
   E-SDC device complies with all current local regulations regarding safety usage, electromagnetic compatibility, temperature range and power supply.

2. Power Supply
   It is allowed to use both AC and DC voltage for power supply. In the case of the AC voltage, a device shall work with the frequency range 50-60 Hz. The power supply circuit used by the E-SDC shall be protected with an automatic circuit breaker, suitable for electronic devices (type I).

3. Ports
   Each E-SDC must provide ports or connectivity for the following purposes

Standards

E-SDC device complies with all current local regulations regarding safety usage, electromagnetic compatibility, temperature range and power supply.

Particularly, in terms of electromagnetic compatibility, a product shall comply with the following standards

- EN 55022:2010
- EN 55032:2012
- EN 55024:2010 A1:2015.

A device's operational temperature range shall be 0° - 70° C (Commercial range).

Power Supply

It is allowed to use both AC and DC voltage for power supply. In the case of the AC voltage, a device shall work

with the frequency range 50-60 Hz. The power supply circuit used by the E-SDC shall be protected with an automatic circuit breaker, suitable for electronic devices (type I).

In order to protect sensitive electronic components, the smart card and data stored on non-volatile memory, besides the mandatory basic protection, it is recommended that device is equipped with the additional fast overcurrent protection in the form of a fast fuse, e-fuse or similar device with the short time of operation.

When the DC voltage is used, protection against the reverse polarity shall be applied.

If the power supply voltages are higher than 75 Vdc or 50 Vac, a manufacturer shall obtain the appropriate certificate from a local authority, or represent a certificate valid in the country of use.

Ports

Each E-SDC must provide ports or connectivity for the following purposes

Secure Element

E-SDC has a smart card reader in compliance with ISO/IEC 7810 and ISO/IEC 7816 standard.

The supported Smart Card sizes are 1FF (credit card size) and 2FF (mini SIM card size).

SIM card	Standard reference	Length (mm)	Width (mm)	Thickness (mm)	Volume (mm^3)
Full-size (1FF)	ISO/IEC 7810:2003, ID-1	85.60	53.98	0.76	3511.72
Mini-SIM (2FF)	ISO/IEC 7810:2003, ID-000	25.00	15.00	0.76	285.00

Audit

If the E-SDC device uses a USB flash drive for a Local Audit, USB connectors "USB Type B female" must be used.

Applied USB communication protocol shall be “USB 2.0” or higher.

For situations when an SD Flash memory card is used for a Local audit, a device must have an easily accessible Micro SD card connector.

For a Remote Audit, these Instructions do not limit a manufacturer in choosing a communication port as long as the invoice signing is not interrupted.

POS Connectivity

POS must be able to connect to the E-SDC using at least one of the following ports:

Ethernet

Ethernet port in compliance with IEEE 802.3 standard, present on an E-SDC device. The minimum speed of the Ethernet port is at least 10 Mb/s.

Wireless

Wireless connection in compliance with IEEE 802.11 (Wi-Fi/Bluetooth) to a POS device and a local network.

Smart Cards

For standard operations, each E-SDC requires a Smart card issued by a Tax Service, which consists of two applets:

- Secure element Applet - used to apply a digital signature and maintain a set of fiscal counters in the offline mode
- PKI Applet - used to authenticate and establish a secure connection with the TaxCore.API web service

Both applets share the same PIN code. PIN is chosen during the process of requesting an Admin Secure element (smart card) - see Registration for Developer Portal- or while requesting an additional POS secure elements - see Requesting Additional Certificates.

PIN can't be changed after it is selected.

Each smart card is uniquely identified by a UID - Unique Identifier. Each digital certificate issued for E-SDCs has UID embedded in the certificate's subject field.

1. Secure Element Applet
   Secure element (SE) is a fiscal component implemented as a special software or a device designed to receive invoice data, perform signing and data processing and generate a response which is sent back to the caller for further actions.

2. PKI Applet
   PKI Applet contains a digital certificate and a private key used to authenticate E-SDC to TaxCore.API web services.

Secure Element Applet

Secure element (SE) is a fiscal component implemented as a special software or a device designed to receive invoice data, perform signing and data processing and generate a response which is sent back to the caller for further actions.

This response provides the authenticity of invoice data.

Secure Element is issued and controlled by the Tax Service. The main purpose of the SE is to sign invoices using a taxpayer's digital certificate, control audits and maintain a set of fiscal counters.

Each taxpayer is uniquely identified using digital certificates based on the Public Key Infrastructure (PKI).

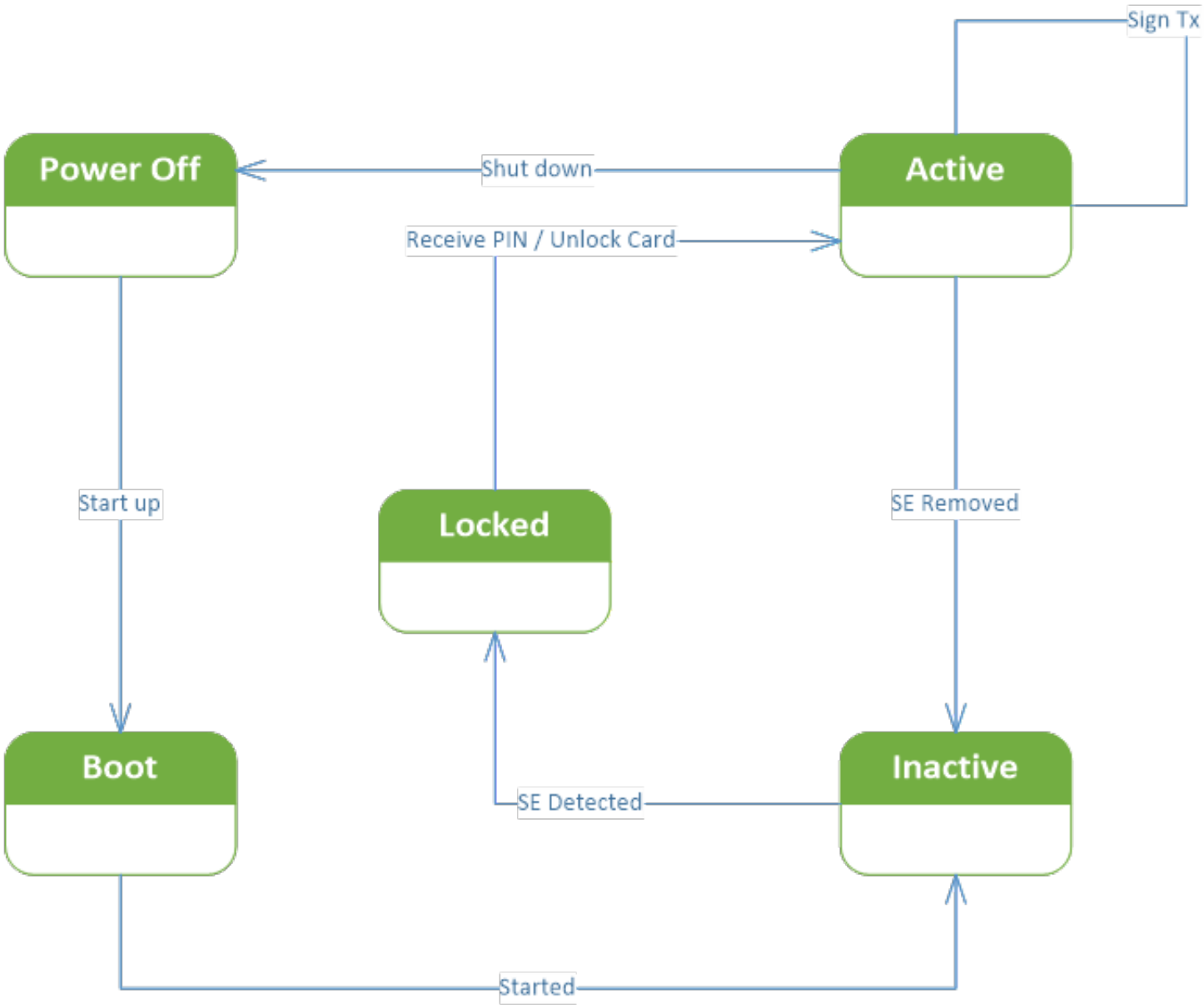
Secure element will stop issuing invoices if the maximum allowed amount for that particular fiscal device is exceeded – this facilitates the regular audit and forces taxpayers to report back to the Tax Service system. Likewise, the SE will continue to produce fiscal invoices once it receives proof from TaxCore.API that audit has been received and stored on the Tax Service's system.

PKI Applet

PKI Applet contains a digital certificate and a private key used to authenticate E-SDC to TaxCore.API web services.

E-SDC States

The diagram below shows the basic E-SDC states and transitions.



E-SDC States – Image showing the basic E-SDC states and transitions

Authentication

Authentication against the Tax Service's system is performed using the taxpayer's digital certificate.

Digital Certificates and PIN Codes

The Tax Service's system issues a Secure Element to a taxpayer as follows:

1. Taxpayer's digital certificate is stored in the Secure Element.
2. The Secure Element is stored on the smart card.
3. The PIN or password is generated and printed on the PIN mailer.
4. The Secure Element and PIN code are securely delivered to the taxpayer.

Digital Certificates for Testing Purpose

The Tax Service will issue the requested number of test digital certificates to each accredited supplier and each accredited taxpayer.

Authentication Token

E-SDC uses an authentication token when calling the TaxCore API web services. Authentication token is obtained from TaxCore API by calling the *RequestAuthenticationToken* web service and providing a Taxpayer's digital certificate.

TaxCore.API

TaxCore.API is a REST API exposed by a Tax Service's system to E-SDC devices. It provides services used by the E-SDCs to submit Audit Packages, to notify TaxCore if the online status has been changed and to receive configuration commands.

An E-SDC is authenticated by TaxCore.API using a client digital certificate and an authentication Token.

For More information please check SDC section of Taxcore.Api documentation.

Data Structures

This section contains descriptions of the main data structures used during the fiscalization and audit processes.

1. Calculate Taxes
   Taxes are calculated by an E-SDC after a POS has sent a valid request. The tax amount for particular items on an invoice is defined by the tax labels associated with an item.

2. UID
   UID is a Unique Identifier (8 alphanumeric characters) assigned to each Smart card and embedded in the Subject field of a digital certificate as the SERIALNUMBER parameter.

3. Amounts
   TaxCore works with amounts presented as decimal numbers with 4 decimal places. For the sake of simplicity, all binary-based protocols, E-SDC to Secure element - APDU commands, use integers representing the amount multiplied by 10,000.

4. Date and Time
   E-SDC has access to the current time. Real-Time Clock or a similar component must be installed and used to maintain the correct time while the power is off.

5. Tax Rates
   E-SDC receives tax groups via *Set Tax Rates Command* on the E-SDC Initialization process, or as an information if taxes have been changed.

6. Tax Amounts
   It is essential to note, that **POS never uses other taxes except the ones received from an SDC**. POS displays the total prices and only the tax values received from an SDC device, in the format described in the previous section.

7. Fiscal Invoices
   For more information please visit Fiscal Invoice

8. Anatomy of a Fiscal Receipt
   A receipt records the sale of goods or the provision of a service. The table below explains the structure of a fiscal receipt. **All elements are mandatory** unless specified otherwise in the detailed explanation below. POS is free to print any content (coupons, logos, etc.) before the beginning and after the ending mark of the fiscal invoice.

9. Audits
   Audit data represent a machine-readable formatted fiscal invoice signed by a taxpayer's private key followed by journal data. Journal data is a textual representation of a fiscal invoice generated by E-SDC.

10. Commands
    Commands are a means of communication between the Tax Service's system and E-SDCs. Commands are stacked in the queue list on the server for a specific E-SDC and submitted to the E-SDC as a part of the response once it reports to the Tax Service's system using a remote or a Local audit.

11. Commands Results
    Commands Results is a confirmation to TaxCore.API that a certain command is executed.

Calculate Taxes

Taxes are calculated by an E-SDC after a POS has sent a valid request. The tax amount for particular items on an invoice is defined by the tax labels associated with an item.

Process of a tax calculation depends on:

- Invoice and Transaction Type
- the tax rates for each label associated with an item on an invoice
- the Type value of tax category to which the label belongs

A POS sends an invoice fiscalization request with the line items. Items are sent with the total amounts (taxes included) and zero or more tax labels associated with them, which participated in the total price calculation.

In order to calculate a tax, the following algorithm is implemented:

1. Make an array of distinct tax labels associated with the items in the POS request (e.g. A, B, C, F, ...).
2. Calculate the tax amount for each individual label in the array:
    - Iterate through all items in the POS request
    - For each item, calculate tax amounts. One item has one or more tax labels, and each label represents a tax amount. Each tax amount is a part of an item's total price. These tax amounts are calculated as follows:
        - ♠ If an item has a label from the **amount-on-quantity** category applied, subtract the tax rate amount for that label, multiplied with quantity, from the item total price. The resulting amount (the remainder), is used in all further calculation steps instead of item total amount.
        - ♠ If none of the labels' tax category type is ***tax-on-total*** (category 1):
            - ♠ Tax amount for one label is:

$$\frac{item\ total\ amount\ *\ label\ rate}{(100 + \Sigma(all\ tax - on - net\ rates\ on\ item))}$$

Example 1: An item has a total price of 10$ and applied labels: A(5%) and B(6%).

$$A = \frac{10\$ * 5}{(100 + \Sigma(5 + 6))}\ \ B = \frac{10\$ * 6}{(100 + \Sigma(5 + 6))}$$

Tax amount for label A=0.4505$ and for label B=0.5405$.

- ♠ If any of the labels' tax category is ***tax-on-total*** (category 1):
    - ♠ Tax amount for every label whose category type is ***tax-on-total*** (category 1) is:

$$\frac{item\ total\ amount}{(1\ +\ \Sigma(all\ tax - on - total\ rates)/100)} * \frac{label\ rate}{100}$$

- ♠ Tax amount for every other label from category 0 is:

$$\frac{item\ total\ amount}{(1\ +\ \Sigma(all\ tax-on-total\ rates)/100)} * \frac{label\ rate}{(100+\Sigma(all\ tax-on-net\ rates\ on\ item))}$$

Example 2: Item has a total price 10$ and applied labels: A(5% tax-on-net), B(6% tax-on-net), C(3% tax-on-total) and F(4% tax-on-total).

$$A = \frac{10\$}{(1\ +\ \Sigma(3+4)/100)} * \frac{5}{(100+\Sigma(5+6))}$$

$$B = \frac{10\$}{(1\ +\ \Sigma(3+4)/100)} * \frac{6}{(100+\Sigma(5+6))}$$

$$C = \frac{10\$}{(1\ +\ \Sigma(3+4)/100)} * \frac{3}{100}$$

$$F = \frac{10\$}{(1\ +\ \Sigma(3+4)/100)} * \frac{4}{100}$$

Tax amount for label A=0.4210$, for label B=0.5052$, for label C=0.2804$ and for label F=0.3738$.

♠

Add calculated labels' tax amounts to the label's total amount sum.

Example 3: the request contains two items from Example 1 and Example 2, the total sum for labels are: A=0.8715$, B=1.0457$, C=0.2804$, F=0.3738$.

```
- Add fixed tax amounts, multiplied with quantity, to the respective labels' to

  Example 4: An item has quantity 2 with total price 10$ and applied labels: A(

  Example 5: An item has quantity 2 with total price 10$ and applied labels: A(
```

3.

After all of the items have been processed, calculate the tax amount for all tax categories found in the request. One tax category can consist of one or more tax labels (e.g. A, B…). The tax amount for a tax category is a sum of all label tax amounts related to the category.

Example 6: The request contains two items from Example 1 and Example 2. Labels A and B are VAT category, C is STT category and F is ET category. Total VAT=1.9172$, STT=0.2804$ and ET=0.3738$.

Once the Tax calculation is completed, assign GroupId of the active tax rate group to the field *TaxGroupRevision* of *InvoiceFiscalizationResult*.

Rounding

An SDC (both E-SDC and V-SDC) rounds all amounts to 4 decimal places, using the half-round up method.

The rules for this are simple:

- Decide which is the last digit to keep (in this case, the fourth decimal place)
- Leave it the same if the next digit is less than 5
- But increase it by 1 if the next digit is 5 or more

Examples:

3.44445555666 → 3.4445

3.4440012345 → 3.4440

3.44466012345 → 3.4447

3.444116012345 → 3.4441

Any amount shall be rounded to 2 (two) decimal places, using the half-round up method, **only** on the textual representation of an invoice.

NOTE:
In SDC's internal memory, the amount will always be stored with 4 decimal places (if the amount has 4 or more decimals), regardless of invoice's textual representation with 2 decimal points.

UID

UID is a Unique Identifier (8 alphanumeric characters) assigned to each Smart card and embedded in the Subject field of a digital certificate as the SERIALNUMBER parameter.

UID can be obtained from the Secure Element Applet using the *Export Certificate* APDU command.

Amounts

TaxCore works with amounts presented as decimal numbers with 4 decimal places. For the sake of simplicity, all binary-based protocols, E-SDC to Secure element - APDU commands, use integers representing the amount multiplied by 10,000.

For example, the amount of 123,4567 will be converted into 1234567.

JSON based protocols use the standard decimal format.

Date and Time

E-SDC has access to the current time. Real-Time Clock or a similar component must be installed and used to maintain the correct time while the power is off.

E-SDC synchronizes the time with the NTP Server, configured by the *Configure Time Server URL Command*.

UTC is the default time used by E-SDC for all purposes, except:

- Date and time sent by a POS to an E-SDC is the local time.
- Date and time printed on a journal (textual representation of an invoice) generated by an E-SDC is local time.
- Date and time sent by an E-SDC to a POS is the local time

JSON-based protocols use date and time according to ISO 8601 where applicable. For example:

Date: **2019-12-11**

Date and time in UTC:

2019-12-11T10:06:33+00:00

2019-12-11T10:06:33Z

20191211T100633Z

Week:

2019-W50

Date with week number:

2019-W50-3

Date without year:

--12-11[1]

Ordinal date:

2019-345

Time offsets are represented in format "**LocalTime+/-Zone**" as in case of **2019-12-11T10:06:33+00:00**

NOTE:
Date and Time display format on all TaxCore portals is: DateTimeDisplayFormat - "dd/MM/yyyy HH:mm:ss" (e.g. 15/10/2018 14:28:24).

All binary-based protocols (E-SDC to Secure element - APDU commands use Unix Timestamp format formatted as a 64bit unsigned integer Big Endian (for example: 1495018011910 is 2017-05-17T10:46:51.910Z).

In case of a power outage, the Real-Time Clock must be able to operate without interruption for up to 6 months.

Tax Rates

Tax rate: a sales tax expressed as a percentage on the sale of goods and services (for a proportional tax), or fixed tax amount, as imposed by the government. One tax rate is uniquely identified by the tax label – a string (usually only one letter) which represents one tax rate. One label is related to one category, and it will never reappear within another category.

Tax category: one or more tax rates are grouped into a tax category – a group of different rates for the same tax type (e.g. VAT, Consumption, ECAL, etc.). If a tax rate (all rates under a category) is applied on the net price, then field *Type* for a category shall be set to 0 (proportional tax). If a tax rate (all rates under a category) is applied on the total amount (with other taxes included), the field *Type* for a category shall be set to 1 (proportional tax-on-total). If a tax (all rates under a category) is a fixed tax amount, the field *Type* for a category shall be set to 2 (amount-per-quantity).

Tax group: one or more tax categories are grouped into a tax group – a set of all taxes imposed by the government applied on any points of sale within a period of time. A tax group consists of an ID, which represents tax revision and a date/time which defines the moment when the new taxes shall take effect.

E-SDC receives tax groups via *Set Tax Rates Command* on the E-SDC Initialization process, or as an information if taxes have been changed.

Model and Example

TaxRateGroup

Field	Type	Description
ValidFrom	Date time	Date when a tax rates group shall enter into force
GroupId	32 bit integer	Revision number for all taxes under a tax rates group
TaxCategories	Array of TaxCategory	All tax categories under one tax rates group

TaxCategory

Field	Type	Description
CategoryId	32 bit integer	Unique ID of a category (system-wise)
Name	Unicode String	Name of a tax (tax category)
Type	32 bit integer	One of the following tax category types: 0 (tax-on-net) - all tax rates from this category are proportional, and shall be applied on the net price; 1 (tax-on-total) - all tax rates from this category are proportional, and shall be applied on the total amount; 2 (amount-per-quantity) - all tax rates from this category are fixed tax amounts, which shall be multiplied with item quantity.
OrderId	32 bit integer	Order number for a tax category. It uniquely identifies the tax. It is related ¬to the category Name meaning even if the name changes for the tax category, it's OrderId will remain the same, pointing to the same tax. It is crucial for Sign Invoice APDU command
TaxRates	Array of TaxRate	All tax rates for a tax (category)

TaxRate

Field	Type	Description
RateId	32 bit integer	Unique ID of a tax rate (system-wise)
Label	Unicode String	Label for a tax rate, unique within a tax group, always belongs to one tax category
Rate	Decimal	Rate percentage for a proportional tax, or tax amount for a fixed tax.

##Example

```
"TaxRateGroup": {
       "ValidFrom": "2017-07-02T00:00:00",
       "GroupId": "1",
       "TaxCategories": [{
              "CategoryId": 1002,
              "Name": "VAT",
              "Type": 0,
              "TaxRates": [{
                     "RateId": 1002,
                     "Rate": 6.0,
                     "Label": "A"
              },
              {
                     "RateId": 1003,
                     "Rate": 10.0,
```

```
                "Label": "B"
            },
            {
                "RateId": 1004,
                "Rate": 20.0,
                "Label": "C"
            }],
            "OrderId": 1,
        },
        {
            "CategoryId": 1003,
            "Name": "STT",
            "Type": 1,
            "TaxRates": [{
                "RateId": 1005,
                "Rate": 25.0,
                "Label": "D"
            }],
            "OrderId": 2,
        },
        {
            "CategoryId": 1004,
            "Name": "ECAL",
            "Type": 2,
            "TaxRates": [{
                "RateId": 1006,
                "Rate": 10.0,
                "Label": "E"
            }],
            "OrderId": 3,
        }]
}
```


Tax Amounts

It is essential to note, that **POS never uses other taxes except the ones received from an SDC**. POS displays the total prices and only the tax values received from an SDC device, in the format described in the previous section.

A tax label can fall into one of the three tax types: Tax, Tax on Total and Amount per Quantity.

Tax amount for a tax label is calculated per the following formulas:

Tax Type	Formula	Explanation
Tax	**Tax Amount** = ∑(Base price x (**Rate/100**))	For each item with this label.
Tax on Total	**Tax Amount** = ∑(Total price x (**Rate/100**))	For each item with this label (Total price here is item base price with all other regular taxes included).
Tax on Quantity	**Tax Amount** = ∑(Fixed amount x quantity)	For each item with this label, item quantity is multiplied with fixed amount as defined by tax Service. If other tax labels are defined on the item, those taxes are

		calculated on the remainder.

Fiscal Invoices

For more information please visit Fiscal Invoice

Content

Unique Identification of a Fiscal Invoice

A fiscal invoice is uniquely identified by Invoice number - the combination of the invoice ordinal number and the Secure Element identification number (UID). Invoice number is defined in the following format:

UID1-UID2-Ordinal_Number

Where:

- **UID1** and **UID2** are identical for the invoice issued by E-SDC and
- **Ordinal_Number** is a number generated by the Secure Element, after each invoice signing, representing a total count of invoices signed by that Secure Element.

Key Elements

A fiscal invoice must contain the following parts (it may also contain additional data if it is required by a specific

industry):

Invoice Request

Invoice Request is created by an Accredited POS and it contains the usual invoice information like items, tax labels and invoice number. The invoice request is submitted by the Accredited POS using the standard, publicly available protocol for communication to E-SDC and the preferred technology of the POS system.

Invoice Response

Invoice Response is generated by E-SDC after data validation. It is an integral part of any fiscal invoice. Without this information, an invoice could not be considered a legal fiscal invoice.

Signature

A digital signature applied to the content of an electronic invoice by the Secure Element.

Internal Data

Internal data contains encrypted fiscal data. The content of the internal data is readable by the Tax Service system only.

Verification URL

URL of verification service is used to verify the authenticity of the particular fiscal invoice for customer convenience. It shall be represented as a QR code on a printed receipt or as a hyperlink in an electronic document (e.g. an email).

JSON Representation of the Invoice

As previously described, each Invoice consists of two separate data structures:

InvoiceFiscalizationRequest – this object is created by POS and submitted to E-SDC

InvoiceFiscalizationResponse – this object is created by E-SDC and returned to POS

Model

This Model is designed and based on OpenAPI-Specification V2 (https://github.com/OAI/OpenAPI-Specification).

```
Invoice {
Request (InvoiceFiscalizationRequest),
Result (InvoiceFiscalizationResult)
}
```

Anatomy of a Fiscal Receipt

A receipt records the sale of goods or the provision of a service. The table below explains the structure of a fiscal receipt. **All elements are mandatory** unless specified otherwise in the detailed explanation below. POS is free to print any content (coupons, logos, etc.) before the beginning and after the ending mark of the fiscal invoice.

Elements of a fiscal receipt

Title line – marks the beginning of the fiscal part of a receipt

```
============ FISCAL INVOICE ============
```

Header data is provided by E-SDC during fiscalization of the invoice and returned to POS as part of the InvoiceFiscalizationResult object (explained in section Sign Invoice. Values are extracted from the subject field of the digital certificate stored in the Secure Element Applet.

```
TIN:                            502579006
Company:                           Golf V
Store:                          Sun Store
Address:                    7 Someplace
District:                            Suva
```

Cashier identification is mandatory for every jurisdiction where local regulations mandate POS to send particular data such as Employee ID or some other information that uniquely identifies the POS cashier.

```
Cashier TIN:                    1234567890
```

Buyer TIN is mandatory **only** for B2B transactions and in that case, it must be printed on the receipt. When creating a B2B invoice for exporting goods/services, a country code prefix must be added to the Buyer TIN.

Buyer Cost Center is optional and reserved for further use, it must be present **only** for B2B transactions. **POS (Invoice) Number** and **POS (Invoice) time** are optional fields.

```
Buyer TIN:                     5123456789
Buyer Cost Centre:                    123
POS number:                   POS2017/998
POS time:            15/6/2017 8:56:23AM
```

Reference (Document) Number is always mandatory for Refund or Copy transactions. **Ref No** is printed on the receipt, containing the **SDC Invoice No** of the document which is being referenced in the format of *RequestedBy-SignedBy-OrdinalNumber*.

Reference combinations are as follows:

		I must/can reference...							
		Normal Sale	Normal Refund	Proforma Sale	Proforma Refund	Copy Sale	Copy Refund	Training Sale	Training Refund
When I issue...	Normal Sale			optional					
	Normal Refund	mandatory							
	Proforma Sale			optional					
	Proforma Refund			mandatory					
	Copy Sale	mandatory		mandatory					
	Copy Refund		mandatory		mandatory				
	Training Sale								
	Training Refund							mandatory	

Anatomy of a Fiscal Receipt - Reference combinations image

Optional reference may be mandatory if requested by a Tax Authority (obligation may apply to certain business activity).

All Reference Numbers are created for connected transactions, and if a **Ref No** is sent to an SDC, it must be displayed on the receipt journal.

If a Copy or Refund is issued for the transaction that was recorded before the introduction of fiscalization, POS should send XXXXXXXX-XXXXXXXX-1 as the value of **Ref No** field.

```
Ref no:          P22VC8VR-JTJC5V65-114906
```

Invoice and **transaction type** description. Normal Sale and Normal Refund are the most common types. Other types of transactions and invoices are defined in section Fiscal Invoices.

```
--------------NORMAL SALE---------------
```

List of items with **gross price**, **tax labels**, **unit price** and **quantity**. Tax Labels and their validity dates are published by the Tax Service and they are mandatory for each item, even when the price is 0.00.

When applying discounts, **Unit price** of the line item displayed on the journal (Price column) shows the discounted price, after **all** discounts have been applied.

```
Items
========================================
Name        Price          Qty.      Total
Sport-100 Helmet, Blue (E)
            34.99           10     349.90
Mountain Bike Socks, M (A)
             9.03            4      36.12
HL Road Frame - Red, 58 (F, A)
          1431.50            2    2863.00
Plastic bag (P)
             0.10            5       0.50
```

Total Purchase, **Tax items** and **Total Tax** are calculated by E-SDC during fiscalization of the invoice and are returned to POS as a part of the response.
Payment Method: Cash, Card, Check, Wire Transfer, Voucher, Mobile Money, or Other. Taxpayer's tax liability is based on these tax amounts, calculated by E-SDC. The calculation is explained in the section Tax Amounts. In case of Refund receipts, **Total Refunded** should be printed instead of **Total Purchase**.

```
--------------------------------------
Total Purchase:                 3249.52
Payment Method:                    Cash
======================================
Label      Name       Rate          Tax
E           STT      6.00%        19.81
A           VAT      9.00%       219.51
F          ECAL     10.00%       240.59
P            PB      0.10%         0.50
--------------------------------------
Total Tax:                       480.41
```

Fiscal metadata added to the invoice through fiscalization. **SDC Invoice No** - Combination of Requested By (7AF4D923), Signed By (E3B30A31) and Ordinal Invoice Number (234) is a system-wide unique identification of fiscal invoice. It may be used instead of the current receipt/invoice number generated by POS. **SDC Time** is the official date and time relevant to the tax calculation and reporting. **Invoice Counter** is generated by V-SDC or E-SDC and explained in section Invoice Response, field IC.

```
======================================
SDC Time:            2017-06-15 08:56:25
SDC Invoice No:    7AF4D923-E3B30A31-234
Invoice Counter:               230/234NS
======================================
```

QR Code contains Invoice verification URL. QR Code also contains Internal data and digital signature used for the invoice verification. Invoice is verifiable by the customer immediately after fiscalization. In case an invoice/receipt is delivered as an electronic document (email), QR Code shall be substituted with Invoice verification URL in (clickable) hyperlink format.

NOTE:
This is just a sample QR code image, not an actual URL.

Anatomy of a Fiscal Receipt – Image of a QR Code

Title line – marks the end of the fiscal part of a receipt

```
======= END OF FISCAL INVOICE ========
```

Custom message returned from E-SDC

```
 This is a custom message.
```

Example of a fiscal receipt

```
============ FISCAL INVOICE ============
TIN:                                  123
Company:                             Test
Store:                               Test
Address:                            nesto
District:                              Ba
Cashier TIN:                    987654321
---------------NORMAL SALE---------------
                      Items
```

==========

Name	Price	Qty.	Total
pizza (P)			
	3.30	2	6.60
juice (P)			
	2.45	3	7.35
chocolate (E)			
	1.50	4	6.00

Total Purchase: 19.95

Payment Method: Card

==========

Label	Name	Rate	Tax
P	PBL	0.20$	1.00
E	STT	6.00%	0.34

Total Tax: 1.34

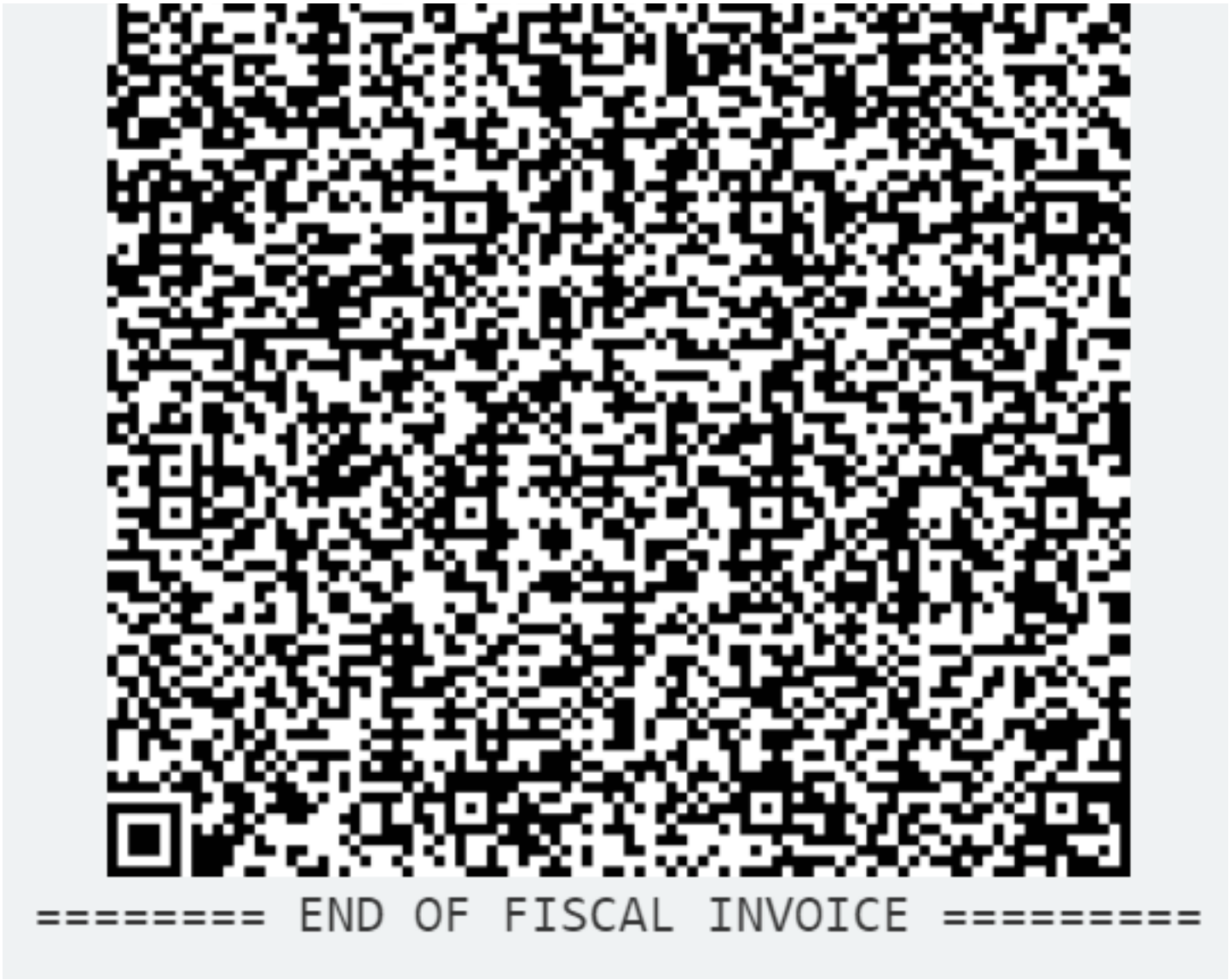
==========

SDC Time: 25/03/2020 09:29:58

SDC Invoice No: XRSYSZWL-373ZQXF4-1

Invoice Counter: 1/1NS

==========



Anatomy of a Fiscal Receipt – Image of a fiscal receipt

Normal Refund Receipt

Receipt for Normal Refund Invoice must contain visible markings **REFUND**, below the receipt header and above the item description section. Totals on the refund receipt are displayed as negative values, starting with (-), except for Total Refunded. Tax Items are displayed as positive values.

For Refund transaction type **Ref no** element (reference to the original Normal Sale) is mandatory.

Example:

```
========== FISCAL INVOICE ===========
TIN:                           502579006
Company:                          Golf V
Store:                         Sun Store
Address:                     7 Someplace
District:                           Suva
Cashier TIN:                  123456789
POS number:                89347415-2017
POS time:            2018-03-09 14:57:25
Ref no:            7AF4D923-E3B30A31-234
------------NORMAL REFUND--------------
Items
=========================================
Name    Price        Qty.          Total
Sport-100 Helmet, Blue (E)
        34.99          10        -349.90
Mountain Bike Socks, M (A)
         9.03           4         -36.12
-----------------------------------------
Total Refunded:                   386.02
Payment Method:                     Cash
=========================================
Label        Name     Rate           Tax
E             STT    6.00%         19.81
A             VAT    9.00%          2.98
-----------------------------------------
Total Tax:                         22.79
=========================================
SDC Time:            2018-03-09 14:57:46
SDC Invoice No:    7AF4D923-E3B30A31-235
Invoice Counter:                4/235NR
=========================================
---- QR code omitted for simplicity ----

======== END OF FISCAL INVOICE =========
```

Normal Refund Receipt - Image of a normal refund receipt

Training or Proforma or Copy Receipt

Receipt for Training or Proforma or Copy Invoice must contain visible markings “TRAINING” or “PROFORMA” or “COPY”, below the receipt header and above the item description section.

Receipt must also contain **THIS IS NOT A FISCAL INVOICE** below the total amount payable. Font size is at least twice the size of the text on the receipt that specifies the total amount payable.

Training or Proforma or Copy receipt is produced in the same way as normal, with an exception that totals are not accounted for.

For Copy invoice type **Reference Document Number** element is mandatory.

Example:

```
===== THIS IS NOT A FISCAL RECEIPT =====
TIN:                              502579006
Company:                             Golf V
Store:                            Sun Store
Address:                        7 Someplace
District:                              Suva
Cashier TIN:                      123456789
POS number:                    89347415-2017
POS time:              2018-03-09 14:57:25
----------TRAINING SALE----------
Items
==========================================
Name     Price        Qty.           Total
Sport-100 Helmet, Blue (E)
         34.99          10          349.90
Mountain Bike Socks, M (A)
          9.03           4           36.12
------------------------------------------
Total Purchase:                     386.02
Payment Method:                       Cash
==========================================
      THIS IS NOT A FISCAL INVOICE
==========================================
Label        Name      Rate            Tax
E             STT     6.00%          19.81
A             VAT     9.00%           2.98
------------------------------------------
Total Tax:                           22.79
==========================================
SDC Time:              2018-03-08 14:57:46
SDC Invoice No:     7AF4D923-E3B30A31-236
Invoice Counter:                  1/236TS
==========================================
---- QR code omitted for simplicity ----
===== THIS IS NOT A FISCAL RECEIPT =====
```

Training or Proforma or Copy Receipt – Image of a training sale receipt

Audits

Audit data represent a machine-readable formatted fiscal invoice signed by a taxpayer's private key followed by journal data. Journal data is a textual representation of a fiscal invoice generated by E-SDC.

Content of audit data is kept in encrypted form (audit package) ensuring no changes have been made and that no one has been able to access its content after creation, except the Tax Service's system, after a successful audit process.

1. Encryption of Audit Data
   Encryption of audit data prevents access to sales data by unauthorized persons.

2. Format of the Audit Package
   The Audit Package is a textual file in JSON format:

Encryption of Audit Data

Encryption of audit data prevents access to sales data by unauthorized persons.

The only one that can decrypt audit data is the Tax Service's system software running on the Tax Service premises and used by the authorized personnel only.

Format of the Audit Package

The Audit Package is a textual file in JSON format:

```
AuditData {
Key (string),
IV (string),
Payload (string)
}
```

Field	Type	Description
Key	Base64 Encoded String	One-time symmetric key (256Bit long) encrypted using RSA with TaxCore public key
IV	Base64 Encoded String	Initialization vector Key encrypted using RSA and TaxCore public key
Payload	Base64 Encoded String	Base64Encoded JSON format of an invoice, as described in section Json Representation of the Invoice, encrypted with **Key** and **IV** using AES256 algorithm.

Commands

Introduction

Commands are a means of communication between the Tax Service's system and E-SDCs. Commands are stacked in the queue list on the server for a specific E-SDC and submitted to the E-SDC as a part of the response once it reports to the Tax Service's system using a remote or a Local audit.

Commands are always delivered as an array structure. Commands are executed in a consecutive order, starting from the first one in the array.

Below is the data structure of a single command:

```
Command {
      "CommandId": (GUID),
      "Type": (enum CommandsType),
      "Payload": (Json string)
    }

   enum CommandsType
    {
        UpdateTaxRates = 0,
        UpdateNTPServiceUrl = 1,
        UpdateVerificationURL = 2,
        UpdatePAC = 3, // reserved for later use
        TaxCorePublicKey = 4 // deprecated,
        EndProofOfAudit = 5
    }
```

CommandId is a unique identifier assigned by Tax Service's system. Once a command is successfully executed, TaxCore.API shall be notified as described in the Commands Results section of this document.

Type defines the type of command and a format of a Payload as described in the following sections. Valid values are defined by CommandsType enum.

Payload transfers the information form TaxCore.API to an E-SDC in JSON format. The format of each type is described in the following sections.

Command Types

1. Configure Time Server URL Command
   E-SDC updates the URL of the time server used to keep a local clock in sync. Payload is the URL of the target NTP server.

2. Set Tax Rates Command
   Payload contains a group of all new tax rates which shall be applied from the specified date and time. The structure of a group is defined in section Tax Rates. The date can be in the past or (more likely) in the future.

3. Update Verification URL Command
   As a part of the invoice fiscalization, an E-SDC creates a unique URL for generating a QR code and validates an invoice. Verification URL is returned to the POS as a part of the Response. This command tells the E-SDC which URL will be used to Create Verification URL.

4. Proof of Audit Command
   Proof of audit command payload is transmitted to the Secure Element applet on the smart card (End Audit APDU command) once the audit process is completed successfully on the Tax Service's system.

Configure Time Server URL Command

E-SDC updates the URL of the time server used to keep a local clock in sync. Payload is the URL of the target NTP server.

Set Tax Rates Command

Payload contains a group of all new tax rates which shall be applied from the specified date and time. The structure of a group is defined in section Tax Rates. The date can be in the past or (more likely) in the future.

Tax rates with the future date shall be stored in non-volatile memory and applied starting from the specified moment. If more than one group has the same date, the one with higher GroupId shall be applied.

The payload of the command is structured as follows:

```
{
"TaxRateGroup": {
"ValidFrom": "2017-07-02T00:00:00",
"GroupId": "1",
"TaxCategories": [{
"CategoryId": 1002,
"Name": "VAT",
"Type": 0,
"TaxRates": [{
"RateId": 1002,
"Rate": 6.0,
"Label": "A"
},
{
"RateId": 1003,
"Rate": 10.0,
"Label": "B"
},
{
"RateId": 1004,
"Rate": 20.0,
"Label": "C"
}],
"OrderId": 1,
},
{
"CategoryId": 1003,
"Name": "STT",
"Type": 1,
"TaxRates": [{
"RateId": 1005,
"Rate": 25.0,
"Label": "D"
}],
"OrderId": 2,
}]
}
}
```

```
}
```

Update Verification URL Command

As a part of the invoice fiscalization, an E-SDC creates a unique URL for generating a QR code and validates an invoice. Verification URL is returned to the POS as a part of the Response. This command tells the E-SDC which URL will be used to Create Verification URL.

Payload is a URL of the server used to verify invoices. Detailed instructions for Verification URL creation are explained in the Create Verification URL section.

Proof of Audit Command

Proof of audit command payload is transmitted to the Secure Element applet on the smart card (End Audit APDU command) once the audit process is completed successfully on the Tax Service's system.

The payload is a byte array encoded as a base64 string.

Commands Results

Commands Results is a confirmation to TaxCore.API that a certain command is executed.

##Model

```
CommandResults {
      "CommandId": (GUID),
      "Success": (boolean),
      "DateAndTime": (string)
    }
```

##Example

```
{
  "CommandResults": [
    {
      "CommandId": "945bb863-5c7f-4826-9ae3-26debcac331a",
      "Success": true,
      "DateAndTime": "2017-06-17T04:33:47+00:00"
    }
  ]
}
```

Processes

This section describes processes performed by an E-SDC.

It contains the following sections:

1. E SDC Initialization
   Prior to the first use, the E-SDC has to be initialized. E-SDC must have access to the Secure Element during the initialization process in order to establish a secure connection with the TaxCore.API to obtain a set of initialization commands. The initialization commands are explained in the section Commands.

2. Standard Operation
   This section contains a description of standard E-SDC operations.

3. Malfunctions and Non serviceable Devices
   If the Secure Element is damaged and its data cannot be restored from the card, but the E-SDC is operational, the Tax Service system shall be able to dump data from the E-SDC device and upload the audit packages using the same application used to upload audit packages submitted by a taxpayer.

E-SDC Initialization

Prior to the first use, the E-SDC has to be initialized. E-SDC must have access to the Secure Element during the initialization process in order to establish a secure connection with the TaxCore.API to obtain a set of initialization commands. The initialization commands are explained in the section Commands.

NOTE:
For instructions on how to download the initialization commands, see Get Initialization Commands.

After processing the received initialization commands, the E-SDC must upload configuration commands results to TaxCore.API.

In case of a poor or no internet connection, configuration commands results can be uploaded via file-based communication as explained in section E-SDC Stores a Command Execution Result to the SD Card or USB Drive.

NOTE:
Initialization of E-SDC has to be performed each time a new smart card is inserted into the reader.

Standard Operation

This section contains a description of standard E-SDC operations.

1. Enter PIN to Unlock the Secure Element
   Before the Secure Element applet can be used, a valid PIN code must be supplied from the POS using the Ethernet connection. Once the E-SDC receives a PIN code, it will try to execute the *Verify Pin* APDU command.

2. Fiscalization of An Invoice
   Invoice fiscalization is the main function of an E-SDC. Fiscalization is the process of handling invoice request from an accredited invoicing system in order to produce fiscal invoices.

3. Audit Process
   An audit is a process of sequential transferring of audit packages from an E-SDC to the Tax Service's system and handling the response generated by the Service's system for the specific device.

4. Notifications
   E-SDC device shall have an appropriate way to show the status of the device, information about the smart card and processes running on the E-SDC.

5. Sync Date and Time
   As an E-SDC is the source of date and time for the invoices, it is of the utmost importance to keep the device clock in sync.

6. Switching Smart Cards During Operation
   During normal operation, taxpayers/cashiers might switch the smart card they are using for issuing fiscal invoices.

7. E SDC Logging
   E-SDC must keep a log about all required error events. It must log every error chronologically by local date and time (exact hour and minute).

Enter PIN to Unlock the Secure Element

Before the Secure Element applet can be used, a valid PIN code must be supplied from the POS using the Ethernet

connection. Once the E-SDC receives a PIN code, it will try to execute the *Verify Pin* APDU command.

Depending on the provided PIN, the SE will remain either unlocked for further use or locked until a valid PIN is entered. E-SDC will send a response to the POS based on the result of the PIN Verify command execution.

It is important to note the Secure Element interprets data as byte containing digits, so the E-SDC must perform appropriate conversion before data is sent to the Secure Element. For example, if a PIN transmitted from a POS is "2017" (0x32 0x30 0x31 0x37 in ASCII hexadecimal representation), data sent to the SE shall be 0x02 0x00 0x01 0x07.

Fiscalization of An Invoice

Introduction

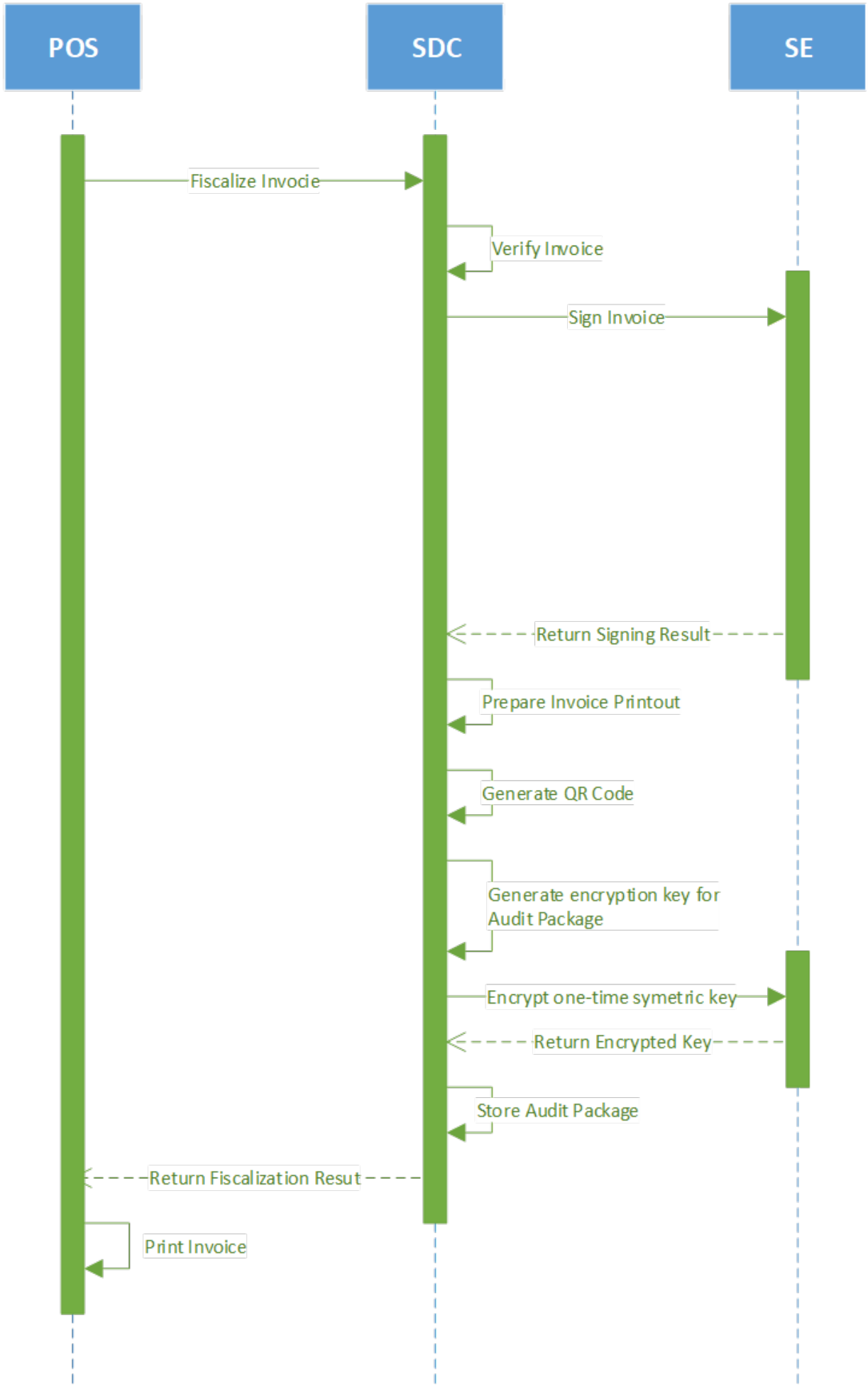
Invoice fiscalization is the main function of an E-SDC. Fiscalization is the process of handling invoice request from an accredited invoicing system in order to produce fiscal invoices.

Process

The following steps are executed by the E-SDC once a request data is received from an Accredited POS:

1. POS generates a request data and sends it as a request to the E-SDC using JSON via HTTP protocol;
2. E-SDC verifies format of the invoice;
3. E-SDC calculates taxes based on the current tax rates;
4. E-SDC sends the invoice data to the Secure Element for fiscalization providing current date and time and PIN code/password if required;
5. Secure element signs the invoice and returns the data to the E-SDC;
6. E-SDC produces a journal – a textual representation of an invoice;
7. E-SDC generates a verification URL;
8. [optionally] E-SDC creates QR Code – a graphical representation of a verification URL;
9. E-SDC creates an invoice with all mandatory elements (receipt data, previously generated signature, verification URL and journal), generates a one-time key and encrypts the invoice using a symmetric algorithm. The E-SDC encrypts a one-time symmetric key using the Tax Service's system public key and adds it to the package so the Tax Service's system decrypts the symmetric key and access the package content once it arrives to the Service's system.
10. E-SDC returns a response to the POS and optionally generated journal data.

The process is illustrated in the figure below.



Fiscalization of An Invoice – Image of the fiscalization process

Content

Calculate Taxes

Taxes are calculated by an E-SDC after a POS has sent a valid request. The tax amount for particular items on an invoice is defined by the tax labels associated with an item.

Process of a tax calculation depends on:

- Invoice and Transaction Type
- the tax rates for each label associated with an item on an invoice
- the Type value of tax category to which the label belongs

A POS sends an invoice fiscalization request with the line items. Items are sent with the total amounts (taxes included) and zero or more tax labels associated with them, which participated in the total price calculation.

In order to calculate a tax, the following algorithm shall be implemented:

1. Make an array of distinct tax labels associated with the items in the POS request (e.g. A, B, C, F, …).
2. Calculate the tax amount for each individual label in the array:
    - Iterate through all items in the POS request
    - For each item, calculate tax amounts. One item has one or more tax labels, and each label represents a tax amount. Each tax amount is a part of an item's total price. These tax amounts are calculated as follows:
        - If an item has a label from the **amount-on-quantity** category applied, subtract the tax rate amount for that label, multiplied with quantity, from the item total price. The resulting amount (the remainder), is used in all further calculation steps instead of item total amount.
        - If none of the labels' tax category type is ***tax-on-total*** (category 1):
            - Tax amount for one label is:

$$\frac{item\ total\ amount\ *\ label\ rate}{(100 + \Sigma(all\ tax - on - net\ rates\ on\ item))}$$

Example 1: An item has a total price of 10$ and applied labels: A(5%) and B(6%).

$$A = \frac{10\$ * 5}{(100 + \Sigma(5+6))}\quad B = \frac{10\$ * 6}{(100 + \Sigma(5+6))}$$

Tax amount for label A=0.4505$ and for label B=0.5405$.

- If any of the labels' tax category is ***tax-on-total*** (category 1):
    - Tax amount for every label whose category type is ***tax-on-total*** (category 1) is:

$$\frac{item\ total\ amount}{(1\ +\ \Sigma(all\ tax - on - total\ rates)/100)} * \frac{label\ rate}{100}$$

- Tax amount for every other label from category 0 is:

$$\frac{item\ total\ amount}{(1\ +\ \Sigma(all\ tax - on - total\ rates)/100)} * \frac{label\ rate}{(100 + \Sigma(all\ tax - on - net\ rates\ on\ item))}$$

Example 2: Item has a total price 10$ and applied labels: A(5% tax-on-net), B(6% tax-on-net), C(3% tax-on-total) and F(4% tax-on-total).

$$A = \frac{10\$}{(1 + \Sigma(3+4)/100)} * \frac{5}{(100 + \Sigma(5+6))}$$

$$B = \frac{10\$}{(1 + \Sigma(3+4)/100)} * \frac{6}{(100 + \Sigma(5+6))}$$

$$C = \frac{10\$}{(1 + \Sigma(3+4)/100)} * \frac{3}{100}$$

$$F = \frac{10\$}{(1 + \Sigma(3+4)/100)} * \frac{4}{100}$$

Tax amount for label A=0.4210$, for label B=0.5052$, for label C=0.2804$ and for label F=0.3738$.

- ♠

  Add calculated labels' tax amounts to the label's total amount sum.

  Example 3: the request contains two items from Example 1 and Example 2, the total sum for labels are: A=0.8715$, B=1.0457$, C=0.2804$, F=0.3738$.

```
- Add fixed tax amounts, multiplied with quantity, to the respective labels' to

  Example 4: An item has quantity 2 with total price 10$ and applied labels: A(

  Example 5: An item has quantity 2 with total price 10$ and applied labels: A(
```

3. After all of the items have been processed, calculate the tax amount for all tax categories found in the request. One tax category can consist of one or more tax labels (e.g. A, B…). The tax amount for a tax category is a sum of all label tax amounts related to the category.

   Example 6: The request contains two items from Example 1 and Example 2. Labels A and B are VAT category, C is STT category and F is ET category. Total VAT=1.9172$, STT=0.2804$ and ET=0.3738$.

Once the Tax calculation is completed, assign GroupId of the active tax rate group to the field *TaxGroupRevision* of *InvoiceFiscalizationResult*.

Rounding

E-SDC shall round all amounts to 4 decimal places using the half-round up method.

Examples:

3.44445555666 → 3.4445

3.4440012345 → 3.4440

3.44466012345 → 3.4447

3.444116012345 → 3.4441

Create Verification URL

Verification URL is created based on values submitted by a POS to an E-SDC and values returned to the E-SDC from APDU commands as follows:

1. Byte array is created:

Start	Offset	Invoice Field	Description
0	1	Version	Current version is 0x02
1	8	RequestedBy	UID, ASCII encoding (e.g. JKGB3K14)
9	8	SignedBy	UID, ASCII encoding (e.g. JKGB3K14)
17	4	TotalCounter	Int32 Little Endian
21	4	DocTypeCounter	Int32 Little Endian
25	8	TotalAmount	TotalInvoiceAmount * 10000 as Uint64 bit Little Endian
29	8	DateAndTime	Unix Timestamp (number of milliseconds), 64bit unsigned integer Big Endian
37	1	InvoiceType	0x00 (Normal), 0x01 (Pro Forma), 0x02 (Copy), 0x03(Training)
38	1	TransactionType	0x00 (Sale), 0x01 (Refund)
39	1	BuyerId Length	Buyer ID length in bytes
40	?	BuyerId	ASCII Encoding
?	?	EncryptedInternalData	Encrypted Internal Data received from SE after Invoice Sign APDU command, 256 or 512 bytes long
?	256	Signature	Signature received from SE after Invoice Sign APDU command, 256 bytes long

2. Created byte array is encoded as base64 string, which is additionally encoded, to comply with the URL standards.
3. Encoded string is appended to the verification URL received from the *Update Verification URL Command*

Create a QR Code

QR code contains a Verification URL which is described created in the previous section Create Verification URL.

It is the most convenient way of exposing the Verification URL because it enables customers to easily scan their fiscal invoices using a QR code reader.

How to create a QR code

Base64 encoded string is created from GIF image bytes and attached to the Invoice Response

Important parameters for creating a QR code:

- Min size = 42x42mm
- ErrorCorrectionLevel = L
- FixedModuleSize = 4
- QuietZoneModules = Zero
- BlackAndWhite
- ImageFormat = Gif

For more information about using QR codes in the TaxCore system, see QR Code

Create a Textual Representation of an Invoice

Introduction

A textual representation of a Receipt shall be created as described in the chapter Anatomy of Fiscal Receipt. One row on a receipt is 40 characters long to fit 2.25 inch / 58 mm wide paper roll commonly used in thermal printers.

SDC Date and Time field printed on a journal (textual representation of an invoice) generated by E-SDC are **locally time-based**.

Any amount shall be rounded to 2 (two) decimal places using the half-round up method only on the textual representation of an invoice.

1. How To Obtain a Taxpayer Identification Number TIN
   Digital certificate exported using the *Export Certificate* APDU command (in DER format) contains taxpayer TIN and POS location (Shop or HQ Address that shall appear on the textual representation of the invoice)

TIN and POS location (Shop or HQ Address that shall appear on the textual representation of the invoice).

2. Mapping Subject to Invoice Fields
   Digital certificate exported using the *Export Certificate* APDU command (in DER format) contains taxpayer TIN and POS location (Shop or HQ Address that shall appear on the textual representation of the invoice).

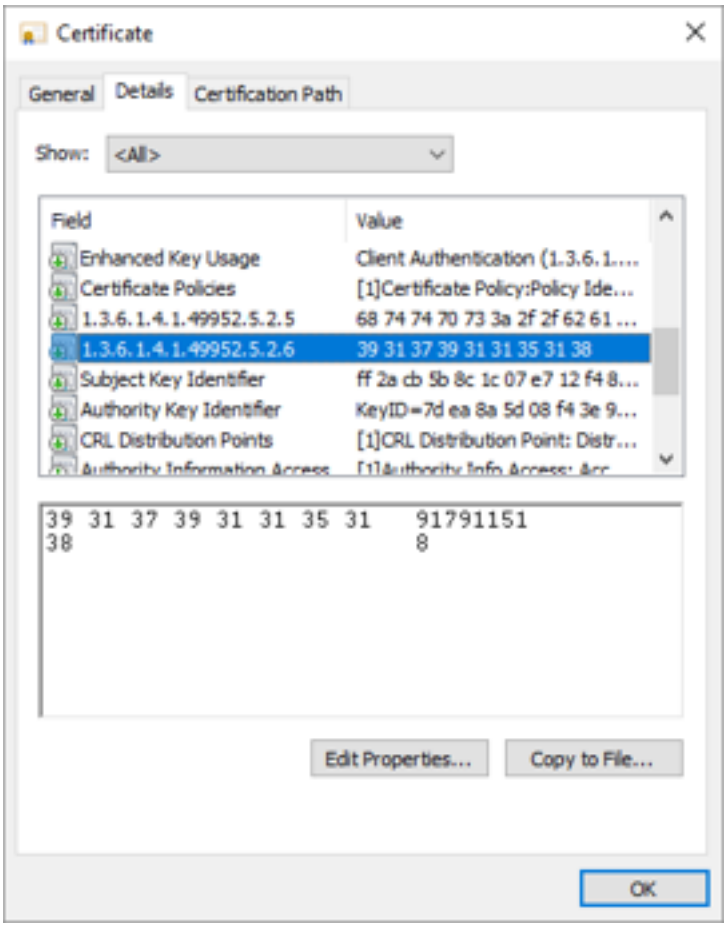
How To Obtain a Taxpayer Identification Number - TIN

Digital certificate exported using the *Export Certificate* APDU command (in DER format) contains taxpayer TIN and POS location (Shop or HQ Address that shall appear on the textual representation of the invoice).

TIN is stored in the digital certificate as an OID value. OID is dynamically created during a smart card personalization and depends on the target environment. The Test and Production environments will have different OIDs.

In order to use the same E-SDC with the Test and Production environments, the correct OID has to be constructed using the following procedure:

1. Get the certificate using the *Export Certificate* APDU command;
2. Read value of EnhancedKeyUsage (for example, 1.3.6.1.4.1.49952.5.2.3.3);
3. The fourth and the third integer to the right identify the environment;
4. Construct the OID that contains TIN, by replacing stars with the numbers using the following pattern - 1.3.6.1.4.1.49952...6;
5. For this example, resulting OID will be 1.3.6.1.4.1.49952.5.2.6;
6. Read the value of resulting OID containing Taxpayer TIN.



How to Obtain a Taxpayer Identification Number – TIN – Image showing the OID and TIN of the certificate

Mapping Subject to Invoice Fields

Digital certificate exported using the *Export Certificate* APDU command (in DER format) contains taxpayer TIN and POS location (Shop or HQ Address that shall appear on the textual representation of the invoice).

This example shows the mapping between a subject name/value pairs and invoice fields.

Subject Field (bolded parameters are always present in the subject):

E = someone@test.taxcore.dti.rs

CN = P22V International Trek Center

SERIALNUMBER = P22VC8VR

G = Albert

SN = Mungin

OU = International Trek Center

O = International Trek Center

STREET = 8844 Garcia

L = West Covina

S = California

C = US

Invoice Field	Subject Parameter Name	Note
TIN	N/A	obtained by OID as explained in How to Obtain Tax Identification Number (TIN)
Business Name	O	Legal name under which the business operates
Shop Name	OU	It may be the same as Business Name if the Company HQ and sales location are the same.
Address	STREET	Street name and number
State	S	State, District or Region
Country	C	ISO 2-letter Country Code. Optional field on the textual representation of the invoice

Creating an Audit Package

Once an invoice is created (*InvoiceFiscalizationRequest* and *InvoiceFiscalizationResult*) the E-SDC is ready to create an audit package and store it in the non-volatile memory. In order to achieve that, follow these steps:

1. Convert all Date and Time data to UTC;
2. Generate a random one-time symmetric key for AES256;
3. Encrypt string JSON representation of the invoice using the one-time key;
4. Convert the encrypted invoice to base64 string and store it in the Payload field of *Json Representation of the Invoice*;
5. Get the TaxCore Public key using *Export TaxCore Public Key* APDU command.
6. Encrypt the one-time key using RSA with TaxCore public key, convert it to base64 string and store it in the Key field;
7. Encrypt Initialization Vector (IV) using RSA with TaxCore public key, convert it to base64 string and store it in the IV field;
8. Save the Audits as an Audit Package file, named as {**UID**}-{**UID**}-{**Ordinal_Number**}.json;
9. (Optionally) Generate a QR code, and attach it to *InvoiceFiscalizationResult* (make sure that the QR code is not stored in the Audit Package);
10. Return InvoiceFiscalizationResult to the POS;
11. If the internet connection is available try to send the Audit Data to TaxCore.API as explained in the section

Remote Audit;

Audit Process

Introduction

An audit is a process of sequential transferring of audit packages from an E-SDC to the Tax Service's system and handling the response generated by the Service's system for the specific device.

There are two specific scenarios: **Remote Audit** and **Local Audit**. Basic rules and processes described in this section apply to both scenarios. Details are explained in separate sections.

An audit is always an asynchronous process. Depending on the amount of data and means of communication, it can take from less than a second to a couple of hours.

Once the E-SDC receives a response (signed invoice and journal) from the Secure Element, it shall be encrypted and stored in the non-volatile memory.

An E-SDC device must be fully functional during an audit. The POS must be able to sign new invoices as long as the Secure Element permits. There must be a mechanism in place that is responsible for the continuous operation of the Secure Element and E-SDC while audit packages are being transmitted to the Tax Service's system.

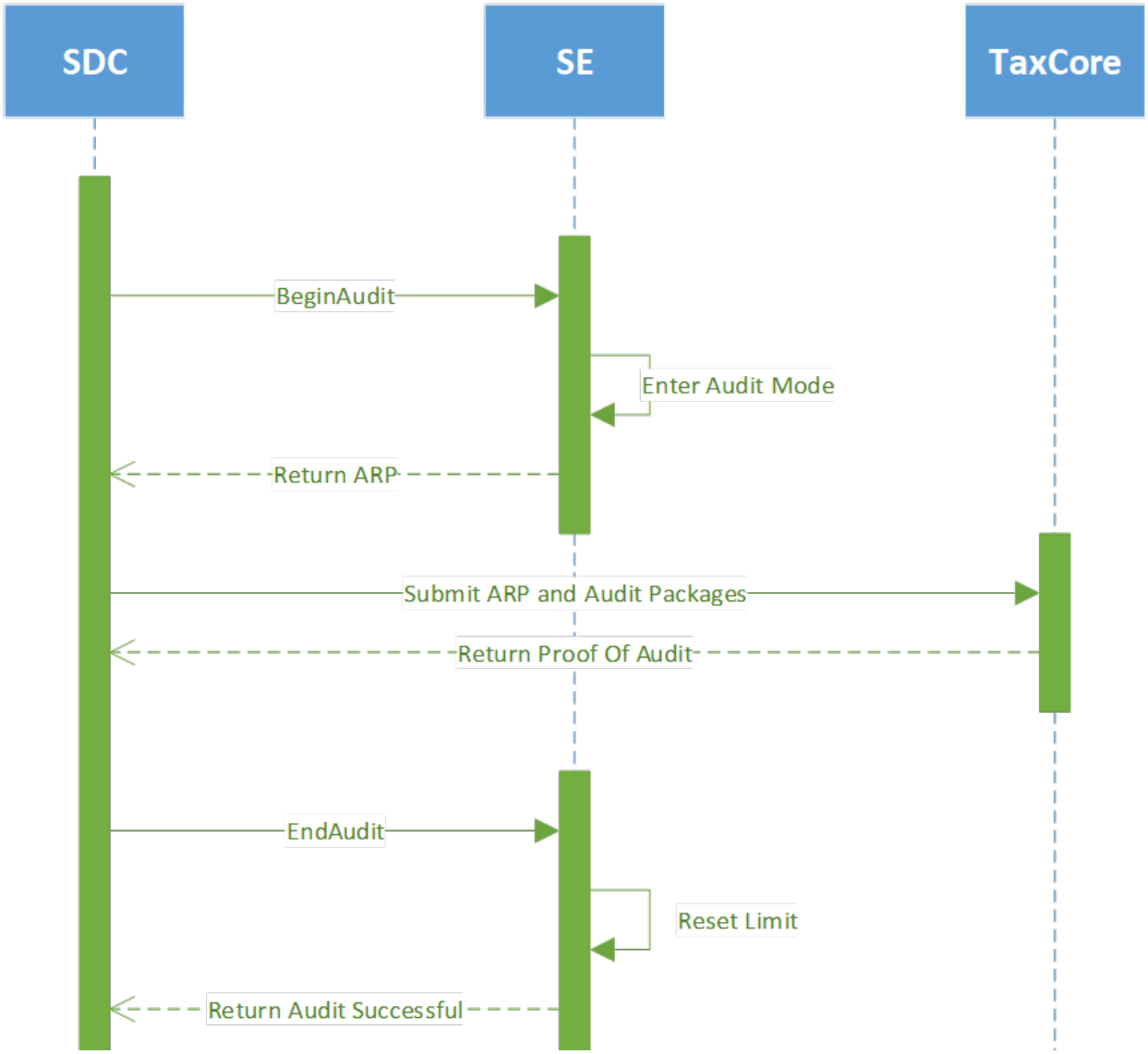
Depending on the connection availability, an audit may be triggered by the arrival of a signed invoice from the Secure Element or after the insertion of an external memory device into the E-SDC. Regardless of the event which has triggered the audit, the following conversation shall take place between the E-SDC, the Tax Service's system and the Secure Element:

1. E-SDC signals the beginning of the audit to the Secure Element (Invokes *Start Audit* APDU command);
2. The Secure Element returns ARP (256 bytes) to the E-SDC;
3. E-SDC starts the audit by sending audit data (over HTTPS) or dumping them on external memory (e.g. SD card, USB flash drive), starting with the oldest unaudited package, in a piecemeal fashion. ARP is sent to the Tax Service's system using the same communication channel;
4. If verification is successful, the Tax Service's system shall generate a proof of audit (PoA) and return it as a *Proof of Audit Command*;
5. E-SDC receives the proof of audit command and passes the payload to the End Audit APDU command;
6. The Secure Element verifies if proof of audit is valid, meaning the audit data has been successfully received by the Tax Service's system;
7. If proof of audit is valid, the Secure Element will conclude the audit process;

ARP should be generated and sent to Tax Service's system periodically (for example, after each 200 audit packages are submitted to TaxCore.API or once per day regardless of the number of submitted audit packages) if HTTP-based communication is used to submit audit packages.

ARP should be generated and saved as a file each time at least one audit package is submitted to the Tax Service's system using a USB memory stick (File-based communication).

The audit process sequence is illustrated in the figure below:



Audit Process - Image of the audit process sequence

Content

1. Remote Audit
   Remote audit is the process of transferring data to the Tax Service's system using the internet connection. It is the most common way to perform audits for any occasionally connected device.

2. Local Audit
   Local audit initiated by a taxpayer is a common scenario for devices that lack the ability to connect to the internet due to the technical limitations of the devices or limited infrastructure.

3. Proof of Audit
   Proof of Audit is generated by the Tax Service's system once all expected audit packages have been received and securely stored on the Tax Service's system.

Remote Audit

Remote audit is the process of transferring data to the Tax Service's system using the internet connection. It is the most common way to perform audits for any occasionally connected device.

An E-SDC checks if TaxCore.API is reachable. If TaxCore.API is reachable, the E-SDC authenticates the Tax Service's system by using a server-side certificate installed on the TaxCore.API endpoint, enabling HTTPS protocol. The Tax Service's system authenticates the E-SDC using a digital certificate issued on the Secure Element. The E-SDC starts sending audit packages, performing a series of audits until no more unaudited data is stored on its non-volatile memory.

A Remote audit is not the only audit option for E-SDC. If the network connection is not available due to the interruption of the service or a missing GPRS modem or network card, E-SDC will still be able to perform a Local audit.

Local Audit

Local audit initiated by a taxpayer is a common scenario for devices that lack the ability to connect to the internet due to the technical limitations of the devices or limited infrastructure.

An audit is initiated by inserting an SD card or a USB Flash drive to an E-SDC device.

During the Local Audit, the E-SDC doesn't submit the ARP and Audit packages to TaxCore.API; instead those files are saved to an SD Card or a USB Flash Drive.

Prescribed data formats for the file-based Local Audits are described in section E-SDC Stores Audit Files on SD Card or USB Drive.

Audit files are uploaded using the Taxpayer Administration Portal (section Upload Audit Packages).

The process of Local Audit:

1. Transfer the audit packages and the ARP file from an E-SDC to an SD card or a USB Flash drive
2. Upload the audit packages using the section Upload audit packages on the Taxpayer Administration Portal
3. Check if there are pending commands for your E-SDC using the section Download Commands on the Taxpayer Administration Portal
4. If there are pending commands, download them to the SD Card or USB Flash Drive
5. Transfer the commands to the E-SDC
6.

Upload the confirmation from E-SDC about receiving commands using the Upload Commands Status section on the Taxpayer Administration Portal

7. Submitting Data Using a Web Application
   Audit packages (up to 30Mb) can be sent to a Tax Service using a public web site. The Service's system shall verify received audit packages and generate the proof of audit as a response.

8. Completing an Audit in Progress
   A taxpayer inserts media with proof of audit file into an E-SDC. The E-SDC loads proof of audit and verifies if the format is valid. If the format is valid, proof of audit is sent to the Secure Element for processing.

Submitting Data Using a Web Application

Audit packages (up to 30Mb) can be sent to a Tax Service using a public web site. The Service's system shall verify received audit packages and generate the proof of audit as a response.

A user will be required to manually delete audit packages from the media and save received proof of audit for later use.

Completing an Audit in Progress

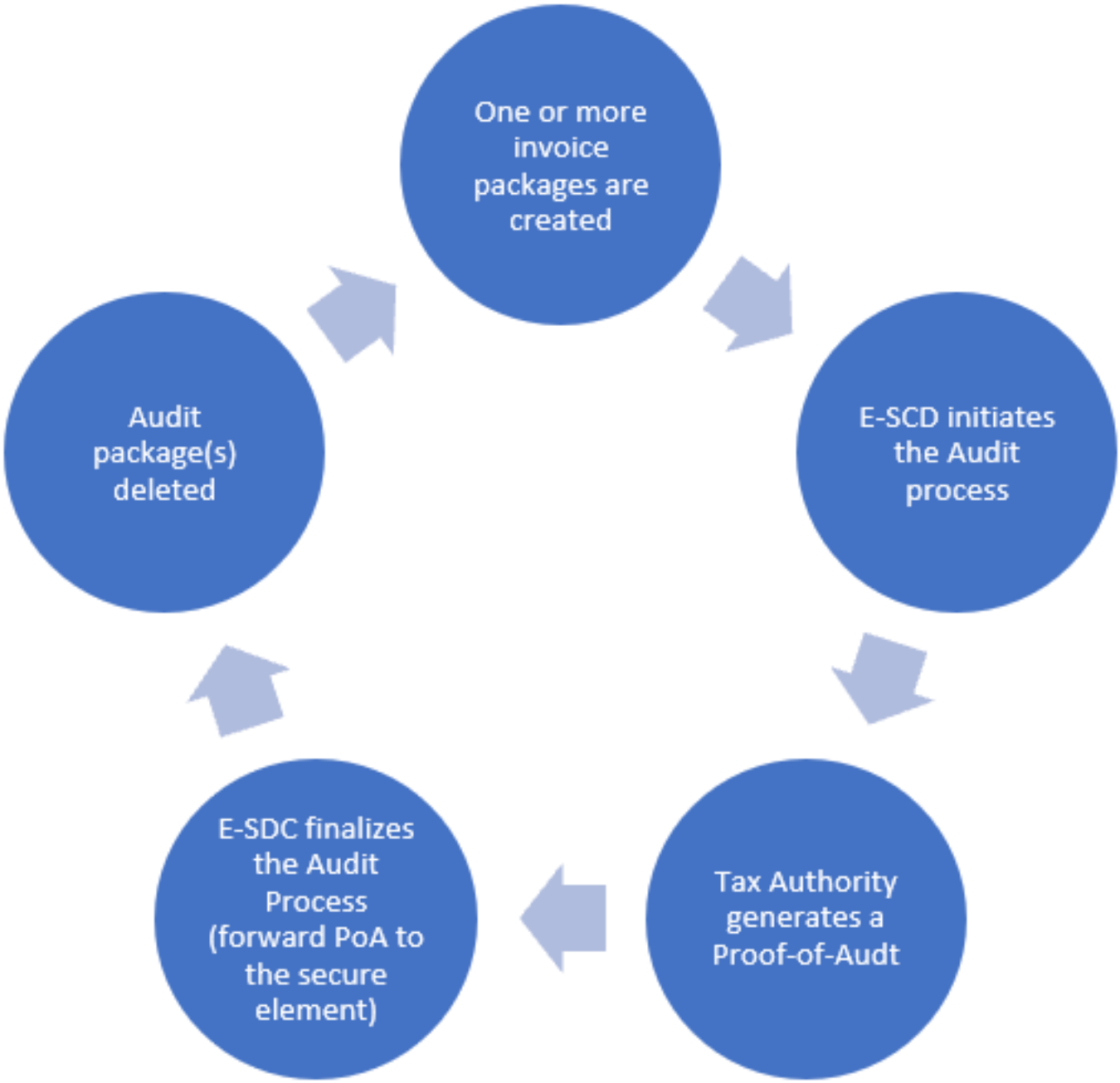
A taxpayer inserts media with proof of audit file into an E-SDC. The E-SDC loads proof of audit and verifies if the format is valid. If the format is valid, proof of audit is sent to the Secure Element for processing.

If the format is invalid or the E-SDC and the Secure Element cannot process proof of audit for any reason, the E-SDC signals error message to the operator.

Proof of Audit

Proof of Audit is generated by the Tax Service's system once all expected audit packages have been received and securely stored on the Tax Service's system.

An Audit cycle begins by E-SDC initiating the *BeginAudit APDU command* (Audit Start) - the period between two Audit Starts must be at least 5 minutes. An Audit cycle finishes by the E-SDC receiving and forwarding a Proof-of-Audit to the secure element (Audit End).



Proof of Audit – Image of the Audit cycle

This means that there are three possible scenarios for completing the Audit cycle:

1. One audit package is created between completing two Audit Ends - two Proof-of-Audits (Case 1 below)
2. Multiple audit packages are created and one Audit Start is initiated between two Audit End s- two Proof-of-Audits (Case 2 below)
3. Multiple audit packages are created and multiple Audit Starts are initiated between two Audit Ends - two Proof-of-Audits (Case 3 below).

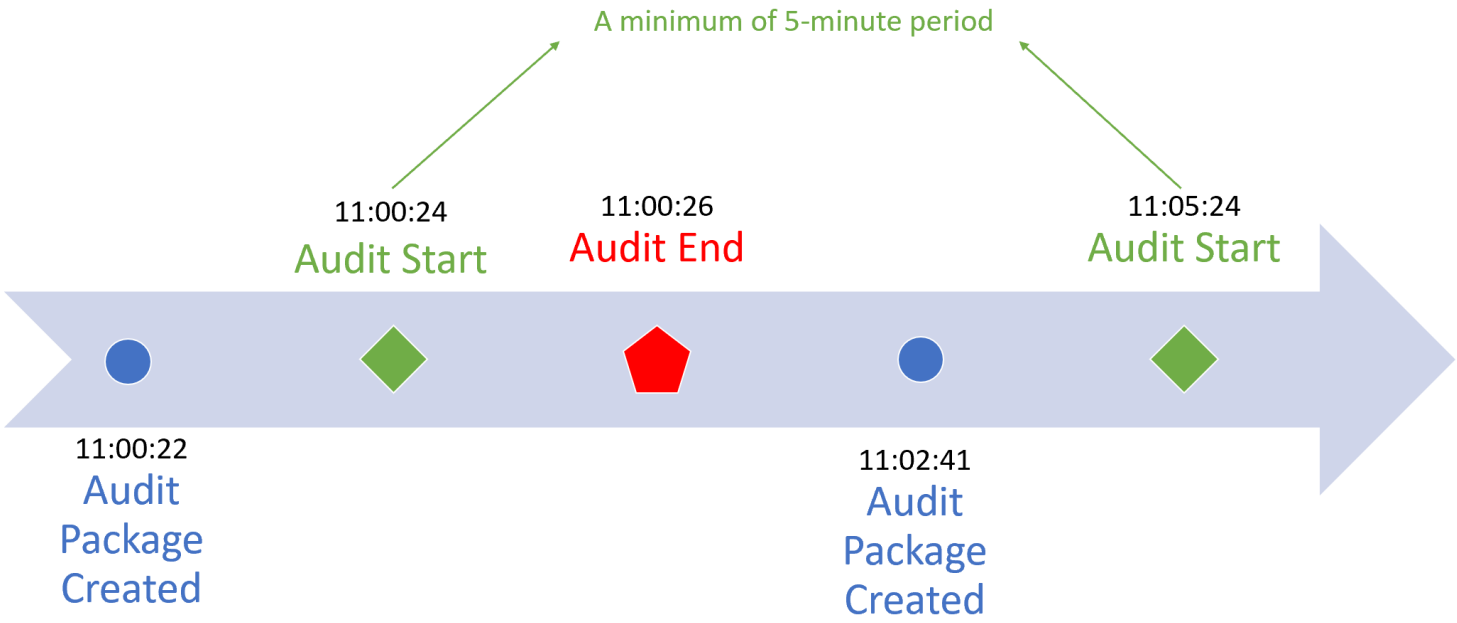
Case 1 – Audit is performed after the creation of each audit package

This is the simplest case, where no additional audit packages are generated during the whole audit process, as following:

1. Create an audit package
2. Initiate the Audit process by invoking BeginAudit APDU command

3. Receive a proof of audit and pass it to EndAudit APDU command
4. If EndAudit returns the value "true", you can safely delete the audit package
5. If EndAudit returns the value "false", continue until a valid proof of audit is received
6. The period until the next Audit Start must be at least 5 minutes after the previous Audit Start

The figure below illustrates the process:



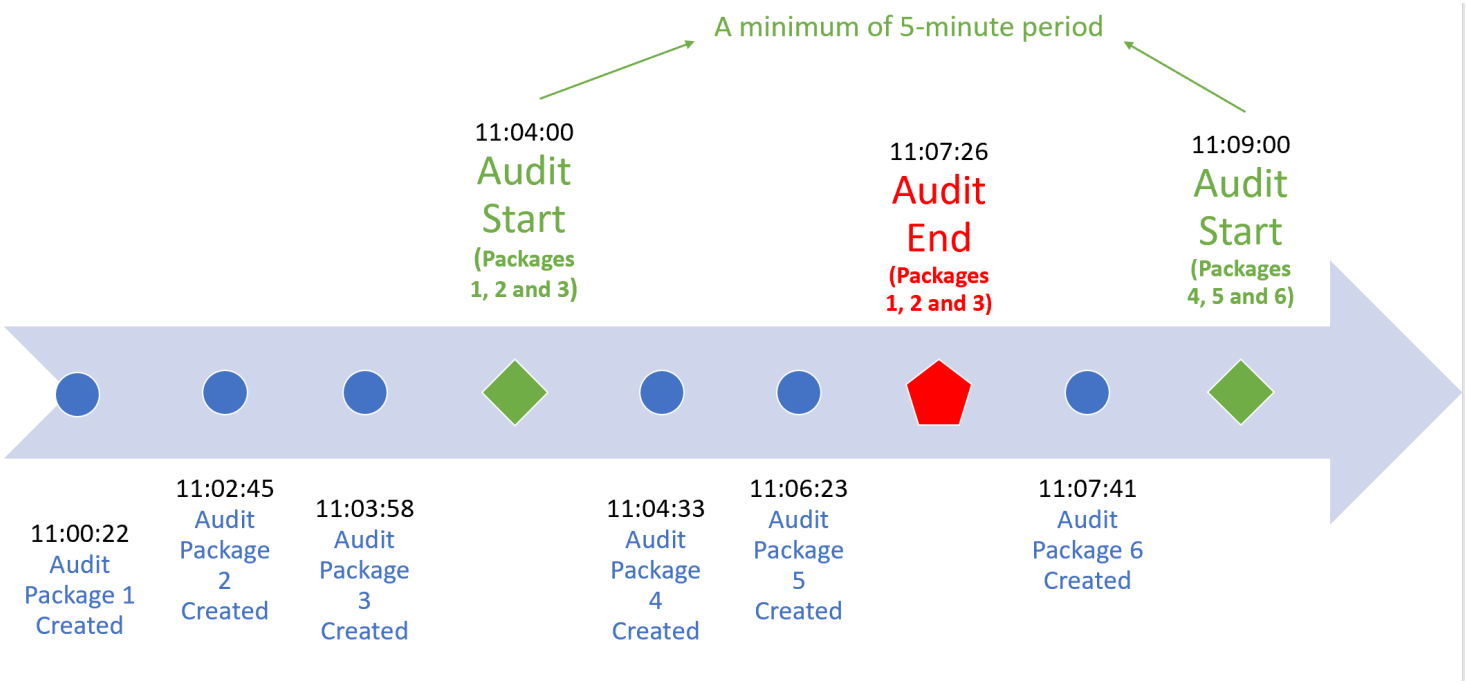
Proof of Audit – Image showing that the audit is performed after the creation of each audit package

Case 2 – Audit is performed after multiple audit packages have been created

In this case, new packages can be created after an audit start:

1. Create audit packages 1-3 (as shown on diagram)
2. Initiate the audit process by invoking BeginAudit APDU command
3. Continue to fiscalize invoices and create audit packages 4-6
4. Receive a proof of audit and pass it to EndAudit APDU command
5. If EndAudit APDU command returns value true you can delete remaining audit packages 1-3 because it is the last initial audit being invoked by E-SDC. Audit packages 4-6 are created after the call to BeginAudit APDU command so they are not audited in this cycle
6. If EndAudit APDU command returns value false, continue (return to point 1) until a valid proof of audit is received
7. The period until the next Audit Start must be at least 5 minutes after the previous Audit Start

The figure below illustrates the process:



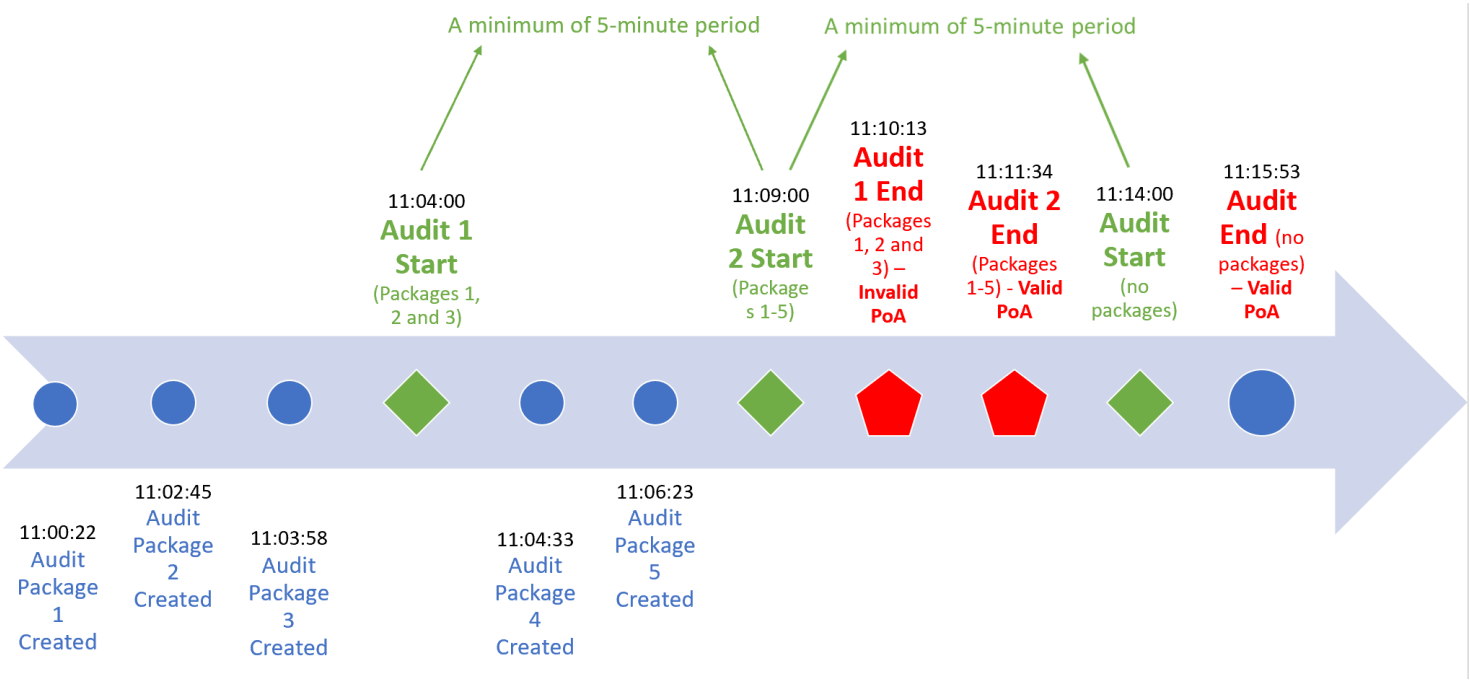
Proof of Audit – Image showing that the audit is performed after multiple audit packages have been created

Case 3 – Audit is started multiple times before the first proof of audit arrived

This case involves multiple audit starts:

1. Create Audit Packages 1-3
2. Initiate the Audit process by invoking BeginAudit APDU command
3. Continue to fiscalize invoices and create Audit Packages 4 and 5
4. Initiate another audit process by invoking BeginAudit APDU command – the previous audit is canceled
5. Receive the Proof-of-Audit and pass it to EndAudit APDU command
6. If EndAudit returns value true you can delete remaining audit packages 1-5 because it is the last BeginAudit being invoked by E-SDC.
7. If EndAudit APDU command returns value false, continue until a valid Proof-of-Audit is received
8. A Proof-of-Audit generated for the first Audit Start (Audit 1 below) is not considered valid. Only the Proof-of-Audit which is generated for the last Audit Start (Audit 2 below) is considered valid and will be forwarded to the secure element.
9. The period until the next Audit Start must be at least 5 minutes after the previous Audit Start

The figure below illustrates the process:



Proof of Audit – Image showing that the audit is started multiple times before the first proof of audit arrived

Notifications

E-SDC device shall have an appropriate way to show the status of the device, information about the smart card and processes running on the E-SDC.

A cashier could get the device notifications by receiving an onscreen message, by observing the colors from the light-emitting diodes (LED) or any other similar component set for displaying visual notifications.

The following visual notifications shall be available to a cashier:

1. Smartcard is inserted but the E-SDC is not yet configured with the tax rates, verification URL or NTP service address. This is a common situation before initialization commands are executed by E-SDC;
2. Enter PIN Code for the Secure element – Smart card is inserted but E-SDC has not received the PIN Code from POS;
3. E-SDC is ready to sign an invoice;
4. Smart card is missing or unavailable;
5. Audit package transfer is in progress (Local audit on an SD card or USB flash drive, or an online audit);
6. Firmware update is in progress (if applicable);
7. Audit data storage is almost full;
8. Audit data storage is full;
9. Time for audit;
10. Commands in progress (currently running)

Sync Date and Time

As an E-SDC is the source of date and time for the invoices, it is of the utmost importance to keep the device clock

in sync.

If the internet connection is available, the E-SDC shall sync time with the recommended NTP service at least once every 48h.

If the E-SDC does not support online or semi-connected operation modes, the manufacturer shall provide and document a simple way to check, set and keep date and time in sync on the E-SDC.

Switching Smart Cards During Operation

During normal operation, taxpayers/cashiers might switch the smart card they are using for issuing fiscal invoices.

In that case, the E-SDC must perform the following activities:

- **if the new smart card is for the same environment** - the E-SDC will first submit the unsubmitted invoices that were created with the previously used smart card
- **if the new smart card is for a different environment** - the E-SDC will keep the unsubmitted invoices (created with the previously used smart card) in its internal memory until they can be submitted (old smart card is returned)

For more information about different environments, see Identification of Environments.

E-SDC Logging

E-SDC must keep a log about all required error events. It must log every error chronologically by local date and time (exact hour and minute).

E-SDC log must be available for easy export (download, USB flash drive...) and presented in a human-readable format.

The following error events must be logged:

- Any Invoice Request sent by POS that E-SDC failed to process
- Any APDU error returned by SE
- Any error returned by TaxCore
- Any error caused by internal E-SDC operations
- Any error during the E-SDC Initialization process (Enter PIN and Command processing).
- Any error during the Invoice Fiscalization process
- Any error during the Audit process (Local and Remote).
- Any error during the process of Date and Time Synchronization

The above errors are the minimum requirements, but E-SDC can also keep a log of other events.

Malfunctions and Non-serviceable Devices

Dump Audit Packages Kept on E-SDC when Secure element is damaged

If the Secure Element is damaged and its data cannot be restored from the card, but the E-SDC is operational, the Tax Service system shall be able to dump data from the E-SDC device and upload the audit packages using the same application used to upload audit packages submitted by a taxpayer.

Protocols

This section describes Application Programming Interfaces (API) and protocols exposed by an E-SDC or used by an E-SDC to communicate with the other components (TaxCore.API, Secure element Applet, PKI Applet or SD Card/USB Flash Drive) required to fulfill its primary role – to safeguard a transaction and to transfer the audit packages to the Tax Service's system.

Accredited POS systems can communicate with the E-SDC using the POS to SDC Protocol.

1. Secure Element Applet API
   Communication with a Secure element Applet API is performed through standard APDU commands.

2. File Based Communication
   This section contains the description of the **File-based** communication with E-SDC.

Secure Element Applet API

Communication with a Secure element Applet API is performed through standard APDU commands.

For a detailed description of APDU communication, APDU commands data structure and particular bytes meaning, please refer to ISO/IEC 7816-4 standard.

Commands are grouped into three categories based on the type of usage:

1. Fiscalization
2. Audit

Important Notes

1. All APDU commands are sent to the Smart Card using T1 communication protocol

1. All APDU commands are sent to the Smart Card using T1 communication protocol
2. All amount values are submitted to the Secure element using Big-endian. Big-endian is an order in which the "big end" (most significant value in the sequence) is stored first (at the lowest storage address)
3. P1 and P2 values are not considered in the request processing, except for the Select Applet Command
4. All APDU commands are sent to the Smart Card using T1 communication protocol
5. All values are submitted to the Secure element using Big-endian. Big-endian is an order in which the "big end" (most significant value in the sequence) is stored first (at the lowest storage address)

Content

General Commands

Secure Element Applet is installed as a non-default applet on a smart card. Before any APDU command is invoked, the applet is selected using the standard Select command.

Select Applet

As previously mentioned, the Smart Card has two applets installed. This command selects the Secure element applet and routes subsequent APDU commands to it.

IsoCase: Case4Short

Class: 0x00

Instruction: 0xa4

Data: 0x10 0xA0 0x00 0x00 0x07 0x48 0x46 0x4A 0x49 0x2D 0x54 0x61 0x78 0x43 0x6F 0x72 0x65

Le: 0x00

Example: 0x00 0xA4 0x04 0x00 0x10 0xA0 0x00 0x00 0x07 0x48 0x46 0x4A 0x49 0x2D 0x54 0x61 0x78 0x43 0x6F 0x72 0x65 0x00

Export Certificate

Exports the taxpayer certificate in a DER format. This certificate contains location data that is present on the textual representation of an invoice.

IsoCase: Case2Extended

Class: 0x88

Instruction: 0x04

Example: 0x88 0x04 0x04 0x00 0x00 0x00 0x00

Get Secure Element Version

IsoCase: Case2Short

Class: 0x88

Instruction: 0x09

Example: 0x88 0x09 0x04 0x00 0x00

Fiscalization

PIN Verify

PIN verification is a method that “unlocks” a card for invoice signing and other operations protected by PIN code. PIN is in a decimal format, example PIN:2017 is represented as 0x02, 0x00, 0x01, 0x07

IsoCase: Case3Short

Class: 0x88

Instruction: 0x11

Example: 0x88 0x11 0x04 0x00 0x04 0x02 0x00 0x01 0x07 (for Pin 2017)

Signature

Sign Invoice

Signs invoice and returns fiscalization data for a submitted invoice.

IsoCase: Case4Extended

Class: 0x88

Instruction: 0x13

Request Data

Start (byte)	Length (byte)	Field	Description
0	8	Date/time	E-SDC timestamp UTC time in Unix Timestamp. Example: 1495018011910 is 2017-05-17T10:46:51.910Z
8	20	Taxpayer ID	Hex encoded byte array, leading bytes filled with 0x00; MSB are sent first Example: Taxpayer ID = 928615467, Byte array = {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x39, 0x32, 0x38, 0x36, 0x31, 0x35, 0x34, 0x36, 0x37} (byte 0x37 is sent last to SE)
28	20	Buyer ID	If unknown, leave zeroes, formatting is the same as for Taxpayer ID
48	1	Invoice type	Values 0, 1, 2, 3, as explained in section Model.
49	1	Transaction Type	Sale=0, Refund=1
50	8	Invoice amount	Sale or refund total amount (including taxes) - depends on applied tax types
58	1	Number of tax categories	Defines the number of tax categories which appear on the invoice (value between 0 and 26). The following data structure **Tax categories** must be repeated exactly this number of times.
77	...		

Tax Categories

Start	Length	Field	Description

Start (byte)	Length (byte)	Field	Description
59	[1]	[Tax category ID]	The first tax category's OrderID, as explained in Tax Rates section (mandatory if **Number of tax categories** > 0)
60	[8]	[Tax category amount]	The first total tax amount for the category specified in preceding field **Tax category ID** (mandatory if **Number of tax categories** > 0)
68	[1]	[Tax category ID]	The next tax category's OrderID (mandatory if **Number of tax categories** > 1)
69	[8]	[Tax category amount]	The next total tax amount for the category specified in preceding field **Tax category ID** (mandatory if **Number of tax categories** > 1)

Request data tables

Response Data

Start (byte)	Length (bytes)	Field	Description	
0	8	Date/time	Same as data sent from E-SDC to SE	
8	20	Taxpayer ID	Same as data sent from E-SDC to SE	
28	20	Buyer ID	Same as data sent from E-SDC to SE	
48	1	Invoice type	Same as data sent from E-SDC to SE	
49	1	Transaction type	Same as data sent from E-SDC to SE	
50	8	Invoice amount	Same as data sent from E-SDC to SE	
58	8	Sale or refund counter value	Depends on request's Tax type field	
66	8	Total counter value (sale+refund)	unsigned int 64bit big endian,	

74	256	512	Encrypted Internal Data	Encrypted Internal Data length depends on the number of available tax rates programmed during personalization. It may be 256 or 512 bytes long.
330	586	256	Digital signature	
586	842			

Response data table

Example: 0x88 0x13 0x04 0x00 0x00 0x00 0x44 0x00 0x00 0x01 0x5D 0x79 0x40 0x3B 0x69 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x30 0x32 0x31 0x31 0x33 0x31 0x36 0x38 0x32 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x24 0x9F 0x00 0x01 0x03 0x00 0x00 0x00 0x00 0x00 0x02 0x12 0xA9 0x00 0x00

Amount Status

Returns 16 bytes long data structure (8 bytes for sum SALE and REFUND, and 8 bytes for Limit Amount)

IsoCase: Case2Short

Class: 0x88

Instruction: 0x14

Example: 0x88 0x14 0x04 0x00 0x00

Audit

Export TaxCore Public Key

Returns 259 bytes data structure represents public card key (256 bytes modulus and 3 bytes exponent). This key is used for Audits.

IsoCase: Case2Extended

Class: 0x88

Instruction: 0x07

Example: 0x88 0x07 0x04 0x00 0x00 0x00 0x00

Export Internal Data

Exports encrypted Internal Data structure only (256 or 512 bytes).

Class: 0x88

Instruction: 0x12

Example: 0x88 0x12 0x04 0x00 0x00

Start Audit

Notifies the Secure element that the audit process has been initialized by E-SDC.

Secure element returns an encrypted message that shall be submitted to TaxCore as Audit Request Payload.

IsoCase: Case2Extended

Class: 0x88

Instruction: 0x21

Example: 0x88 0x21 0x04 0x00 0x00 0x00 0x00

End Audit

Notifies the Secure element that the audit process has been finalized by TaxCore. If APDU Command status is OK (0x90 0x00) consider audit operation is successfully completed.

IsoCase: Case3Extended

Class: 0x88

Instruction: 0x20

Example: 0x88 0x20 0x04 0x00 0x00 0x01 0x00 0x53 0x8B 0x46 0xC8 0x86 0x48 0x74 0xE4 0x33 0x46 0xA7 0x13 0x81 0x58 0x5E 0xF4 0xD6 0xDC 0xB8 0xB9 0x92 0x42 0x23 0x1B 0xCA 0x60 0xAD 0x41 0x0A 0x70 0x74 0x7B 0xD4 0x8D 0x5F 0xA1 0x21 0x18 0x85 0x07 0x73 0x6B 0x24 0xA3 0x3E 0x4F 0xFE 0x98 0x8C 0x99 0xC2 0x4E 0x77 0x2E 0xF9 0x6F 0xF8 0x72 0x99 0xB8 0x20 0x16 0x2F 0xAD 0xC6 0x97 0xCD 0x42 0xC0 0xA9 0xF1 0x96 0xF8 0x22 0x00 0x7C 0xD4 0xD1 0xE9 0x41 0x19 0x33 0x24 0xF4 0xB0 0x01 0xE1 0x6D 0x40 0xEB 0x9D 0xE1 0xC3 0xBE 0xBE 0x22 0x67 0x4B 0xAC 0xA6 0x23 0x99 0x3F 0xF5 0xA5 0xA2 0x7F 0x67 0x7A 0x01 0x8B 0xC8 0x3E 0x45 0x08 0x7E 0x34 0xCD 0xEA 0x2F 0x0B 0xCF 0x59 0x5F 0xCE 0x9D 0x6B 0xFE 0x36 0x80 0x85 0x86 0x40 0xD3 0xB4 0x3F 0xD7 0x06 0x90 0x79 0x35 0xCE 0x07 0x4B 0x9F 0xAA 0xB8 0x70 0x95 0x5F 0xAC 0x15 0x40 0xE2 0x8A 0x0D 0x5C 0x81 0x27 0x72 0x14 0x00 0xBD 0x68 0x52 0x9B 0x23 0xE5 0xD2 0x23 0x63 0x62 0x87 0x32 0x98 0xA2 0x7A 0x2E 0xDD 0x88 0xD5 0x10 0x0E 0x2B 0x5E 0xA0 0x66 0x89 0xEF 0xD3 0x7E 0x61 0xF9 0x6A 0x6A 0x73 0x4E 0xFE 0xCF 0x6F 0xA6 0xFC 0x67 0xFA 0x88 0xC2 0xA4 0xD5 0x13 0x31 0x12 0x5F 0xC1 0xE8 0x28 0x98 0x87 0x2C 0x43 0xF9 0x11 0x1E 0xC9 0x76 0x16 0xD6 0x9D 0x9D 0x68 0x89 0x7D 0x85 0x0D 0x61 0xB4 0x12 0xB3 0xB5 0x95 0x84 0xCD 0xCA

0x44 0x92 0x9E 0x10 0x22 0x4A 0x10 0x8F 0xB1 0xEE 0xC1 0x1D 0xD4 0xAF

Secure Element Specific APDU Error Codes

This table contains the expected error codes and descriptions that a caller may encounter while working with the Secure Element Applet.

Error Code	Description
0x6301	PIN verification required before executing a command
0x6302	PIN verification failed – wrong PIN code
0x6303	Wrong PIN size
0x6304	Maximum number of tax categories exceeded
0x6305	Maximum amount of tax exceeded (Sign Invoice) or Audit has not been started yet (End Audit)
0x6306	Audit has not started yet
0x6310	The number of allowed PIN entries exceeded
0x63FF	8-byte arithmetic operation overflow
0x6700	Data must be 256 bytes long
0x6A80	Audit Identification is not valid

File-Based Communication

This section contains the description of the **File-based** communication with E-SDC.

1. SD Cards or Flash Memory Drives Format
   Each E-SDC shall work with the following file system formats of SD Cards and USB Flash drives:

2. Tax Inspector Configures a New E SDC Using an SD Card

JSON file (*.commands) with commands shall be stored in a subfolder named after UID of the card inserted in the E-SDC. e.g D:\YJ37C9Z9\YJ37C9Z9.commands

3. E SDC Executes Commands Received via SD Card or USB Drive
   An E-SDC shall process commands automatically upon insertion of SD Card or USB Flash drive. Command execution takes precedence over a Local audit. Command types and the structure are explained in the section Commands.

4. E SDC Stores a Command Execution Result to the SD Card or USB Drive
   After commands have been executed, E-SDC must store a JSON file (.results) with the result to the UID folder on the SD Card/USB drive.

5. E SDC Stores Audit Files on SD Card or USB Drive
   An E-SDC shall perform an audit automatically once an SD Card or USB drive is inserted. If any commands are received on the same medium, they shall be executed before the proceeding with the Local audit.

SD Cards or Flash Memory Drives Format

Each E-SDC shall work with the following file system formats of SD Cards and USB Flash drives:

- FAT
- FAT32
- NTFS

Tax Inspector Configures a New E-SDC Using an SD Card

JSON file (*.commands) with commands shall be stored in a subfolder named after UID of the card inserted in the E-SDC. e.g D:\YJ37C9Z9\YJ37C9Z9.commands

JSON File Format

```
{
  "Commands": [
    {
      "CommandId": "GUID",
      "Type": 0,
      "Payload": "Command Specific Json as string",
      "UID": "string"
    }
  ]
}
```

}

E-SDC Executes Commands Received via SD Card or USB Drive

An E-SDC shall process commands automatically upon insertion of SD Card or USB Flash drive. Command execution takes precedence over a Local audit. Command types and the structure are explained in the section Commands.

JSON file with commands shall be stored in the subfolder named after UID value.

E-SDC shall execute only those commands with the same UID as UID assigned to the digital certificate of the Secure Element (stored in the *SerialNumber* field of the certificate subject).

E-SDC Stores a Command Execution Result to the SD Card or USB Drive

After commands have been executed, E-SDC must store a JSON file (.results) with the result to the UID folder on the SD Card/USB drive.

Example: **G:\BJ3PN1S9\BJ3PN1S9.results**

where **G is the root of the SD card/USB drive** and **BJ3PN1S9 is an example UID** of the smart card in use

The existing file on the SD Card/USB drive must be overwritten.

E-SDC Stores Audit Files on SD Card or USB Drive

An E-SDC shall perform an audit automatically once an SD Card or USB drive is inserted. If any commands are received on the same medium, they shall be executed before the proceeding with the Local audit.

All files shall be stored in the subfolder titled Audit of the UID folder.

Example: G:\BJ3PN1S9\Audit\
where G is the root of the SD card/USB drive

If the folder does not exist, an E-SDC shall create a new one.

An Audit consists of two file types:

1. One ARP.bin file containing the result of the invocation of Begin Audit APDU command (256Bytes)
2. One or more audit package files named using the following convention: {UID}-{UID}-{Ordinal_Number}.json

Manuals

E-SDC must have a user manual that explains the following topics in detail:

1. System requirements
2. Installation instructions for the technicians performing the installation and integration of an E-SDC device or software at a sales point
3. Properly connecting the E-SDC to a POS
4. User instructions for the operator (cashier or shopkeeper) explaining normal operations in detail
    - Information about the supported card readers or integrated card reader
    - How commands are received and executed via USB drive or SD card
    - How to obtain help/support regarding the product
    - Steps for troubleshooting
    - How the product license is activated
    - Local and/or remote audit instructions
    - How to properly shut down or uninstall the product
5. A detailed explanation (with visuals) for each product notification
6. Explanation of each custom error (manufacturer-specific)