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

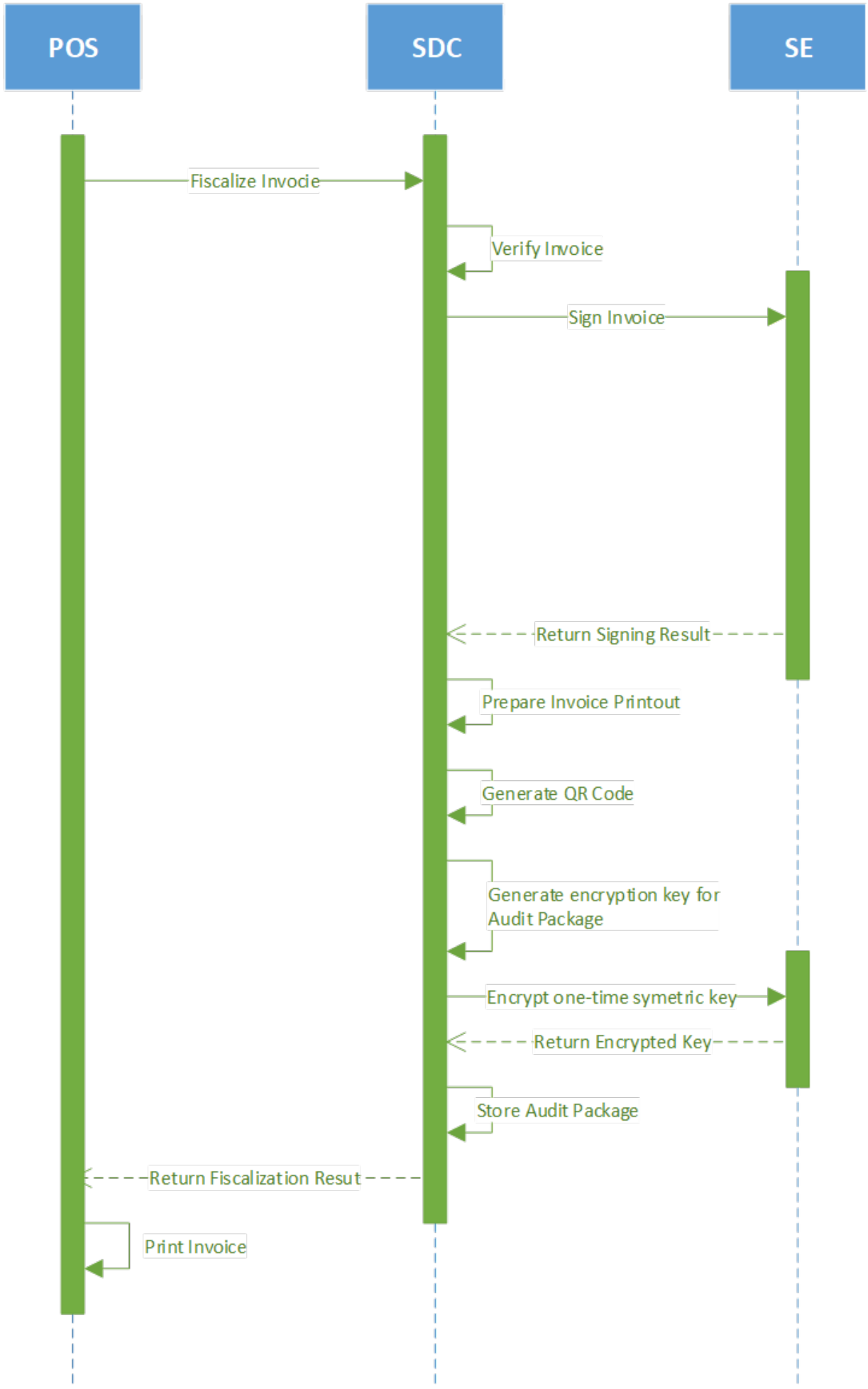
Invoice fiscalization is the main function of an E-SDC. Fiscalization is the process of handling invoice request from an accredited invoicing system in order to produce fiscal invoices.

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).

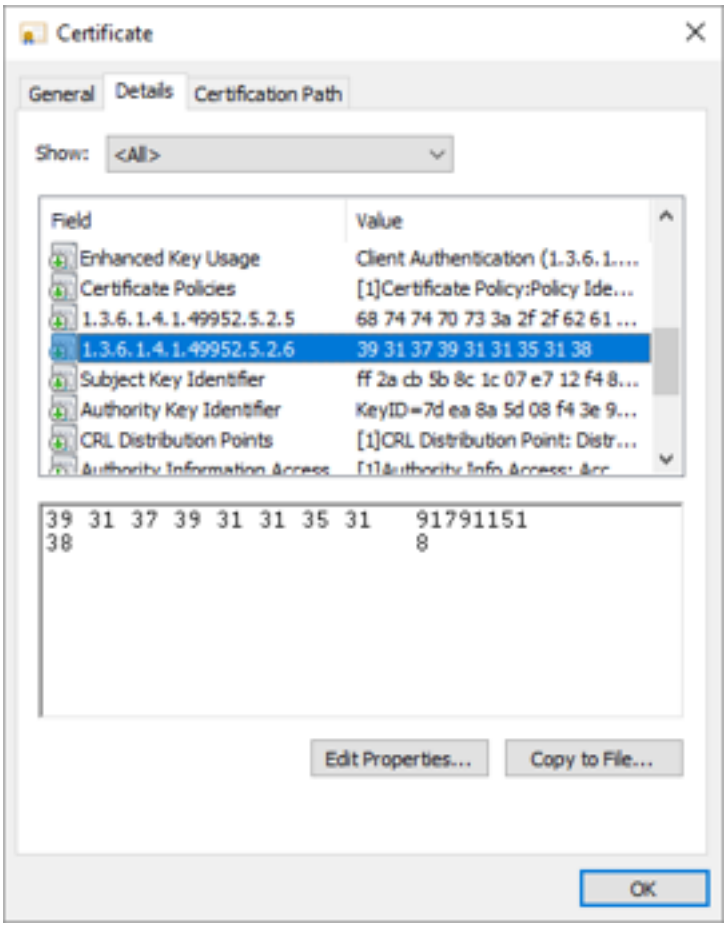
How To Obtain a Taxpayer Identification Number - TIN

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.

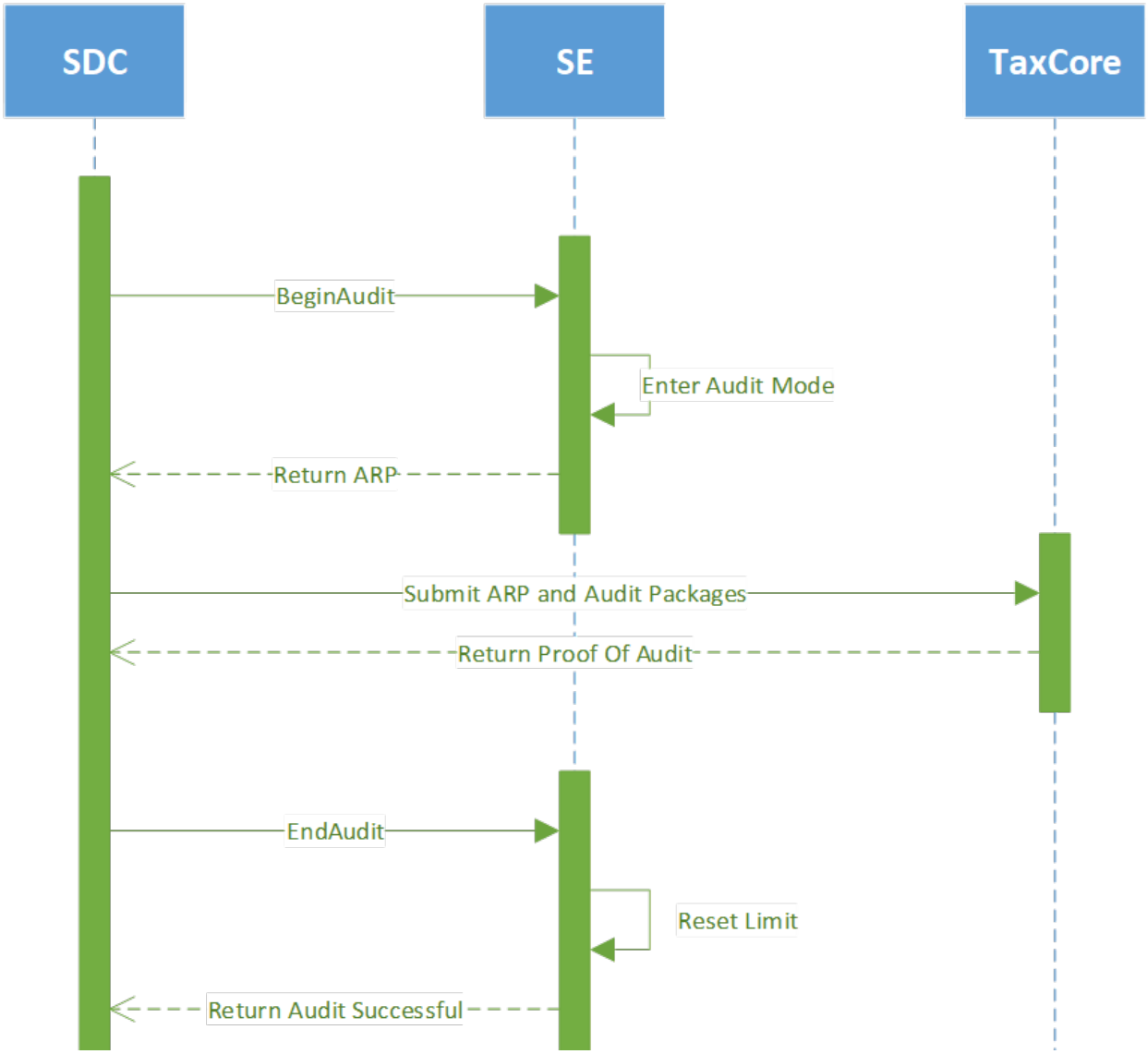
Depending on the connection availability, an audit may be triggered by the arrival of a signed invoice from the Secure Element or after the insertion of an external memory device into the E-SDC. Regardless of the event which has triggered the audit, the following conversation shall take place between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

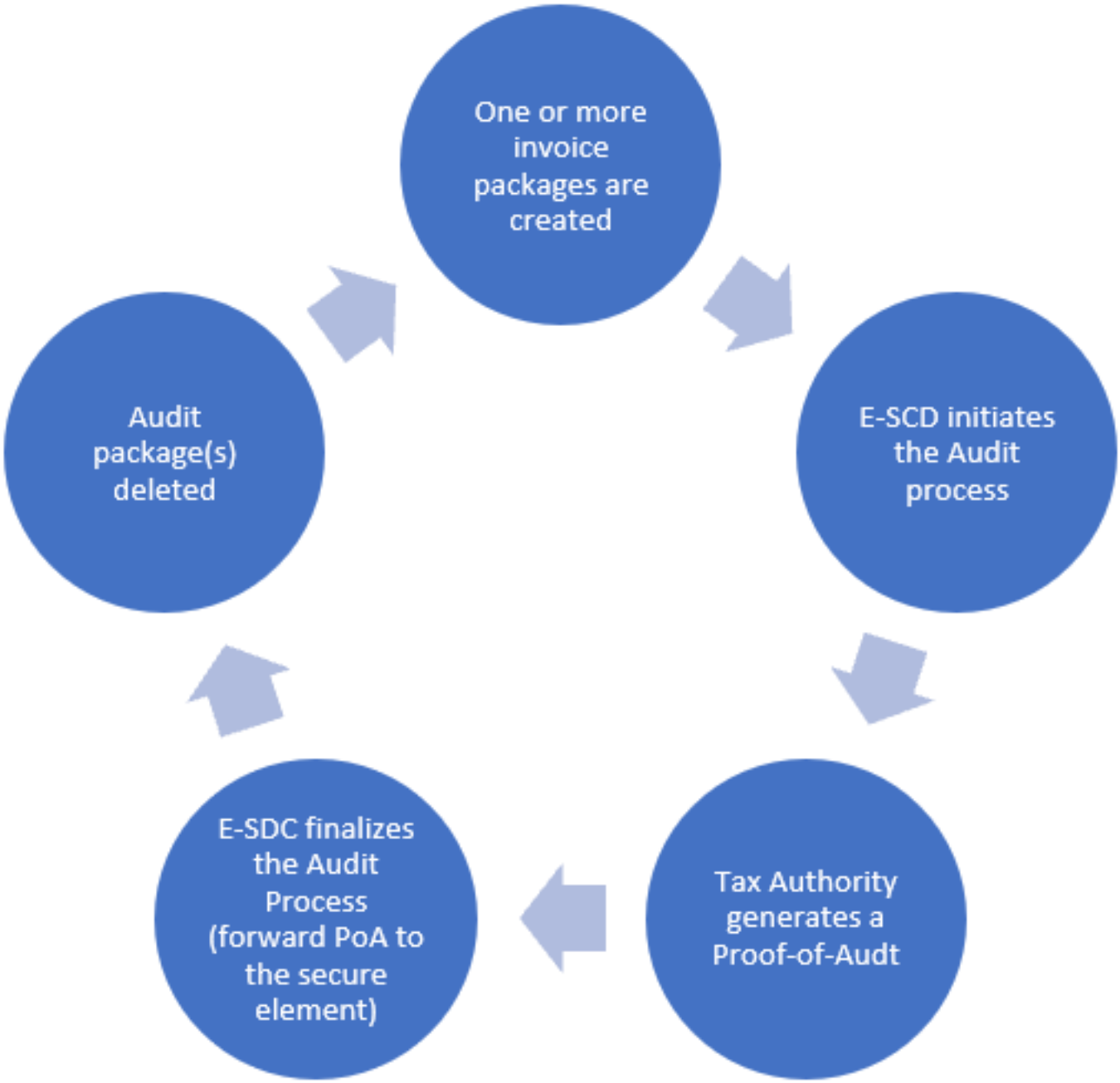
A taxpayer inserts media with proof of audit file into an E-SDC. The E-SDC loads proof of audit and verifies if the format is valid. If the format is valid, proof of audit is sent to the Secure Element for processing.

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).

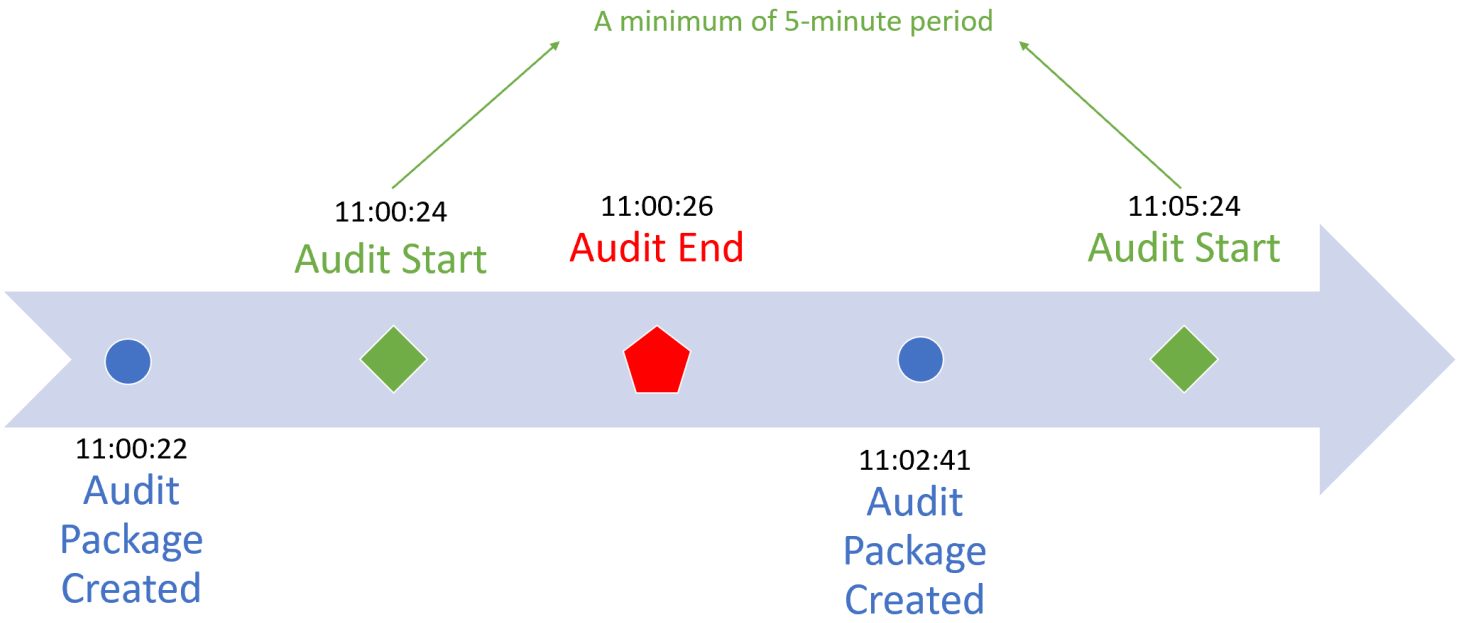
Case 1 – Audit is performed after the creation of each audit package

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:



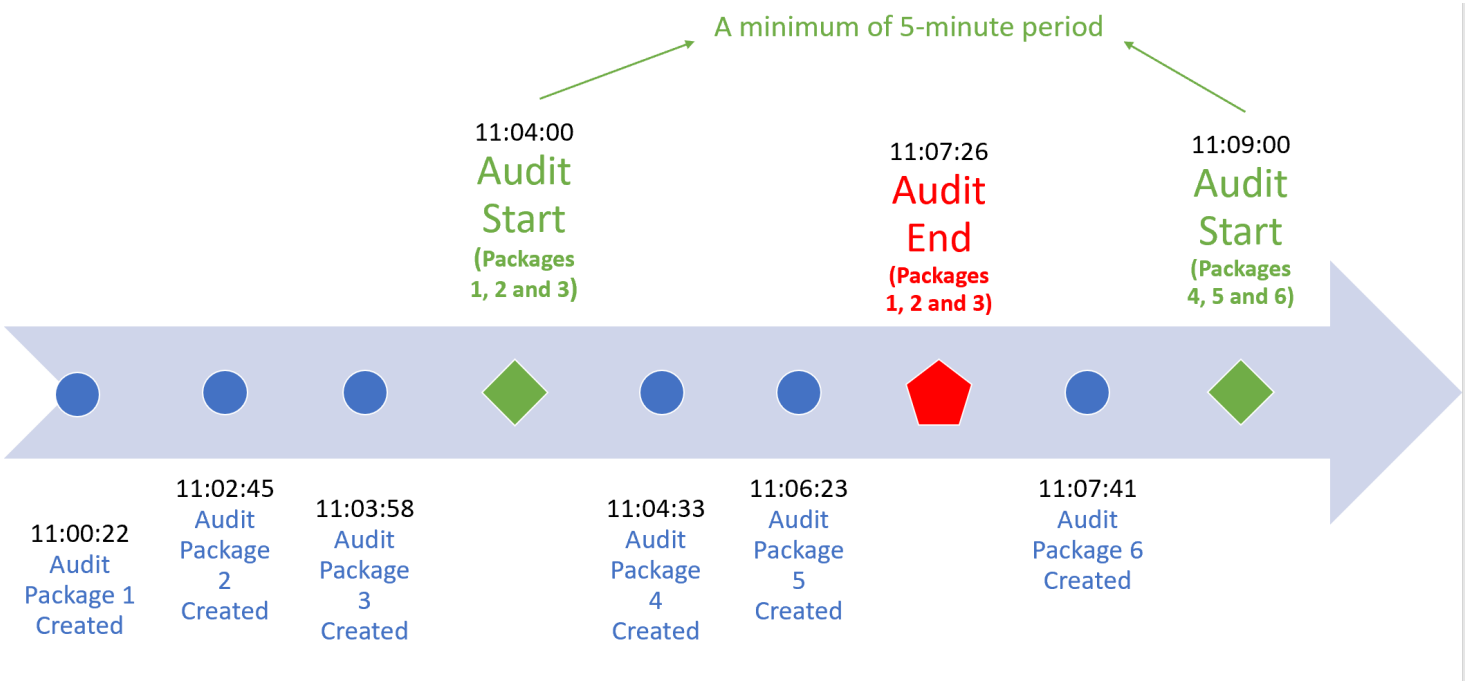
Proof of Audit – Image showing that the audit is performed after the creation of each audit package

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:



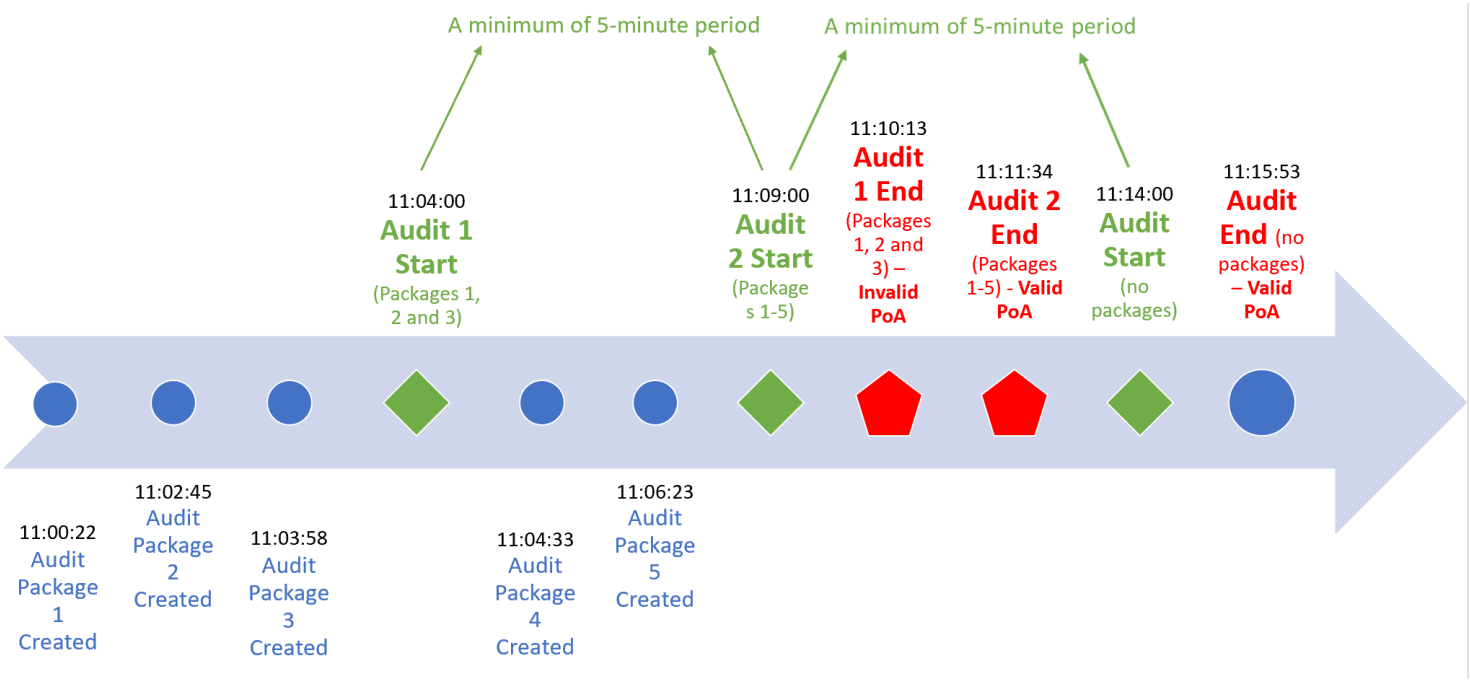
Proof of Audit – Image showing that the audit is performed after multiple audit packages have been created

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.